Institut für Numerische Simulation
Rheinische Friedrich-Wilhelms-Universität Bonn

Wegelerstraße 6 • 53115 Bonn • Germany
phone +49 228 73-3427 • fax +49 228 73-7527
www.ins.uni-bonn.de

Chr. Feuersänger, M. Griebel

Principal Manifold Learning by Sparse Grids

INS Preprint No. 0801

April 2008

Principal manifold learning by sparse grids

Christian Feuersänger, Michael Griebel
Institute for Numerical Simulation, University of Bonn

Abstract

In this paper we deal with the construction of lower-dimensional manifolds from high-dimensional data which is an important task in data mining, machine learning and statistics. Here, we consider principal manifolds as the minimum of a regularized, non-linear empirical quantization error functional. For the discretization we use a sparse grid method in latent parameter space. This approach avoids, to some extent, the curse of dimension of conventional grids like in the GTM approach. The arising non-linear problem is solved by a descent method which resembles the expectation maximization algorithm. We present our sparse grid principal manifold approach, discuss its properties and report on the results of numerical experiments for one-, two- and three-dimensional model problems.

AMS Subject Classification: 65N30, 65F10, 65N22, 41A29, 41A63, 65D15, 65D10.
Key words: sparse grids, regularized principal manifolds, high-dimensional data.

1 Introduction

The reconstruction of lower-dimensional manifolds from high-dimensional data is an important task in data mining, machine learning and statistical learning theory. It offers a powerful framework for nonparametric dimension reduction and has many practical applications in e.g. speech and image processing, sonification, or process monitoring. The key idea is to find the most succinct low dimensional structure that is embedded in a higher dimensional space. With its help, algorithms can work directly in the lower dimensional latent space of the manifold instead of the high-dimensional data space and thus get computationally feasible. Applications range from clustering over feature extraction to recognition tasks. Here, besides the classical principal component analysis (PCA) [45], various non-linear local and non-local methods have been developed in the recent decade. Popular approaches are, among others, multidimensional scaling (MDS), Kohonen's SOM, generative topological mapping (GTM), locally linear embedding (LLE), Isomap, Laplacian eigenmaps, Hessian eigenmaps, local tangent space alignment (LTSA), curvilinear distance analysis (CDA), diffusion wavelets, auto-associative neural networks, Kernel PCA, nonlinear principal component analysis and regularized principal manifolds. For a survey on these techniques and potential applications, see [21, 43] and the references cited therein, as well as the web pages [1, 2] and the links therein.[1]

To this end, the lower-dimensional manifold has to be modeled properly, either explicitly or implicitly. Here, often radial basis approaches are used where kernel functions are attached to the data points. Then, the corresponding algorithms scale in general cubically with the number of data points. This allows to deal with sets of high dimensionality but limits the applications to a moderate amount of data. Alternatively, if an approximation of the manifold is explicitly represented by some sort of parametrization, grids are employed. This way, principal curves and surfaces can be constructed by polygonal line algorithms. Also Kohonen's SOM and the generative topographic mapping use a grid in latent space. Then, quite large data sets can be dealt with. But, due to the curse of dimensionality, the dimension of the manifold is restricted to at most three (or four) which limits the applicability of these methods to some extent. In general, just two-dimensional grids are presently used in practice.

In this paper, we propose to represent the manifold parametrically and to approximate the component functions of the parametric mapping using a so-called sparse grid instead of a conventional grid. Sparse grids are based on tensor products of one-dimensional multiscale functions. The coefficients in the resulting

[1] Note that such dimensionality reduction algorithms can often be formulated in terms of what they preserve about the original data. Some, like PCA preserve variance, others, like MDS or Isomap large distances (metric, non-metric or geodesic), others like SOM or LLE preserve nearby neighbours.

multivariate series representation of a sufficiently smooth mapping function then exhibit a specific decay with the number of levels involved. For certain function classes, i.e. for d-dimensional functions with dominating r-th mixed derivatives, truncation of the associated series expansion results in sparse grid spaces which need only $O(m \log(m)^{d-1})$ degrees of freedom instead of $O(m^d)$ degrees of freedom for the case of uniform full grids, see [18] and the references cited therein. Here, m denotes the number of grid points in one coordinate direction. With $h \sim 1/m$, the achieved accuracy is however only slightly reduced from $O(h^r)$ to $O(h^r(\log h^{-1})^{d-1})$ in the L^2-norm if piecewise polynomials of degree $r-1$ are used in the underlying one-dimensional multilevel basis. With respect to the energy norm even the same order $O(h^{r-1})$ of accuracy can be obtained for both cases. For a general survey on the sparse grid method and its various variants and applications, see [18].

