LECTURE NOTES IN COMPUTATIONAL SCIENCE AND ENGINEERING 88

Jochen Garcke · Michael Griebel *Editors*

Lecture Notes
in Computational Science
and Engineering 88

Editors:



For further volumes:
http://www.springer.com/series/3527

Jochen Garcke • Michael Griebel
Editors

# Sparse Grids and Applications

Springer

*Editors*
Jochen Garcke
Michael Griebel
Institut für Numerische Simulation
Universität Bonn
Bonn
Germany

ISSN 1439-7358
ISBN 978-3-642-31702-6 ISBN 978-3-642-31703-3 (eBook)
DOI 10.1007/978-3-642-31703-3
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012950005

Mathematics Subject Classification (2010): 65D99, 65M12, 65N99, 65Y20, 65N12, 62H99



*Cover illustration*: By courtesy of Dirk Pflüger

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Preface

In the recent decade, there has been growing interest in the numerical treatment of high-dimensional problems. It is well known that classical numerical discretization schemes fail in more than three or four dimensions due to the curse of dimensionality. The technique of sparse grids allows to overcome this problem to some extent under suitable regularity assumptions. This discretization approach is obtained from a multi-scale basis by a tensor product construction and subsequent truncation of the resulting multiresolution series expansion.

Hans-Joachim Bungartz, Jochen Garcke, Michael Griebel, and Markus Hegland organized a workshop specifically to strengthen the research on the mathematical understanding and analysis of sparse grid discretization. Particular focus was given to aspects arising from applications. More than 40 researchers from four different continents attended the workshop in Bonn, Germany, from May 16–20, 2011.

This volume of LNCSE now comprises selected contributions from attendees of the workshop. The contents range from numerical analysis and stochastic partial differential equations to applications in data analysis, finance, and physics.

The workshop was hosted by the Institut für Numerische Simulation and the Hausdorff Research Institute for Mathematics (HIM) of the Rheinische Friedrich-Wilhelms-Universität Bonn as part of the Trimester Program *Analysis and Numerics for High Dimensional Problems*. Financial support of the HIM is kindly acknowledged. We especially thank Christian Rieger for his efforts and enthusiasm in the local organization of the workshop and the staff of the HIM for their assistance.

Bonn, Germany

Jochen Garcke
Michael Griebel

# Contents

# Contributors

**Bastian Bohn** Institute for Numerical Simulation, University of Bonn, Bonn, Germany

**Hans-Joachim Bungartz** Technische Universität München, Garching, Germany

**Mike Espig** Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany

**Jochen Garcke** Institut für Numerische Simulation, Universität Bonn, Bonn, Germany

Fraunhofer SCAI, Schloss Birlinghoven, Sankt Augustin, Germany

**Thomas Gerstner** Institut für Mathematik, Johann Wolfgang Goethe-Universität, Frankfurt am Main, Germany

**Michael Griebel** Institute for Numerical Simulation, University of Bonn, Bonn, Germany

**Wolfgang Hackbusch** Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany

**Helmut Harbrecht** Mathematisches Institut, Basel, Switzerland

**Markus Hegland** Australian National University, Canberra, Australia

**Stefan Heinz** Institut für Mathematik, Johann Wolfgang Goethe-Universität, Frankfurt am Main, Germany

**Alexander Hullmann** Institute for Numerical Simulation, University of Bonn, Bonn, Germany

**John D. Jakeman** Department of Mathematics, Purdue University, West Lafayette, IN, USA

**Frank Jenko** Max-Planck-Institut für Plasmaphysik, Garching, Germany

**Christoph Kowitz** Institute for Advanced Study, Technische Universität München, Munich, Germany

**Alexander Litvinenko** Technische Universität Braunschweig, Braunschweig, Germany

**Hermann G. Matthies** Technische Universität Braunschweig, Braunschweig, Germany

**Benjamin Peherstorfer** Technische Universität München, Garching, Germany

**Michael Peters** Mathematisches Institut, Basel, Switzerland

**Dirk Pflüger** Institute for Parallel and Distributed Systems, University of Stuttgart, 70569 Stuttgart, Germany

**Christoph Reisinger** Mathematical Institute, University of Oxford, Oxford, UK

**Stephen G. Roberts** Australian National University, Canberra, Australia

**Markus Siebenmorgen** Mathematisches Institut, Basel, Switzerland

**Elmar Zander** Technische Universität Braunschweig, Braunschweig, Germany

**Stefan Zimmer** Universität Stuttgart, Stuttgart, Germany

# An Adaptive Sparse Grid Approach for Time Series Prediction

**Bastian Bohn and Michael Griebel**

**Abstract** A real valued, deterministic and stationary time series can be embedded in a—sometimes high-dimensional—real vector space. This leads to a one-to-one relationship between the embedded, time dependent vectors in $\mathbb{R}^d$ and the states of the underlying, unknown dynamical system that determines the time series. The embedded data points are located on an $m$-dimensional manifold (or even fractal) called attractor of the time series. Takens' theorem then states that an upper bound for the embedding dimension $d$ can be given by $d \leq 2m + 1$.

The task of predicting future values thus becomes, together with an estimate on the manifold dimension $m$, a scattered data regression problem in $d$ dimensions. In contrast to most of the common regression algorithms like support vector machines (SVMs) or neural networks, which follow a data-based approach, we employ in this paper a sparse grid-based discretization technique. This allows us to efficiently handle huge amounts of training data in moderate dimensions. Extensions of the basic method lead to space- and dimension-adaptive sparse grid algorithms. They become useful if the attractor is only located in a small part of the embedding space or if its dimension was chosen too large.

We discuss the basic features of our sparse grid prediction method and give the results of numerical experiments for time series with both, synthetic data and real life data.

B. Bohn (✉) · M. Griebel
Institute for Numerical Simulation, University of Bonn, 53115, Bonn, Germany
e-mail: bohn@ins.uni-bonn.de; griebel@ins.uni-bonn.de

J. Garcke and M. Griebel (eds.), *Sparse Grids and Applications*, Lecture Notes in Computational Science and Engineering 88, DOI 10.1007/978-3-642-31703-3_1,

## 1 Introduction and Problem Formulation

One of the most important tasks in the field of data analysis is the prediction of future values from a given time series of data. In our setting, a time series $(s_j)_{j=1}^{\infty}$ is an ordered set of real values. The task of forecasting can now be formulated as:

Given the values $s_1, \ldots, s_N$, predict $s_{N+1}$!

To tackle the forecasting problem, we assume that the values $s_j$ stem from an underlying stationary process which evolves in time. The aim is then to reconstruct the domain of this process as good as possible from the data $s_1, \ldots, s_N$ and to use this reconstruction for the prediction of the value $s_{N+1}$. To this end, let $M_0$ represent the phase space of the underlying system, let $\phi : M_0 \to M_0$ denote the corresponding equations of motion and let $o : M_0 \to \mathbb{R}$ be an observable which defines a time series by

$$(s_j)_{j=1}^{\infty} = \left(o\left(\phi^j(\mathbf{x}_0)\right)\right)_{j=1}^{\infty}, \tag{1}$$

where $\mathbf{x}_0 \in M_0$ is an arbitrary initial condition of the process and

$$\phi^j = \underbrace{\phi \circ \phi \circ \ldots \circ \phi}_{j \text{ times}}.$$

In practice $M_0$, $\phi$ and $o$ are of course not known, but only the values $s_j$ of the time series are given. To tackle the forecasting problem, we need to find a connection between the past values of the time series and the next one.