These properties make the sparse grid technique a good candidate for manifold reconstruction problems. To this end, a vector-valued version of the sparse grid approach is employed for the finite-dimensional approximation of the component functions of the parametric representation of a manifold. This representation is determined as the sparse grid solution of a non-linear minimization problem which involves an empirical quantization error measuring the distance of the manifold from the given data points and a regularization term incorporating the smoothness assumption on the manifold. This approach is closely related to regularization networks, see [28] and compare especially [60]. The solution of the resulting discrete non-linear problem is computed by a descent method. It turns out that sparse grids allow to reconstruct manifolds in a more cost effective way than the conventional full grids that are commonly employed in, e.g., the GTM approach. Furthermore, they open a way to deal with higher-dimensional manifolds than just two- or three-dimensional ones.

The remainder of this paper is organized as follows. In section 2 we state the manifold reconstruction task as the minimization of a regularized empirical error functional. In section 3 we present the discretization of the problem in a general finite dimensional space and discuss a descent method similar to the expectation maximization algorithm (EM) as a way to locally solve the resulting nonlinear system. Then, in section 4 we introduce sparse grids for the approximate representation of general manifolds and use them in the discretization of the regularized empirical error functional. It turns out that the sparse grid approach and the corresponding algorithm scales favorably with respect to both the number of data points given and the number of degrees of freedom involved in the discretization. In section 5, we present the results of numerical experiments using our sparse grid manifold reconstruction approach. Here, besides principal curves, we deal with principal surfaces and more general principal manifolds. We compare the sparse grid approach to the conventional full grid method and discuss its properties for typical model problems. Finally, we give some conclusions in section 6.

2 The problem

Given are N data points $\{x_1, \ldots, x_N\} \subset X = \mathbb{R}^n$ which we assume to be drawn iid from an unknown underlying probability distribution $P(x), x \in X$. We define an index set T, e.g. $\mathbb{R}^d$, and consider the maps $f : T \to X$, and a class $\mathcal{F}$ of such maps with e.g. additional properties to be fixed later on. The aim is now to find a map $f \in \mathcal{F}$ such that the quantization error

$$R(f) = \int_X \min_{t \in T} c(x, f(t)) \; dP(x)$$

is minimized in $\mathcal{F}$. Here, c denotes a loss function which is typically chosen as $c(x, f(t)) = \|x - f(t)\|_2^2$. Of course, this problem is unsolvable since the probability distribution $P(x)$ is not known. Therefore, we replace $P(x)$ by the empirical density

$$P_N(x) = \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i)$$

and minimize the empirical quantization error

$$\int_X \min_{t \in T} \|x - f(t)\|_2^2 \; dP(x) \approx \frac{1}{N} \sum_{i=1}^{N} \min_{t \in T} \|x_i - f(t)\|_2^2 =: R_{emp}(f) \tag{2.1}$$

in $\mathcal{F}$.

Note that various situations with finite and infinite sets T can be described in this framework, compare also [58]. In case of a finite T, this involves codes with discrete quantization. For example, if $T = \{1\}$, $f(1) \in X$ and $\mathcal{F}$ the set of constant functions, we obtain the sample mean as result of the minimization problem. If $T = \{1, \ldots, k\}$, $f : i \to f_i$, $f_i \in X$ and $\mathcal{F}$ the set of associated functions, we obtain the distortion error of a vector quantizer for which a local minimum can be found by the well-known k-means algorithm.

Moreover, for infinite T interesting applications may be modelled as well: Then, instead of discrete quantization, a mapping onto a manifold of dimensionality lower than the input space can be considered. For example, if $T = \mathbb{R}$, $f : t \to f_0 + tf_1$, $f_0, f_1 \in X$, $\|f_1\| = 1$ and $\mathcal{F}$ the space of all such line segments, we obtain from the minimization of (2.1) over $\mathcal{F}$ the line parallel to the direction of the largest variance in P which just resembles the well-known principal component analysis (PCA), see [45] for further details. An example is given in Figure 2.1 (left).

Figure 2.1: Linear and nonlinear principal component analysis. First component of linear PCA (left) and of nonlinear PCA, i.e. principal curve (right).

Furthermore, for $T = [0, 1]^d$, $f : t \to f(t) = (f_{(1)}(t), \ldots, f_{(n)}(t))$, $f \in \mathcal{F}$ where $\mathcal{F}$ is the class of n-tuples of continuous $\mathbb{R}^d$-valued functions, we obtain with $d = 1$ the so-called principal curve problem [42] which is a nonlinear generalization of PCA. An example is given in Figure 2.1 (right). A further discussion and results on various versions of principal curves can be found in [19, 22, 47, 48, 56, 63]. Finally, in the case $d > 1$, general principal surfaces and principal manifolds [10, 41, 60, 58] are modelled which are an instance of nonlinear principal component analysis, compare also [25, 27, 51] for related approaches. An example is shown in Figure 2.2.

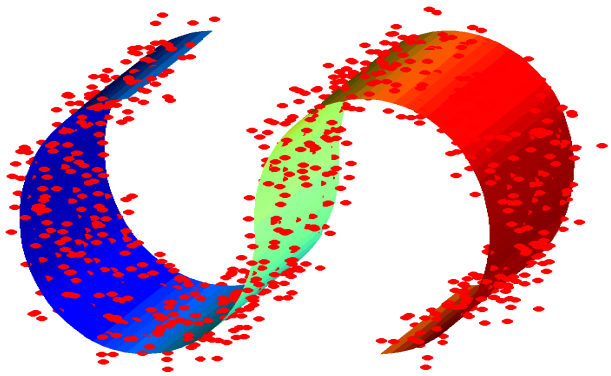

Figure 2.2: Nonlinear principal manifold, $d = 2, n = 3$.

Note that we encounter here a nonlinear problem due to the general choice of f. Note furthermore that the minimization of the associated functional (2.1) on $\mathcal{F}$ is now an ill-posed problem. To nevertheless obtain a well-posed (nonlinear) problem one usually employs some sort of regularization. To this end, we consider

$$R_{reg}(f) = R_{emp} + \lambda S(f) \tag{2.2}$$

where R_{emp} again denotes the empirical error functional (2.1), $S(f)$ is a smoothing functional which enforces a certain regularity on f and $\lambda \in \mathbb{R}^+$ denotes the regularization parameter which balances the two terms.

Depending on the problem under consideration, there are many ways to choose the smoothing term $S(f)$ in practical applications. First of all, S should be a convex, non-negative functional of f. For example, for

the construction of principal curves, i.e. for the case $d = 1$, S can be geometrically chosen and interpreted as a length constraint[2] on the curve f:

$$S(f) = \|Gf\|^2_{L_2(T,X)} = (Gf, Gf) = \sum_{i=1}^{n} (Gf_{(i)}, Gf_{(i)}) \text{ with e.g. } G = \nabla = \partial_t. \tag{2.3}$$

A similar relation between derivatives of f and geometrical interpretations exists for surfaces, i.e. for the case $d = 2$. A standard result from differential geometry states that

$$\text{surf}(f) = \int_T \sqrt{\left(\frac{\partial}{\partial t_1} f\right)^2 \left(\frac{\partial}{\partial t_2} f\right)^2 - \left[\left(\frac{\partial}{\partial t_1} f\right)^T \left(\frac{\partial}{\partial t_2} f\right)\right]^2}\, dt \tag{2.4}$$

is the surface area of a (sufficiently smooth) parameterized surface, compare with [46]. A geometric/ arithmetic mean argument shows that

$$S(f) = \frac{1}{2} \int_T \left(\frac{\partial}{\partial t_1} f\right)^2 + \left(\frac{\partial}{\partial t_2} f\right)^2 dt = \frac{1}{2} \sum_{i=1}^{n} (Gf_{(i)}, Gf_{(i)}) \text{ again with } G = \nabla \tag{2.5}$$