Takens' theorem [2, 25] provides the theoretical background to construct algorithms for this purpose. Assuming that a given equidistant time series consists of measurements, i.e. evaluations of the observable $o$, of an $m$-dimensional process which follows some deterministic equation of motion $\phi$, there is the possibility to find a regular $m$-dimensional submanifold $U$ of $\mathbb{R}^{2m+1}$ which is diffeomorphic to the phase space $M_0$ of the underlying system. The most common construction of such a submanifold works via delay embedding. In this case the time-dependent observations $s_j$ are themselves used as coordinates to represent $U \subset \mathbb{R}^{2m+1}$. Here, since an element $\mathbf{x} \in U$ corresponds to one specific point in phase space, the dynamics of the process and thus the evolution of the time series itself are completely determined by $\mathbf{x}$ and the forecasting problem translates into an ordinary regression-like task in $\mathbb{R}^{2m+1}$. For an overview of the delay embedding scheme and different approaches to time series analysis see [21].

In this paper we use a regularized least squares approach to find an adequate approximation to the solution of the prediction problem. In combination with a FEM-like grid discretization this leads to a non-data-based approach whose computational costs grow only linearly in the number of elements of the given time series. Then, in contrast to most data-based techniques like support vector machines or standard neural networks using radial basis functions, this approach

is able to handle huge time series. But, if $d = 2m + 1$ denotes the dimension of the ambient space and $2^t$ is the number of grid points in one direction, the number of points in a conventionally discretized ambient space would grow like $O\left(2^{td}\right)$. Thus, this naive approach suffers from the curse of dimensionality which restricts the application of a conventional discretization to low-dimensional, i.e. to one-, two- or three-dimensional spaces.

To circumvent this problem the sparse grid discretization [1] is used in this paper. A first approach for the prediction of financial time series with sparse grids has been presented in [8]. For *regular sparse grids*, the number of grid points increases only like $O(2^t \cdot t^{d-1})$, i.e. the curse of dimensionality is now just present with respect to the term $t$. This way, we are able to efficiently deal with huge amounts of data in moderate dimensions up to about $d = 10$. Moreover, for most time series the high-dimensional data does not fill the whole space. The process obtained by using the delay embedding method then visits only a small fraction of the whole discretized area. This observation justifies a *space-adaptive sparse grid* [13] discretization which resolves the trajectory of the process. Finally, an ANOVA-like approach leads to *dimension-adaptive sparse grids* [10, 17] that are useful if our a priori choice of $d$ is too large.

Thus, we will introduce two different adaptive algorithms in this paper: The space-adaptive algorithm locally adapts to features of the prediction function whereas the dimension-adaptive algorithm refines by employing subspaces which are relevant for an efficient representation of the prediction function in its ANOVA-decomposition.

In summary, each of our algorithms processes the following steps:

1. Estimation of the dimension $m$ of the underlying process
2. Rewriting the forecasting problem as a regression problem in $\mathbb{R}^d$ with $d = 2m+1$
3. Approximating the solution of the regression problem in a discretized (regular, space-adaptive or dimension-adaptive) sparse grid space
4. Predicting the value $s_{N+1}$ by point evaluation of the computed sparse grid function at $(s_{N-2m}, \ldots, s_N)^T$

Altogether, we obtain a new class of algorithms for the prediction of time series data which scale only linearly with the length of the given time series, i.e. the amount of data points, but still allow us to use reasonably large window sizes for the delay embedding due to our sparse grid approach. The new methods give excellent prediction results with manageable computational costs.

The remainder of this paper is organized as follows: In Sect. 1, we describe the delay embedding scheme and review some crucial issues concerning the application of Takens' theorem. In Sect. 2, we show how the forecasting problem can be rewritten as a regression problem. We also derive the regularized least squares functional which determines our predictor function. In Sect. 3, we deal with the regular sparse grid approximation. We deduce the associated linear system and solve it using a preconditioned CG-algorithm. Then, we introduce and discuss space- and

dimension-adaptive sparse grid algorithms. In Sect. 4, we give the results of numerical experiments which illustrate the favorable properties of our new methods.

## 2 Takens' Theorem and the Delay Embedding Scheme

We now provide the essential theory concerning Takens' theorem [25] and give a hint to some modifications from [2].

For an arbitrary $d \in \mathbb{N}$ we can create vectors

$$\mathbf{t}_j := \left(s_{j-d+1}, s_{j-d+2}, \ldots, s_{j-1}, s_j\right)^T \in \mathbb{R}^d, \ j \geq d$$

following the so-called delay embedding scheme. Each vector consists of $d$ consecutive past time series values. A connection between these delay vectors and the unknown evolution of the process in the phase space is established by the following theorem:

**Theorem 1.** *Let $M_0$ be a compact $m$-dimensional $C^2$-manifold, let $\phi : M_0 \to M_0$ denote a $C^2$-diffeomorphism and let $o \in C^2(M_0, \mathbb{R})$. Then, $\boldsymbol{\rho}_{(\phi,o)} : M_0 \hookrightarrow \mathbb{R}^{2m+1}$ defined by*

$$\boldsymbol{\rho}_{(\phi,o)}(\mathbf{x}) := \left(o(\mathbf{x}), o(\phi(\mathbf{x})), o\left(\phi^2(\mathbf{x})\right), \ldots, o\left(\phi^{2m}(\mathbf{x})\right)\right) \tag{2}$$

*is generically*[1] *an embedding.*

This is Takens' theorem for discrete time series, see [19, 25]. The embedding $\boldsymbol{\rho}_{(\phi,o)}$ establishes a one-to-one connection between a state in the phase space $M_0$ and a $(2m+1)$-dimensional delay vector constructed by (2). It can formally be inverted and we obtain

$$\begin{aligned}
\phi^{j-2m}(\mathbf{x}_0) &= \boldsymbol{\rho}^{-1}_{(\phi,o)}\left(\left(o\left(\phi^{j-2m}(\mathbf{x}_0)\right), o\left(\phi^{j-2m+1}(\mathbf{x}_0)\right), \ldots, o\left(\phi^{j}(\mathbf{x}_0)\right)\right)\right) \\
&= \boldsymbol{\rho}^{-1}_{(\phi,o)}\left(\left(s_{j-2m}, s_{j-2m+1}, \ldots, s_j\right)\right) \\
&= \boldsymbol{\rho}^{-1}_{(\phi,o)}\left(\mathbf{t}_j\right)
\end{aligned}$$

[1] Here, "generically" means the following:
If $X_l := \left\{\mathbf{x} \in M_0 \mid \phi^l(\mathbf{x}) = \mathbf{x}\right\}$ fulfills $|X_l| < \infty$ for all $l \leq 2m$ and if the Jacobian matrix $\left(\mathrm{D}\phi^l\right)_{\mathbf{x}}$ of $\phi^l$ at $\mathbf{x}$ has pairwise distinct eigenvalues for all $l \leq 2m, \mathbf{x} \in X_l$, then the set of all $o \in C^2(M_0, \mathbb{R})$ for which the embedding property of Theorem 1 does not hold is a null set. As $C^2(M_0, \mathbb{R})$ is an infinite dimensional vector space, the term "null set" may not be straightforward. It should be understood in the way that every set $Y \supset \left\{o \in C^2(M_0, \mathbb{R}) \mid \boldsymbol{\rho}_{(\phi,o)} \text{ is an embedding}\right\}$ is prevalent.

for all $j \geq d$ with $\mathbf{t}_j \in \mathbb{R}^{2m+1}$ and thus $d = 2m + 1$. Applying $o \circ \phi^{2m+1}$ on both sides we obtain

$$o\left(\phi^{j+1}(\mathbf{x}_0)\right) = o\left(\phi^{2m+1}\left(\boldsymbol{\rho}^{-1}_{(\phi,o)}\left(\mathbf{t}_j\right)\right)\right). \tag{3}$$