is an upper bound for surf(f). The use of constraints on area and volume, curvature and higher order derivatives make sense and is subject of actual research. To this end, we use a sum of length constraints like (2.3) for the manifold edges and surface constraints like (2.5) for surface elements. The associated choice of squared gradients leads to a (squared) weighted generalized[3] version of the variation of Hardy and Krause. For a definition of the variation of Hardy and Krause and further details on it, see e.g. [54]. We set

$$S(f) = \sum_{i=1}^{n} V_{HK}^{(U,V,W)}(f_{(i)}) \tag{2.6}$$

with

$$V_{HK}^{(U,V,W)}(f_{(i)}) = \sum_{\substack{u \in U \\ v \in V(u)}} w_{u,v} \left(\int_{[\mathbf{0}^{\bar{u}}, \mathbf{1}^{\bar{u}}]} \left(\partial^v f_{(i)}(t; \mathbf{0}^u)\right)^2 dt^{\bar{u}} + \int_{[\mathbf{0}^{\bar{u}}, \mathbf{1}^{\bar{u}}]} \left(\partial^v f_{(i)}(t; \mathbf{1}^u)\right)^2 dt^{\bar{u}} \right) \tag{2.7}$$

where, for a vector $t \in \mathbb{R}^d$, non-empty[4] u and $u \subset \{1, \ldots, d\}$, the expression t^u denotes the components u of t, i.e. all components $\{1, \ldots, d\} \setminus u$ are eliminated. Here, $\mathbf{0} = (0, \ldots, 0)^T$, $\mathbf{1} = (1, \ldots, 1)^T \in \mathbb{R}^d$, denote the two diagonal corners of T, respectively. Furthermore, $f_{(i)}(\cdot; \mathbf{b}^u)$ means that $f_{(i)}$ is only evaluated at points t where $t_j = b_j$ for $j \in u$, and

$$\partial^v f_{(i)}(t) := \frac{\partial^{|v|}}{\prod_{j \in v} \partial t_j} f_{(i)}(t)$$

is the partial derivative of $f_{(i)}$ taken once with respect to each t_j for $j \in v$. Finally, $\bar{u} := \{1, \ldots, d\} \setminus u$. Here, the sets U, V and the weights W are given. U determines a set of *fixed* directions (which define slices $\mathbf{0}^u$ and $\mathbf{1}^u$ with $u \subset U$). The sets $V(u)$ define derivative directions. Since any derivatives in directions $i \in u$ vanish identically[5], each $v \in V(u)$ should be a subset of $\bar{u}$.

Formula (2.7) becomes clearer if we consider two examples: For $U = \{\emptyset\}$, $V(\emptyset) = \{v : |v| = 1\}$, $W = \{w_{u,v} = 1\}$ we just obtain (2.3). For $d = 2$, if we set $U = \big\{\emptyset\big\} \cup \big\{|u| = 1\big\} = \big\{\emptyset, \{1\}, \{2\}\big\}$,

[2] To be precise, $\|\partial_t f\|^2_{L_2(T,X)} = \int \dot{f}^2_{(1)} + \dot{f}^2_{(2)} + \ldots + \dot{f}^2_{(n)} dt$ is an integral over the squared speed of the curve f. Then, since a re-parametrization of f to constant speed does not change R_{reg} but minimizes the regularization term, $\|\partial_t f\|^2_{L_2(T,X)}$ equals the squared *length* of the curve at the optimal solution [60].

[3] Note that the classical variation in the sense of Hardy and Krause integrates absolute values, uses just $V(u) = \{\bar{u}\}$ and only employs $\mathbf{0}$ in its construction. It also does not employ weights w_{uv}.

[4] Note that for $u = \emptyset$ both integral terms would be equal. Then we retain just one of them.

[5] Note that a definition with V formally independent of u would lead to the same result – the additional terms would vanish due to this property.