This means that the value $s_{j+1} = o\left(\phi^{j+1}(\mathbf{x}_0)\right)$ is completely determined by the previous $2m+1$ values $s_{j-2m}, \ldots, s_j$.[2] Note that not necessarily all of the preceding $2m + 1$ values are essential to specify the current one, but Theorem 1 states that $2m + 1$ values are always sufficient to do so. Note furthermore that not only the next but all following values are determined by $2m + 1$ consecutive time series values. To see this one can just recursively follow the scheme in (3). Thus, if we have for example an equidistant time series with a 1 min gap between successive values but are interested in a 15 min forecast, we can still use $2m + 1$ consecutive values as input in our regression algorithm later on.[3]

Often, the equations of motion are described by a system of time-continuous differential equations instead of a time-discrete mapping $\phi$ as in Theorem 1. To this end, let $V$ denote a vector field in $C^2(M_0, TM_0)$, let $o \in C^2(M_0, \mathbb{R})$ and let $\mathbf{z} : \mathbb{R}^+ \to M_0$ fulfill the differential equation

$$\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = \mathbf{V}(\mathbf{z}),\ \mathbf{z}(0) = \mathbf{z}_0 \tag{4}$$

for given $\mathbf{z}_0 \in M_0$. We define $\phi_t(\mathbf{z}_0) := \mathbf{z}(t)$ as the flow of the vector field $\mathbf{V}$. Now $\phi_\tau$ can be used in Theorem 1 instead of $\phi$ for an arbitrary $\tau \in \mathbb{R}^+$ and the time-continuous setting is covered as well. For a thorough treatment of this case we refer to [2, 25].

A main requirement for Takens' theorem is the compactness of the manifold $M_0$, i.e. the domain of the process which contains all possible states. Sometimes the dynamics tends to form a so-called "strange attractor", which means that the trajectories of the system do not form a manifold anymore but just a point set $A$ of non-integer dimension. In [2] it was shown that it is possible to generalize Theorem 1 also to this case:

**Theorem 2.** *Let $A \subset M_0 \subset \mathbb{R}^k$ where $M_0$ is an open subset of $\mathbb{R}^k$ and $A$ is a compact subset of $M_0$ which possesses box-counting dimension $\widehat{\dim}(A) = m$. Furthermore, let $\phi : M_0 \to M_0$ be a $C^2$-diffeomorphism and let $o \in C^2(M_0, \mathbb{R})$. Then, for $\boldsymbol{\rho}_{(\phi,o)} : M_0 \hookrightarrow \mathbb{R}^{\lfloor 2m+1 \rfloor}$ defined as in (2), the properties*

[2] All functions on the right hand side of (3) are at least twice differentiable. As $M_0$ is compact, the concatenation of these functions lies in the standard Sobolev space $H_2(\boldsymbol{\rho}_{(\phi,o)}(M_0))$, where $\boldsymbol{\rho}_{(\phi,o)}(M_0) \subset \mathbb{R}^{2m+1}$ denotes the image of $M_0$ under $\boldsymbol{\rho}_{(\phi,o)}$.

[3] An alternative would be to simulate a time series with 15 min gaps by omitting intermediate values which would lead to a considerable reduction of the number of points. This is however not advantageous, as more points usually lead to better prediction results for the numerical algorithm.

1. $\boldsymbol{\rho}_{(\phi,o)}$ *is one-to-one on* $A$ *and*
2. $\boldsymbol{\rho}_{(\phi,o)}$ *is an immersion on each compact subset* $C$ *of a smooth manifold contained in* $A$

*generically*[4] *hold.*

Here, $\lfloor a \rfloor$ denotes the largest integer which is smaller or equal to $a \in \mathbb{R}^+$.

In real world applications, the set $A$ is not a priori known. But for the delay embedding scheme to work we only need to know the box-counting dimension $\widehat{\dim}(A)$ of the set $A$. Its estimation is an elaborate task by its own. To this end, various approaches exist in the literature [22, 23, 26]. Here, we recommend using the Grassberger-Procaccia algorithm [12] to estimate the correlation dimension $\tilde{m}$ as an approximation of the box-counting dimension $m$ since this worked best in our experiments. The delay length is then set to $d = \lfloor 2\tilde{m} + 1 \rfloor$.

In summary we have a theory which provides us with a justification to use delayed vectors like in (2) as input for a learning tool.

## 3 The Regression Problem and the Regularized Least Squares Approach

In this section, we describe how the task of predicting a time series can be recast into a higher-dimensional regression problem by means of delay embedding. Furthermore, we motivate a specific regularized least squares approach.

We assume that we have an infinite time series $\left(s_j\right)_{j=1}^{\infty}$ which is just an observation of a deterministic process on an $m$-dimensional attractor, compare Sect. 2. From now on, let $d := \lfloor 2m + 1 \rfloor$ denote the embedding dimension used for the delay scheme. We define

$$\mathbf{t}_j := \left(s_{j-d+1}, s_{j-d+2}, \ldots, s_{j-1}, s_j\right)^T \in \mathbb{R}^d, \; j \geq d, \tag{5}$$

to be the $j$-th delay vector. Due to Takens' theorem, there exists a $\hat{g} : \mathbb{R}^d \to \mathbb{R}$, with $\hat{g} := o \circ \phi^d \circ \boldsymbol{\rho}_{(\phi,o)}^{-1}$, cf. (3), such that

$$\hat{g}\left(\mathbf{t}_j\right) = s_{j+1} \text{ for all } j \geq d. \tag{6}$$

If we assume that $\left(s_j\right)_{j=1}^{N}$ is known a priori then our goal is to find a good approximation $g$ to $\hat{g}$ with the help of $N - d + 1$ training patterns

[4] Here "generically" means the following:
If $\tilde{X}_l := \left\{\mathbf{x} \in A \mid \phi^l(\mathbf{x}) = \mathbf{x}\right\}$ fulfills $\widehat{\dim}\left(\tilde{X}_l\right) \leq \frac{l}{2}$ for all $l \leq \lfloor 2m + 1 \rfloor$ and if $\left(\mathrm{D}\phi^l\right)_{\mathbf{x}}$ has pairwise distinct eigenvalues for all $l \leq \lfloor 2m + 1 \rfloor, \mathbf{x} \in \tilde{X}_l$, then the set of all $o \in C^2\left(M_0, \mathbb{R}\right)$ for which the properties in Theorem 2 do not hold is a null set.

$$\left(\mathbf{t}_j, s_{j+1}\right) \in \mathbb{R}^d \times \mathbb{R},\ j = d, \ldots, N-1. \tag{7}$$

Thus, we now have to deal with a regression problem instead of the forecasting problem. Our approach is to choose $g \in X \subset \left\{f : \mathbb{R}^d \to \mathbb{R}\right\}$ as

$$g = \arg\min_{f \in X} \mathcal{F}(f)$$

where $\mathcal{F} : X \to \mathbb{R}^+ \cup \{\infty\}$ is a functional that expresses how good functions from $X$ approximate $\hat{g}$. The function space $X$ still has to be specified.

To this end, as we do not know any embedded points on which we want to evaluate $g$ afterwards, it is common to minimize the expectation of some Lebesgue measurable cost function $c : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+ \cup \{\infty\}$ with respect to the density $p : \mathbb{R}^d \times \mathbb{R} \to [0, 1]$ of all possible input patterns. This leads to

$$\mathcal{F}(f) = \mathbb{E}_p\left[c\left(y, f(\mathbf{x})\right)\right]$$

and thus gives

$$g = \arg\min_{f \in X} \mathbb{E}_p\left[c\left(y, f(\mathbf{x})\right)\right] = \arg\min_{f \in X} \int_{\mathbb{R}^d \times \mathbb{R}} c\left(y, f(\mathbf{x})\right) p(\mathbf{x}, y)\left(\mathrm{d}\mathbf{x} \otimes \mathrm{d}y\right).$$

Note here that we have to restrict $X$ to contain only Lebesgue measurable functions to make this term well-defined.

Since we have to cope with training patterns and do not know the exact density $p$, we use the empirical density

$$\hat{p}(\mathbf{x}, y) = \frac{1}{N-d} \sum_{j=d}^{N-1} \delta_{\mathbf{t}_j}(\mathbf{x}) \delta_{s_{j+1}}(y)$$

instead of $p$. This results in the problem of finding the argument of the minimum of

$$\mathcal{F}(f) = \int_{\mathbb{R}^d \times \mathbb{R}} c\left(y, f(\mathbf{x})\right) \hat{p}(\mathbf{x}, y)\left(\mathrm{d}\mathbf{x} \otimes \mathrm{d}y\right) = \frac{1}{N-d} \sum_{j=d}^{N-1} c\left(s_{j+1}, f(\mathbf{t}_j)\right). \tag{8}$$

Note that if we want to calculate the point evaluations $f\left(\mathbf{t}_j\right)$, then the set of admissible functions $X$ has here to be restricted further to contain only functions for which point evaluations are well defined.

We decided to use $c(a, b) := (a - b)^2$. One can easily show that for this specific cost function a minimizer of $\mathcal{F}$ maximizes the likelihood of the given input data under the assumption of a Gaussian noise term being added to each exact time series value, see e.g. Sect. 3.3 in [24].[5]

[5] Other cost functions can be used as well but these might lead to non-quadratic or even non-convex minimization problems.

The minimization of (8) for $f \in X$ still leads to an ill-posed problem and a further restriction of the space of admissible functions is therefore needed. To this end, Tikhonov proposed to add a constraint of the form $\Psi(f) \leq c$ with an arbitrary positive constant $c$ and a nonnegative functional $\Psi : X \to \mathbb{R}^+$ which is strictly convex on a certain subspace depending on the problem itself, see [27]. Using the method of Lagrange multipliers we then obtain the new minimization problem

$$g = \arg\min_{f \in X} \mathcal{F}(f) := \arg\min_{f \in X} \left( \frac{1}{N-d} \sum_{j=d}^{N-1} c\left(s_{j+1}, f(\mathbf{t}_j)\right) + \lambda \Psi(f) \right) \tag{9}$$

which is well-posed if $\lambda$ is positive. We will employ the Sobolev semi-norm

$$\Psi(f) := |f|_{H^1_{\text{mix}}} = \sum_{|\mathbf{a}|_\infty = 1} \left\| \frac{\mathrm{d}^{a_1}}{\mathrm{d}x_1^{a_1}} \cdots \frac{\mathrm{d}^{a_d}}{\mathrm{d}x_d^{a_d}} f \right\|^2_{L_2(\mathbb{R}^d)} \tag{10}$$

since this perfectly fits after discretization to our basis functions as we will see later. Here $\mathbf{a} = (a_1, \ldots, a_d)$ denotes a multi index and $|\mathbf{a}|_\infty := \max_{i=1,\ldots,d} |a_i|$. We will use the function $g \in X$ defined in (9) as continuous approximation to $\hat{g}$ from now on.

Instead of our $H^1_{\text{mix}}$-semi-norm, a method using gradient penalties—which corresponds to the $H^1$ semi-norm—was presented in [9] and error bounds were provided for a discrete solution achieved by the so-called combination technique. Note that some of these results rely on the assumption of independent and uniformly distributed samples. Nevertheless, similar results can be given for our case under the assumption of independently drawn samples according to the probability distribution on the reconstructed attractor. The resulting errors then refer to the attractor measure instead of the Lebesgue measure.

### 3.1 Minimization for an Arbitrary Basis

Now let $\{\gamma_i\}_{i=1}^\infty$ be a basis of $\Gamma := \{f \in X \mid \Psi(f) \leq c\}$. Our task is to find a

$$\mathbf{w} := (w_1, w_2, \ldots)$$

with $w_i \in \mathbb{R}$ for each $i \in \mathbb{N} \setminus \{0\}$, which minimizes

$$\frac{1}{N-d} \sum_{j=d}^{N-1} \left( s_{j+1} - \sum_{i=1}^{\infty} w_i \gamma_i\left(\mathbf{t}_j\right) \right)^2 + \lambda \sum_{i=1}^{\infty} \sum_{k=1}^{\infty} \left( w_i w_k \sum_{|\mathbf{a}|_\infty = 1} \langle \mathrm{D}^{\mathbf{a}} \gamma_i, \mathrm{D}^{\mathbf{a}} \gamma_k \rangle_{L_2(\mathbb{R}^d)} \right) \tag{11}$$

where $\mathrm{D}^{\mathbf{a}} = \frac{\mathrm{d}^{a_1}}{\mathrm{d}x_1^{a_1}} \ldots \frac{\mathrm{d}^{a_d}}{\mathrm{d}x_d^{a_d}}$ denotes a multivariate derivative. The corresponding function

$$g(\mathbf{x}) = \sum_{i=1}^{\infty} w_i \gamma_i(\mathbf{x})$$

then would give us the approximate prediction $s_{N+1} \approx g(\mathbf{t}_N)$ by point evaluation at $\mathbf{t}_N$. As (11) is a sum of strictly convex functions, the argument $g$ of the minimum can be found by identifying the zeroes of $\frac{\mathrm{d}}{\mathrm{d}w_l} \mathcal{F}(f)$ for all $l \in \mathbb{N} \setminus \{0\}$. For $\eta := (N-d)\lambda$ this leads to the *infinite* system

$$\sum_{j=d}^{N-1} s_{j+1} \gamma_l(\mathbf{t}_j) = \sum_{i=1}^{\infty} w_i \left( \sum_{j=d}^{N-1} \gamma_l(\mathbf{t}_j) \gamma_i(\mathbf{t}_j) + \eta h(\gamma_i, \gamma_l) \right) \tag{12}$$

for all $l \in \mathbb{N} \setminus \{0\}$, where $h : \Gamma \times \Gamma \to \mathbb{R}$ denotes the semi-definite bilinear form

$$h(s,t) = \sum_{|\mathbf{a}|_\infty = 1} \langle \mathrm{D}^{\mathbf{a}} s, \mathrm{D}^{\mathbf{a}} t \rangle_{L_2(\mathbb{R}^d)}.$$

## 3.2 Minimization for a Kernel Basis in a Reproducing Kernel Hilbert Space

To derive a finite solution procedure, the following approach is standard in the mathematical learning community. For the case $\Psi(f) = ||f||_{\mathcal{H}}^2$, with $\mathcal{H}$ being a reproducing kernel Hilbert space, we can write $g$ from (9) as a finite linear combination of evaluations of the reproducing kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ in the points corresponding to the training patterns

$$g(\mathbf{x}) = \sum_{j=d}^{N-1} g_j k(\mathbf{t}_j, \mathbf{x})$$

with some real-valued weights $g_j$. This is known as the representer theorem for reproducing kernel Hilbert spaces, see e.g. [24]. Analogous observations as above result with the property $\langle k(\mathbf{t}_i, \cdot), k(\mathbf{t}_j, \cdot) \rangle_{\mathcal{H}} = k(\mathbf{t}_i, \mathbf{t}_j)$ in the finite system

$$\sum_{j=d}^{N-1} s_{j+1} k(\mathbf{t}_j, \mathbf{t}_l) = \sum_{j=d}^{N-1} g_j \left( \sum_{i=d}^{N-1} k(\mathbf{t}_i, \mathbf{t}_j) k(\mathbf{t}_i, \mathbf{t}_l) + \eta k(\mathbf{t}_j, \mathbf{t}_l) \right)$$

for all $l \in \{d, \ldots, N-1\}$. If the $\left(k\left(\mathbf{t}_j, \mathbf{x}\right)\right)_{j=d}^{N-1}$ are linearly independent[6] this leads to the linear system

$$\mathbf{s} = (\mathbf{K} + \eta \mathbf{I})\, \mathbf{g} \tag{13}$$

where $\mathbf{K} \in \mathbb{R}^{(N-d)\times(N-d)}$ is the kernel matrix with entries $\mathbf{K}_{i,j} = k\left(\mathbf{t}_i, \mathbf{t}_j\right)$, $\mathbf{I} \in \mathbb{R}^{(N-d)\times(N-d)}$ is the identity matrix and $\mathbf{g} = (g_d, \ldots, g_{N-1})^T \in \mathbb{R}^{N-d}$, $\mathbf{s} = (s_{d+1}, \ldots, s_N)^T \in \mathbb{R}^{N-d}$.