$V(u) = \{v \subset \bar{u},\, |v| = 1\}$ and $W = \{w_{u,v} = 1\}$, we obtain

$$\underbrace{\int_0^1 \int_0^1 (\partial_{t_1} f_{(i)}(t_1,t_2))^2\, dt_1\, dt_2}_{u=\emptyset,\, v=\{1\}} + \underbrace{\int_0^1 \int_0^1 (\partial_{t_2} f_{(i)}(t_1,t_2))^2\, dt_1\, dt_2}_{u=\emptyset,\, v=\{2\}} + \underbrace{\int_0^1 (\partial_{t_1} f_{(i)}(t_1,0))^2\, dt_1}_{u=\{2\},\, v=\{1\}}$$
$$+ \underbrace{\int_0^1 (\partial_{t_1} f_{(i)}(t_1,1))^2\, dt_1}_{u=\{2\},\, v=\{1\}} + \underbrace{\int_0^1 (\partial_{t_2} f_{(i)}(0,t_2))^2\, dt_2}_{u=\{1\},\, v=\{2\}} + \underbrace{\int_0^1 (\partial_{t_2} f_{(i)}(1,t_2))^2\, dt_2}_{u=\{1\},\, v=\{2\}}. \quad (2.8)$$

In the general case, we may choose S as a prior assumption on the function class of the reconstructed manifold. To be more precise, we assume f to live in a certain function space $\mathcal{F} = \{f \in L_2(T,X) : \|f\| \leq c < \infty\}$ with associated norm $\|\cdot\|$. This space might be a subspace of $\mathcal{H}$ which is a reproducing kernel Hilbert space [4, 64]. Then, $S(f)$ just corresponds to $\|f\|^2_{\mathcal{H}}$. In other words, we minimize the empirical quantization error (2.1) under the side constraint $\|f\|_{\mathcal{H}} = c$. The Lagrange approach then results (up to a constant term) in (2.2) with λ being the Lagrange multiplier. In the simple case of Sobolev spaces and related function spaces, we have $S(f) = \|Gf\|^2_{L_2(T,X)}$ with G a specifically chosen differential operator which expresses the additional regularity of f. In the general case of a reproducing kernel Hilbert space with associated kernel $k(\cdot,\cdot)$, the corresponding G is no longer a differential operator but merely a pseudo-differential operator. For example, for the widely used kernel $k(x,y) = \exp(-\|x-y\|_2^2/(2\sigma^2))$, we have by means of Fourier analysis

$$\|Gf\|^2_{L_2(T,X)} = \int \sum_{j=0}^{\infty} \frac{\sigma^{2j}}{j!2^j} (D^j f(x))^2 dx$$

with $D^{2j} = \Delta^j$ and $D^{2j+1} = \nabla\Delta^j$. Note that the relation between a representation with (2.3) and a representation with the associated reproducing kernel is often given via some sort of representer theorem, see e.g. [49, 59]. There is a wide range of possible kernels, a further discussion on kernels and their relation to smoothing operators can be found in [58] and the references cited therein.

3 Discretization and solution

Now we choose a countable basis $\{\phi_j(t)\}, j = 1, \ldots, \infty, \phi_j : t \to \mathbb{R}$ of $\mathcal{F}$, and expand f as infinite series in this basis, i.e.

$$f(t) = \sum_{j=1}^{\infty} \alpha_j \cdot \phi_j(t)$$

with coefficient vector $\alpha_j = (\alpha_{j,1}, \ldots, \alpha_{j,n})^T \in \mathbb{R}^n$. Note that the multiplication $\alpha_j \cdot \phi_j(t)$ has to be understood component-wise, i.e.

$$\alpha_j \cdot \phi_j(t) = (\alpha_{j,1}\phi_j(t), \ldots, \alpha_{j,n}\phi_j(t))^T \quad \text{and} \quad f_{(i)}(t) = \sum_{j=1}^{\infty} \alpha_{j,i}\phi_j(t), i = 1, \ldots, n.$$

We thus employ the same basis function set for each component of f. We can then reformulate our problem (2.2) as follows: Find the minimum of

$$\underset{\substack{t_1,\ldots,t_N \in T \\ \alpha_1,\ldots,\alpha_\infty}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \|x_i - f(t_i; \vec{\alpha}_\infty)\|_2^2 + \lambda \|Gf(\cdot; \vec{\alpha}_\infty)\|^2_{L_2(T,X)}. \quad (3.1)$$

The infinite parameter vector $\vec{\alpha}_\infty = (\alpha_1^T, \ldots, \alpha_\infty^T)^T$ collects here all the component vectors $\alpha_j, j = 1, \ldots, \infty$. It enters the notation of the function f in a parametric way to indicate its dependence on the coefficients $\alpha_j, j = 1, \ldots, \infty$. Note that the interior minimization $\min_{t \in T} \|x_i - f(t)\|_2^2$ from (2.1) is now translated into N independent minimizations (one for each x_i) and can thus be written in front of the sum.