Note that for the case (10) we only regularized with a semi-norm of a reproducing kernel Hilbert space but still get the representation

$$g(\mathbf{x}) = \sum_{j=d}^{N-1} g_j k\left(\mathbf{t}_j, \mathbf{x}\right) + g_0(\mathbf{x})$$

with a $g_0 : \mathbb{R}^d \to \mathbb{R}$ from the null space of $\Psi$ and a certain kernel function $k$, see [24].

One could now try to solve the linear system (13). The major problem of this approach—besides the knowledge of an explicit formulation of the reproducing kernel[7]—is the complexity with respect to the number of input patterns. The direct solution of (13) would involve a number of operations of the order $O\left(N^3\right)$ since we have to deal with a full system matrix here. But even if one does not compute the inverse of $\mathbf{K} + \eta\mathbf{I}$ directly and uses an appropriate iterative scheme instead, the complexity for solving this system is at least $O\left(N^2\right)$ because of the dense kernel matrix $\mathbf{K}$. Therefore, in the next section, we will consider the infinite system (12) in the first place and resort to a further approximation of our prediction problem by discretization.

## 4 Discretization via Sparse Grids

To find an approximate solution to (12) we restrict ourselves to a finite dimensional subspace $\Gamma_M := \operatorname{span}\{\gamma_i\}_{i=1}^M \subset \Gamma := \{f \in X \mid \Psi(f) \leq c\}$ for some $M \in \mathbb{N}$. For the naive full grid approach the curse of dimensionality then shows up in the number of necessary grid points which grows exponentially with $d$. To deal with this issue, we will employ the sparse grid discretization technique and its adaptive enhancements here. To this end, we will assume that the domain of $\hat{g}$ (and thus $g$) is the $d$-dimensional hypercube

$$\mathbf{H}_d := [0,1]^d.$$

[6] If this is not the case we can choose a linearly independent subsystem and continue analogously.

[7] See [28] for several reproducing kernels and their corresponding Hilbert spaces.

Note that this is not a restriction since the domain of the underlying original process is compact (cf. Theorems 1 and 2). By rescaling the resulting domain of the reconstructed process we always can obtain the domain $[0, 1]^d$.

## 4.1 Multilevel Hierarchical Bases and Regular Sparse Grids

First, we recall the construction of a full grid space using a piecewise linear hierarchical basis and discuss its relation to a sparse grid space. Let the one-dimensional hat function $\phi : \mathbb{R} \to [0, 1]$ be defined by

$$\phi(x) := \begin{cases} 1 - |x|, & \text{if } x \in [-1, 1] \\ 0 & \text{else} \end{cases}$$

and let

$$\phi_{l,i}(x) := \phi(2^l \cdot x - i)|_{[0,1]}$$

for any $l, i \in \mathbb{N}$ be a dilated and rescaled version of $\phi$ restricted to the interval $[0, 1]$. One can easily see that $\operatorname{supp}(\phi_{l,i}) = \left((i-1)2^{-l}, (i+1)2^{-l}\right) \cap [0, 1]$. The construction of a $d$-dimensional hat function is straightforward via the tensor product

$$\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) := \prod_{j=1}^{d} \phi_{l_j,i_j}(x_j),$$

where $\mathbf{l} = (l_1, \ldots, l_d) \in \mathbb{N}^d$ is the multivariate level and $\mathbf{i} = (i_1, \ldots, i_d) \in \mathbb{N}^d$ denotes the multivariate position index. Furthermore, we define $\mathbf{x}_{\mathbf{l},\mathbf{i}} := \mathbf{i} \cdot 2^{-\mathbf{l}}$, where the multiplication has to be understood componentwise, i.e. $\mathbf{x}_{\mathbf{l},\mathbf{i}} = \left(x_{l_1,i_1}, \ldots, x_{l_d,i_d}\right)^T$ with $x_{l_j,i_j} := i_j \cdot 2^{-l_j}$. For a fixed $\mathbf{l} \in \mathbb{N}^d$, we then have with

$$\Omega_{\mathbf{l}} := \left\{\mathbf{x}_{\mathbf{l},\mathbf{i}} \mid \mathbf{0} \leq \mathbf{i} \leq 2^{\mathbf{l}}\right\}$$

the full grid of level $\mathbf{l}$. Here, the inequalities are to be understood componentwise and $\mathbf{0} = (0, \ldots, 0)$ is the null index. The space of piecewise $d$-linear functions on the grid $\Omega_{\mathbf{l}}$ is

$$V_{\mathbf{l}} := \operatorname{span}\{B_{\mathbf{l}}\} \quad \text{with } B_{\mathbf{l}} = \left\{\phi_{\mathbf{l},\mathbf{i}} \mid \mathbf{0} \leq \mathbf{i} \leq 2^{\mathbf{l}}\right\}.$$

$B_{\mathbf{l}}$ is called nodal basis since the value of a function $f_{\mathbf{l}}(\mathbf{x}) = \sum_{\mathbf{0} \leq \mathbf{i} \leq 2^{\mathbf{l}}} f_{\mathbf{l},\mathbf{i}} \cdot \phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) \in V_{\mathbf{l}}$ on one of the grid points $\mathbf{x}_{\mathbf{l},\mathbf{j}}$ of $\Omega_{\mathbf{l}}$ is given by the coefficient $f_{\mathbf{l},\mathbf{j}} \in \mathbb{R}$ that corresponds to $\phi_{\mathbf{l},\mathbf{j}}$.

Now, let

$$\mathbf{I}_{\mathbf{l}} := \left\{ \mathbf{i} \in \mathbb{N}^d \;\middle|\; \begin{array}{ll} 0 \leq i_j \leq 1, & \text{if } l_j = 0 \\ 1 \leq i_j \leq 2^{l_j} - 1,\ i_j \text{ odd} & \text{if } l_j > 0 \end{array} \text{ for all } 1 \leq j \leq d \right\}. \quad (14)$$

Then, $W_{\mathbf{l}} := \operatorname{span}\left\{\phi_{\mathbf{l},\mathbf{i}} \mid \mathbf{i} \in \mathbf{I}_{\mathbf{l}}\right\}$ is a hierarchical increment space (or detail space) because of the property

$$W_{\mathbf{l}} = \operatorname{span}\left\{ B_{\mathbf{l}} \setminus \bigcup_{j=1}^{d} B_{\mathbf{l}-\mathbf{e}_j} \right\}$$

where $\mathbf{e}_j$ denotes the $j$-th unit vector and $B_{\mathbf{k}} := \emptyset$ for each $\mathbf{k} = (k_1, \ldots, k_d)$ with $k_j < 0$ for some $j = 1, \ldots, d$. Thus we get

$$V_{\mathbf{l}} = \bigoplus_{\mathbf{k} \leq \mathbf{l}} W_{\mathbf{k}} = \operatorname{span}\left\{\tilde{B}_{\mathbf{l}}\right\}$$

with the hierarchical basis

$$\tilde{B}_{\mathbf{l}} := \left\{\phi_{\mathbf{k},\mathbf{i}} \mid \mathbf{i} \in \mathbf{I}_{\mathbf{k}}, \mathbf{k} \leq \mathbf{l}\right\}.$$

Now, we can define the space of piecewise $d$-linear functions on the regular (isotropic) full grid

$$\Omega_t := \Omega_{(t,\ldots,t)} = \left\{\mathbf{x}_{\mathbf{k},\mathbf{i}} \mid |\mathbf{k}|_\infty \leq t, \mathbf{i} \in \mathbf{I}_{\mathbf{k}}\right\}$$

of level $t \in \mathbb{N}$ by

$$V_t := V_{(t,\ldots,t)} = \bigoplus_{|\mathbf{k}|_\infty \leq t} W_{\mathbf{k}}.$$

If

$$f_t = \sum_{|\mathbf{k}|_\infty \leq t} \sum_{\mathbf{i} \in \mathbf{I}_{\mathbf{k}}} f_{\mathbf{k},\mathbf{i}} \phi_{\mathbf{k},\mathbf{i}}$$

is the interpolant of $f \in H^2(\mathbf{H}_d)$ in $V_t$ it holds that

$$||f - f_t||_{L_2(\mathbf{H}_d)} = O\left(2^{-2t}\right). \tag{15}$$

Next, we define the regular sparse grid of level $t$ by

$$\Omega_t^s := \left\{\mathbf{x}_{\mathbf{k},\mathbf{i}} \mid n_d(\mathbf{k}) \leq t, \mathbf{i} \in \mathbf{I}_{\mathbf{k}}\right\} \tag{16}$$

and the corresponding function space by

$$V_t^s := \bigoplus_{\substack{\mathbf{k} \in \mathbb{N}^d \\ n_d(\mathbf{k}) \leq t}} W_{\mathbf{k}},$$

where $n_d(\mathbf{0}) := 0$ and

$$n_d(\mathbf{k}) := |\mathbf{k}|_1 - d + |\{m \mid \mathbf{k}_m = 0\}| + 1$$

for every other $\mathbf{k} \in \mathbb{N}^d$. Here, $|\mathbf{k}|_1 := \sum_{j=1}^{d} |\mathbf{k}_j|$ denotes the $\ell^1$ norm. This specific definition of $n_d$ guarantees that the resolution of grids on the boundary is the same as the resolution of grids in the interior of the domain.

If

$$f_t^s(\mathbf{x}) = \sum_{\substack{\mathbf{k}\in\mathbb{N}^d \\ n_d(\mathbf{k})\le t}} \sum_{\mathbf{i}\in\mathbf{I}_\mathbf{k}} \alpha_{\mathbf{k},\mathbf{i}} \phi_{\mathbf{k},\mathbf{i}}(\mathbf{x}) \in V_t^s$$

is the interpolant of $f \in H^2_{\text{mix}}(\mathbf{H}_d)$ in $V_t^s$, it holds that

$$||f - f_t^s||_{L_2(\mathbf{H}_d)} = O\left(2^{-2t} t^{d-1}\right).$$

Thus, compared to (15), the accuracy is only slightly worse by a factor $t^{d-1}$. However, the number of points in the full grid is $|\Omega_t| = O\left(2^{td}\right)$ and suffers from the curse of dimensionality for large $d$ whereas, in the sparse grid case, $M := |\Omega_t^s| = O\left(2^t \cdot t^{d-1}\right)$ holds and the exponential dependence of $d$ now only affects the level $t$ instead of $2^t$. For a thorough treatment of sparse grids, approximation results and complexity issues we refer to [1] and the references therein.

By solving (9) in the discrete space $V_t^s \subset \Gamma$ we get (analogously to (12))

$$\sum_{j=d}^{N-1} s_{j+1} \phi_{\mathbf{l},\mathbf{i}}\left(\mathbf{t}_j\right) = \sum_{\substack{\mathbf{k}\in\mathbb{N}^d : n_d(\mathbf{k})\le t, \\ \mathbf{m}\in\mathbf{I}_\mathbf{k}}} \alpha_{\mathbf{k},\mathbf{m}} \left( \sum_{j=d}^{N-1} \phi_{\mathbf{l},\mathbf{i}}\left(\mathbf{t}_j\right) \phi_{\mathbf{k},\mathbf{m}}\left(\mathbf{t}_j\right) + \eta h\left(\phi_{\mathbf{l},\mathbf{i}}, \phi_{\mathbf{k},\mathbf{m}}\right) \right) \tag{17}$$

for all $\mathbf{l} \in \mathbb{N}^d : n_d(\mathbf{l}) \le t$ and $\mathbf{i} \in \mathbf{I}_\mathbf{l}$.

A preconditioned multilevel conjugate gradient (pCG) algorithm is used to solve the linear system (17) iteratively. Here, for reasons of simplicity, we employ as preconditioner the inverse of the diagonal of the system matrix of (17) after its transformation to a prewavelet representation, see [15]. As we only need to implement matrix-vector-multiplications for the pCG algorithm, the system matrices are not assembled explicitly. The hierarchical structure and the compact support of our basis functions allow a fast application[8] of the first term in the brackets on the right hand side of (17) in $O\left(N \cdot t^d\right)$ operations. Because of the product structure of $H^1_{\text{mix}}$ an efficient implementation of the unidirectional principle can be employed for the on-the-fly multiplication of the term corresponding to the bilinear form $h$, see e.g. [4]. This needs $O(M)$ operations. Thus, the costs of a single iteration of the pCG algorithm are only $O\left(N \cdot t^d + M\right) = O\left((N \cdot t + 2^t) \cdot t^{d-1}\right)$ operations. For a detailed review of computational issues on the implementation of sparse grid methods, grid traversal strategies and linear system solvers, we refer to [4]. See Fig. 1 for two sparse grid examples.

[8] Note that the use of the combination technique [16] even allows here for a slight improvement to $O\left(N \cdot t^{d-1}\right)$. In both cases, however, the constant in the $O$-notation grows exponentially with $d$.

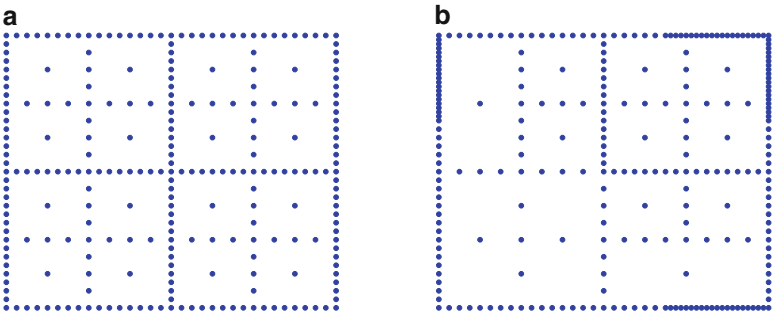


**Fig. 1** Different sparse grid examples in two dimensions. (**a**) Regular sparse grid of level 5. (**b**) Space-adaptive sparse grid

## *4.2 Space-Adaptive Sparse Grids*

Since most attractors only fill a sparse pattern of $\mathbf{H}_d$, it is obvious that a regular grid is not necessarily the best structure to approximate a function on such an attractor. On the one hand, there might not be enough grid points in relevant regions to fit the function which leads to bad approximations. On the other hand, there might be too many grid points in irrelevant areas which causes overfitting and results in an unnecessary high cost complexity, see [9] for a thorough treatment of this issue. One would prefer a grid which rather matches the shape of the trajectory than the ambient space $\mathbf{H}_d$. Such a grid (and of course the corresponding function space) can be derived using an iterative algorithm which adaptively creates finer grid resolutions where needed. The main component of such a procedure is an appropriate error indicator which decides if the grid has to be locally refined in a certain region. We here simply use

$$\epsilon_{\mathbf{l},\mathbf{i}} := \left|\left|\alpha_{\mathbf{l},\mathbf{i}}\phi_{\mathbf{l},\mathbf{i}}\right|\right|_{L_\infty(\mathbf{H}_d)} = \left|\alpha_{\mathbf{l},\mathbf{i}}\right|$$

as such an indicator. For more elaborate techniques and details on how to choose a reliable *and* efficient indicator $\epsilon_{\mathbf{l},\mathbf{i}}$ for the case of specific norms of the error, we refer to [13].