So far, (3.1) is a non-linear minimization problem with an infinite dimensional search space (in $\vec{\alpha}_\infty$) which is not computationally feasible yet. To allow for a numerical solution we have to resort to some sort of discretization. To this end, we restrict ourselves to a finite dimensional subset $\mathcal{F}_M \subset \mathcal{F}$ with finite basis, i.e. $\text{span}\{\phi_1, \ldots, \phi_M\} = \mathcal{F}_M$. This leads to a finite-dimensional approximation f_M of f, i.e.

$$f(t) \approx f_M(t; \vec{\alpha}_M) = \sum_{j=1}^{M} \alpha_j \cdot \phi_j(t) \tag{3.2}$$

with associated coefficient vector $\vec{\alpha}_M = (\alpha_1^T, \ldots, \alpha_M^T)^T \in (\mathbb{R}^n)^M$ and to the following finite-dimensional problem: Find the minimum of

$$\underset{\substack{t_1,\ldots,t_N \in T \\ \alpha_1,\ldots,\alpha_M}}{\text{argmin}} \frac{1}{N} \sum_{i=1}^{N} \|x_i - f_M(t_i; \vec{\alpha}_M)\|_2^2 + \lambda \|G f_M(\cdot; \vec{\alpha}_M)\|_{L_2(T,X)}^2. \tag{3.3}$$

Here, to obtain fast convergence of f_M towards f the choice of a proper sequence of bases for the sequence of function spaces $\{\mathcal{F}_M\}_M$ and a certain smoothness of f is important. A solution of the discrete nonlinear minimization problem (3.3) can then be gained by a conventional descent method which is closely related to the well-known expectation maximization algorithm [24]:

1. Choose some initial values, for example as the result of a PCA of the given data $\{x_i\}$.
2. Projection step: Keep $\{\alpha_j, j = 1, \ldots, M\}$ fixed and minimize with respect to $\{t_i, i = 1, \ldots, N\}$:

$$\underset{t_i}{\text{argmin}} \|x_i - f_M(t_i; \vec{\alpha}_M)\|_2^2, \quad i = 1, \ldots, N. \tag{3.4}$$

   To this end, N different, decoupled non-linear minimization problems of size d must be solved.[6]
3. Adaption step: Keep $\{t_i, i = 1, \ldots, N\}$ fixed and minimize with respect to $\{\alpha_j, j = 1, \ldots, M\}$:

$$\underset{\alpha_1,\ldots,\alpha_M}{\text{argmin}} \frac{1}{N} \sum_{i=1}^{N} \|x_i - f_M(t_i; \vec{\alpha}_M)\|_2^2 + \lambda \|G f(\cdot; \vec{\alpha}_M)\|_{L_2(T,X)}^2. \tag{3.5}$$

   Note that this is just a vector-valued regression problem with the data (t_i, x_i), $i = 1, \ldots, N$. Now, since we assumed that G acts componentwise on f, differentiation with respect to $\alpha_j, j = 1, \ldots, M$ results in n linear systems of equations

$$(B^T B + N\lambda \cdot C)\vec{\alpha}_M^{(k)} = B^T \vec{x}^{(k)}, \quad k = 1, \ldots n, \tag{3.6}$$

   where B denotes the $N \times M$ matrix with entries $B_{ij} = \phi_j(t_i)$ and C denotes the $M \times M$ matrix with entries $C_{ij} = \int G\phi_i(t) G\phi_j(t) dt$. Here, the data vector $\vec{x}^{(k)}$ consists of the k-th coordinates of the data points x_i, i.e. $\vec{x}^{(k)} = (x_{1,k}, \ldots, x_{N,k})^T$ as does the unknown vector $\vec{\alpha}_M^{(k)} \in \mathbb{R}^M$, i.e. $\vec{\alpha}_M^{(k)} = (\alpha_{1,k}, \ldots, \alpha_{M,k})^T$, $k = 1, \ldots, n$. We thus have to solve the same system with n different right hand sides.