Our overall algorithm proceeds as follows: First, it starts with a regular sparse grid for some low level $\Omega^s_{\text{adp}} = \tilde{\Omega}^s_{\text{adp}} := \Omega^s_t$ and solves (17) on this grid. Then, it checks for each $\left\{(\mathbf{l},\mathbf{i}) \mid \mathbf{x}_{\mathbf{l},\mathbf{i}} \in \tilde{\Omega}^s_{\text{adp}}\right\}$ if $\epsilon_{\mathbf{l},\mathbf{i}} > \varepsilon$, where $\varepsilon \in \mathbb{R}^+$ is some fix threshold. If this is the case for the pair $(\mathbf{l},\mathbf{i})$ with odd $i_j$ or $i_j = 0$ for each $j \in \{1,\ldots,d\}$, all of its child nodes are inserted into the grid $\Omega^s_{\text{adp}}$ if they are not already contained.[9] In the one-dimensional case the child nodes are defined as

[9]Note here that it is not enough to check the surplus of points which have been inserted in the last iteration. The hierarchical surplus of all other points can change as well when calculating the solution on the refined grid.

$$\text{child}\,(x_{l,i}) := \begin{cases} \{x_{l+1,2i\pm1}\} & \text{if } l > 0, \\ \{x_{1,1}\} & \text{if } l = 0, i = 1, \\ \{x_{0,1}\} & \text{if } l = 0, i = 0. \end{cases} \tag{18}$$

In the multivariate case we define $\text{child}\left(\mathbf{x_{l,i}}\right)$ as

$$\left\{ \mathbf{x_{k,m}} \in \Omega_{\mathbf{k}} \;\middle|\; \begin{array}{l} \text{There exists } j \in \{1,\ldots,d\}, \text{ s.t. } x_{k_j,m_j} \in \text{ child}\left(x_{l_j,i_j}\right) \\ \text{and } k_h = l_h, m_h = i_h \text{ for all } h \in \{1,\ldots,d\} \setminus \{j\} \end{array} \right\}. \tag{19}$$

After the insertion it has to be guaranteed—by e.g. inserting further nodes where needed—that all hierarchical ancestors of every inserted point are contained in the resulting grid. Otherwise, an incorrect hierarchical basis representation for the corresponding function space would result and common grid traversal algorithms would run into problems. To achieve this we simply insert each missing direct ancestor and proceed recursively with the inserted points until each direct ancestor to every grid point has been inserted into $\Omega^s_{\text{adp}}$. The direct ancestors of points $\mathbf{x_{l,i}}$ with odd $i_j$ or $i_j = 0$ for each $j = \{1,\ldots,d\}$ are defined by

$$\text{directAnc}\left(\mathbf{x_{l,i}}\right) := \left\{ \mathbf{x_{k,m}} \in \Omega_{\mathbf{l}} \mid \mathbf{x_{l,i}} \in \text{ child}\left(\mathbf{x_{k,m}}\right) \right\}. \tag{20}$$

**Algorithm 1** The space-adaptive sparse grid algorithm

**Input:** starting level $t$, threshold $\varepsilon$, #iterations $L$, error indicators $\epsilon_{\mathbf{l,i}}$, time series $\left(s_j\right)_{j=1}^{N}$, embedding dimension $d$, regularization parameter $\lambda$
**Output:** space-adaptive sparse grid $\Omega^s_{\text{adp}}$

initialize: $\Omega^s_{\text{adp}} \leftarrow \Omega^s_t$, $\tilde{\Omega}^s_{\text{adp}} \leftarrow \Omega^s_t$, It $\leftarrow 0$
**while** It $< L$ **do**
  solve (17) on $\tilde{\Omega}^s_{\text{adp}}$
  **for all** $(\mathbf{k},\mathbf{m})$ with odd $m_j$ or $m_j = 0$ for each $j \in \{1,\ldots,d\}$ and $\mathbf{x_{k,m}} \in \tilde{\Omega}^s_{\text{adp}}$ **do**
    **if** $\epsilon_{\mathbf{k,m}} > \varepsilon$ **then**
      $\Omega^s_{\text{adp}} \leftarrow \Omega^s_{\text{adp}} \cup \text{ child}\left(\mathbf{x_{k,m}}\right)$
    **end if**
  **end for**
  **if** $\tilde{\Omega}^s_{\text{adp}} = \Omega^s_{\text{adp}}$ **then**
    **return** $\Omega^s_{\text{adp}}$
  **end if**
  $\tilde{\Omega}^s_{\text{adp}} \leftarrow \Omega^s_{\text{adp}}$
  **for all** $\mathbf{x_{k,m}}$ with odd $m_j$ or $m_j = 0$ for each $j \in \{1,\ldots,d\}$ and $\mathbf{x_{k,m}} \in \tilde{\Omega}^s_{\text{adp}}$ **do**
    $\Omega^s_{\text{adp}} \leftarrow \Omega^s_{\text{adp}} \cup \text{AllAncestors}(\mathbf{k}, \mathbf{m}, d)$
  **end for**
  $\tilde{\Omega}^s_{\text{adp}} \leftarrow \Omega^s_{\text{adp}}$
  It $\leftarrow$ It+1
**end while**
**return** $\Omega^s_{\text{adp}}$

**Algorithm 2** AllAncestors($\mathbf{l}$, $\mathbf{i}$, $d$)

**Input:** multivariate level $\mathbf{l}$, multivariate index $\mathbf{i}$, embedding dimension $d$
**Output:** set $X$ of all ancestors of $\mathbf{x}_{\mathbf{l},\mathbf{i}}$

initialize: $X \leftarrow \emptyset$
$X \leftarrow X \cup \text{directAnc}(\mathbf{x}_{\mathbf{l},\mathbf{i}})$
**for all** $\mathbf{x}_{\mathbf{k},\mathbf{m}} \in \text{directAnc}(\mathbf{x}_{\mathbf{l},\mathbf{i}})$ with odd $m_j$ or $m_j = 0$ for each $j \in \{1, \ldots, d\}$ **do**
    $X \leftarrow X \cup \text{AllAncestors}(\mathbf{k}, \mathbf{m}, d)$
**end for**
**return** $X$

When every relevant grid point of $\tilde{\Omega}^s_{\text{adp}}$ has been visited and treated accordingly, we set $\tilde{\Omega}^s_{\text{adp}} = \Omega^s_{\text{adp}}$ and start anew. This iteration runs until either no point needs to be refined or the number of iterations reaches some fixed limit $L \in \mathbb{N}$. A summary of the procedure can be found in Algorithm 1. For details on runtime and technical issues we refer to [4].

### *4.3 Dimension-Adaptive Sparse Grids*

In the case of attractors which fill a highly anisotropic part of the ambient space $\mathbf{H}_d$ or in case the ambient space dimension was overestimated, it is desirable to employ dimension-adaptive refinement instead of pure space-adaptive refinement. There, refinement takes place globally but only in directions which are relevant for the construction of a good forecasting function. Dimension-adaptivity for sparse grids has been introduced in [17]. The application of dimension-adaptive algorithms has been studied for integration in [10] and for approximation in [6, 7]. The approach which we use in the following is a little bit different though, it can be found in [4].

To motivate the idea of dimension-adaptive grids we will shortly review the concept of the ANOVA (Analysis of Variance) decomposition. We introduce a splitting

$$V = \mathbf{1} \oplus \mathcal{C} \tag{21}$$

of a space $V$ of univariate functions with domain $[0, 1]$ into the space of constant functions $\mathbf{1}$ and the remainder $\mathcal{C}$. This is done using the identity

$$f = P(f) + (f - P(f))$$

for some projector $P : V \to \mathbf{1}$ with $P|_{\mathbf{1}} = \text{id}$.

For multivariate tensor product function spaces $V$ we apply the splitting in every direction, i.e.

$$
\begin{aligned}
V &= \bigotimes_{i=1}^{d} V_i = \bigotimes_{i=1}^{d} (\mathbf{1}_i \oplus \mathcal{C}_i) \\
&= \mathbf{1}_1 \otimes \ldots \otimes \mathbf{1}_d \\
&\oplus \bigoplus_{i=1}^{d} (\mathbf{1}_1 \otimes \ldots \otimes \mathbf{1}_{i-1} \otimes \mathcal{C}_i \otimes \mathbf{1}_{i+1} \otimes \ldots \otimes \mathbf{1}_d) \\
&\oplus \bigoplus_{i=1}^{d} \bigoplus_{j=i+1}^{d} \left(\mathbf{1}_1 \otimes \ldots \otimes \mathbf{1}_{i-1} \otimes \mathcal{C}_i \otimes \mathbf{1}_{i+1} \otimes \ldots \otimes \mathbf{1}_{j-1} \otimes \mathcal{C}_j \otimes \mathbf{1}_{j+1} \otimes \ldots \otimes \mathbf{1}_d\right) \\
&\vdots \\
&\oplus \mathcal{C}_1 \otimes \ldots \otimes \mathcal{C}_d, \qquad (22)
\end{aligned}
$$

and receive a unique splitting of a function $f \in V$ into the sum of a constant function, $d$ univariate functions, $\frac{d(d-1)}{2}$ bivariate functions, and so on, i.e.

$$
f(x_1, \ldots, x_d) = f_0 + \sum_{i=1}^{d} f_i(x_i) + \sum_{i=1}^{d} \sum_{j=i+1}^{d} f_{ij}(x_i, x_j) + \ldots + f_{1,\ldots,d}(x_1, \ldots, x_d). \qquad (23)
$$

We call $f_0$ the ANOVA component of order 0, the $f_i$ are ANOVA components of order 1, and so on.
The most common choice for $P$ is

$$
P(f) := \int_{[0,1]} f(x) \mathrm{d}x
$$

for $V \subset L_2([0,1])$, which just gives the classical $L_2$-ANOVA decomposition. Another choice is

$$
P(f) := f(a)
$$

which leads to a well-defined decomposition if the point evaluation in $a$ is well-defined for all functions in $V$. This results in the so-called anchored ANOVA decomposition with anchor $a$. It is well suited to our piecewise linear basis functions.

Here, to transfer the concept of the multivariate anchored ANOVA decomposition to the piecewise linear hierarchical basis discretization, we have to change the index set introduced in (14) as in [6]. We define

$$
\tilde{\mathbf{I}}_{\mathbf{l}} := \left\{ \mathbf{i} \in \mathbb{N}^d \;\middle|\; \begin{array}{ll} i_j = 0, & \text{if } l_j = -1 \\ i_j = 1, & \text{if } l_j = 0 \\ 1 \le i_j \le 2^{l_j} - 1,\ i_j \text{ odd} & \text{if } l_j > 0 \end{array} \quad \text{for all } 1 \le j \le d \right\} \qquad (24)
$$

and allow the negative level $-1$. Furthermore, we define the one-dimensional basis function $\phi_{-1,0} := \chi_{[0,1]}$ to be the indicator function of the interval $[0, 1]$. With this and the definition

$$\tilde{W}_{\mathbf{l}} := \operatorname{span}\{\phi_{\mathbf{l},\mathbf{i}} \mid \mathbf{i} \in \tilde{\mathbf{I}}_{\mathbf{l}}\}$$

we see[10] that

$$\begin{aligned}\tilde{V}_{\mathbf{l}} := \bigoplus_{-\mathbf{1}\le\mathbf{k}\le\mathbf{l}} \tilde{W}_{\mathbf{k}} &= \bigoplus_{-\mathbf{1}\le\mathbf{k}\le\mathbf{l}} \operatorname{span}\{\phi_{\mathbf{k},\mathbf{m}} \mid \mathbf{m} \in \tilde{\mathbf{I}}_{\mathbf{k}}\} \\ &= \bigoplus_{\mathbf{0}\le\mathbf{k}\le\mathbf{l}} \operatorname{span}\{\phi_{\mathbf{k},\mathbf{m}} \mid \mathbf{m} \in \mathbf{I}_{\mathbf{k}}\} = \bigoplus_{\mathbf{0}\le\mathbf{k}\le\mathbf{l}} W_{\mathbf{k}} = V_{\mathbf{l}}\end{aligned}$$

for all $\mathbf{l}$ with $l_j \ge 0$ for all $j = 1, \ldots, d$. This way, we just have split the space of linear functions on $[0, 1]$, which was previously spanned by the two linear basis functions associated to the two boundary points, further into the sum of one constant (level $-1$) and one linear function (level 0). If we define the norm of a multivariate level index with possibly negative coordinates as

$$|\mathbf{l}| := |(\max(l_1, 0), \ldots, \max(l_d, 0))|$$

we can maintain our previous definition for sparse grids (16) using

$$\tilde{n}_d(\mathbf{k}) := \begin{cases} 0 & \text{if } k_j \le 0 \text{ for all } 1 \le j \le d \\ |\mathbf{k}|_1 - d + |\{m \mid \mathbf{k}_m \le 0\}| + 1 & \text{else} \end{cases}$$

instead of $n_d(\mathbf{k})$. But we now are able to identify functions which are constant in direction $j$ as they are elements of $\tilde{V}_{(l_1,\ldots,l_{j-1},-1,l_{j+1},\ldots,l_d)}$. This approach fits to a discretized anchored ANOVA decomposition with $a = 0$. To this end, we now define an infinite-dimensional univariate function space

$$V = \tilde{V}_{-1} \oplus \bigoplus_{i=0}^{\infty} \tilde{W}_i \tag{25}$$

and, with the choice $\mathbf{1}_i = \left(\tilde{V}_{-1}\right)_i$ and $\mathcal{C}_i = \left(\bigoplus_{j=0}^{\infty} \tilde{W}_j\right)_i$ in (21), we again obtain the splitting (22) which is now conform to the infinite-dimensional tensor product-hierarchical basis. In other words, if we use the alternative basis that is defined by the index set $\tilde{\mathbf{I}}_l$, the only univariate basis function $\psi$ for which $P(\psi) \neq 0$ is $\psi = \phi_{-1,0}$ for $P(f) := f(0)$ and the anchored ANOVA decomposition completely fits to the hierarchical tensor product basis.

So far, the subspaces of the ANOVA decomposition are (up to the very first one) still infinite-dimensional and need to be further discretized. To this end, for a

[10] Note that $W_{\mathbf{l}}$ and $\tilde{W}_{\mathbf{l}}$ are the same for a multilevel index $\mathbf{l}$ with $l_j \ge 1$ for all $j = 1, \ldots, d$.