In analogy to the expectation maximization algorithm, one iteration of the steps 2 and 3 does not increase the value of the target function and successive iteration will eventually converge to a local minimum of (3.3). Although also other optimization methods like Newton's approach may need less iterations, our problem-adapted decomposition into projection and adaption reduces the overall computational complexity significantly.

So far, we were not specific about the choice of the smoothing operator S nor on the choice of the basis functions ϕ_j. In the case $d = 1$, i.e. for principal curves, a natural choice for the regularization operator

[6] In principle, this can be achieved by any standard nonlinear minimization method which allows for jumps in the derivative, like e.g. the downhill simplex approach or the Max-Powell method [53]. In the following, we employ the piecewise linear structure of our basis functions and use a domain decomposition approach to identify smooth parts of $\|x_i - f_M(t; \vec{\alpha}_M)\|_2^2$ where we then use a (local) Newton type method to find the minimum with a few iterations.

is the constraint of a fixed curve length. This translates to $S(f) = \|\nabla f\|^2_{L_2(T,X)} = \sum_{i=1}^n \|\dot{f}_{(i)}\|^2_{L_2}$, see [48]. Furthermore, f is approximated by a polygonal line f_M which is spanned by M points. In the case $d = 2$ and $d = 3$, a natural extension would be a smoothing operator S like (2.6) which relates to the area and volume of the manifold, respectively.

Furthermore, a 2- or 3-dimensional mesh of points may be used to span the approximand f_M. But in case of a general d, the degrees of freedom involved in a uniform mesh behave as $M = O(m^d)$ where m denotes the number of points in one coordinate direction of the mesh. Here, the curse of dimension shows up, i.e. the number of degrees of freedom scale exponentially with the dimension d. Thus, for $d > 4$ such an approach gets impossible due to the huge number of degrees of freedom involved.

Another approach is to rely on the theory of reproducing kernel Hilbert spaces (RKHS). To this end, the function to be found is assumed to belong to a RKHS $\mathcal{H}$. Then, the smoothing operator is chosen as $S(f) = \|f\|^2_{\mathcal{H}}$ where $\|.\|_{\mathcal{H}}$ denotes the norm associated to the RKHS $\mathcal{H}$, for details see [4, 64]. Here, an associated kernel function $k(x, x')$ uniquely determines the RKHS $\mathcal{H}$. In the kernel approach, M points $q_j \in T, j = 1, \ldots, M$ are chosen and to each point the kernel function is attached accordingly. We thus have $\phi_j(t) = k(q_j, t)$ and the associated finite expansion reads

$$f_M(t) = \sum_{j=1}^{M} \alpha_j \cdot k(q_j, t). \tag{3.7}$$

A solution of the corresponding discrete nonlinear minimization problem (3.3) can be gained by the above-mentioned descent algorithm. The resulting linear system in the projection step 2 again reads as (3.6), where now the matrix B contains the values $B_{ij} = k(q_j, t_i)$ and the matrix C is just $C_{ij} = k(q_j, q_i)$.

The overall costs of the kernel approach are then as follows: The projection step involves $O(NM)$ operations[7] which is due to the globality of the kernel. The setup of the matrices in the adaption step needs $O(M^2N)$ operations, and the solution of the linear system (3.6) involves $O(M^3)$ operations since the matrix is usually full due to the globality of the kernel k. Altogether we see that this approach scales linear in the number of data but it scales cubic in the number of parameters α_j. Thus only a moderate number M of parameters can be employed in such a model. Note furthermore that a good choice of the points q_j is not straightforward and, moreover, associated to this question, neither the convergence of f_M to f nor its convergence rate is completely clear.

Interestingly, the so-called generative topographic mapping (GTM) method can be reinterpreted as a variant of the kernel based discretization involving a grid based approach. It was introduced in [10] as a probabilistic reformulation of the self-organizing map (SOM) and got further developed in [11] and [19, 20]. The GTM is a probability density model which describes the distribution of data in high-dimensional space in terms of a smaller number of latent variables using a *uniform* grid of points q_j in latent space T. Here, the mesh points are equipped with non-linear basis functions $\phi_j(t)$ which might be Gaussians or sigmoidal functions and f_M is again spanned as linear combination (3.2) so that each point t in latent space T is mapped to a corresponding point x in the n-dimensional data space X. We may again write $f_M(t)$ as (3.7). Now, if we denote the node locations in T by $t_{j'}, j' = 1, \ldots, M$, then (3.7) defines a corresponding set of vectors

$$z_{j'} = f_M(t_{j'}). \tag{3.8}$$

Each of these vectors forms the center of an isotropic Gaussian distribution in data space, whose inverse variance we denote by β, such that

$$p(x|j') = \left(\frac{\beta}{2\pi}\right)^{n/2} \exp(-\frac{\beta}{2}\|z_{j'} - x\|_2^2). \tag{3.9}$$

The probability density function for the GTM model is then obtained by summing over all of the Gaussian components, i.e.

$$p(x|\vec{\alpha}_M, \beta) = \sum_{j'=1}^{M} P(j')p(x|j') = \sum_{j'=1}^{M} \frac{1}{M}\left(\frac{\beta}{2\pi}\right)^{n/2} \exp(-\frac{\beta}{2}\|z_{j'} - x\|_2^2)$$

[7] Here we assume a constant number of iteration steps to achieve a local minimum.

where we have taken the prior probabilities $P(j')$ of each of the components j' to be constant and equal to $1/M$. Note that due to (3.8), $z_{j'} = z_{j'}(\vec{\alpha}_M)$ which has been omitted to simplify the notation. Altogether, the GTM model is just a constraint mixture of Gaussians with adaptive parameters α_j and β where the Gaussian distribution (3.9) represents a noise model.

Furthermore, for further regularization, a prior over the class of mappings f is needed. In [10] a Gaussian prior over the parameters α_j was employed, i.e. $P(\vec{\alpha}_M) = \prod_j^M \left(\frac{\beta}{2\pi}\right)^{n/2} \exp(-\frac{\beta}{2}\|\alpha_j\|_2^2)$. This approach depends strongly on the number of basis functions $\phi_j(t) = k(q_j, t)$ and easily results in overfitting. To overcome this problem a Gaussian process prior

$$P(\vec{\alpha}_M) = (2\pi)^{-n/2}|k|^{1/2} \exp\left(-\sum_{j,j'} \alpha_j \cdot \alpha_{j'} k(q_j, q_{j'})\right)$$

was introduced in [11]. Then, since a parametric probability density model $p(x|\vec{\alpha}_M, \beta)$ can be fitted to a data set $\{x_1, \ldots, x_N\}$ by maximum likelihood, we obtain the log likelihood function

$$L(\vec{\alpha}_M, \beta) = \sum_{i=1}^{N} \ln p(x_i|\vec{\alpha}_M, \beta) + \ln P(\vec{\alpha}_M) + C$$

after taking the log posterior probability and exploiting the i.i.d. assumption on the data set. With (3.7) and (3.8), maximization with respect to $\vec{\alpha}_M$ just results in the discretization (3.3) of (2.2) with the choice $S(f) = \|f\|^2_{\mathcal{H}(k)}$ where $\|.\|_{\mathcal{H}(k)}$ denotes the norm associated to the RKHS $\mathcal{H}(k)$ associated to the kernel $k(\cdot,\cdot)$ and k is the Gaussian. Here, the parameter $\beta/2$ can be absorbed into the regularization parameter λ and the forefactors $(\beta/(2\pi))^{n/2}$ and $(2\pi)-n/2|k|^{1/2}$ enter the constant C which plays no role after maximization of $L(\vec{\alpha}_M, \beta)$ anyway. For further details, see [11] and the discussion in [58], sections 17.4.1. and 17.4.2.

The latent space of the GTM is generally chosen to have a low dimensionality (typically d=2). Although it is straightforward to formulate the GTM for latent spaces of any dimension, the model becomes computationally intractable if d gets large. The reason is the curse of dimensionality, i.e. the number of nodes in the grid grows exponentially with d (as does the number of basis functions). Note that the same problem arises with the SOM. While there are attempts to use random sampling in latent space or to apply semi-linear models to face that problem [11], such methods do not really cure it.

4 Sparse grids

To avoid the above-mentioned problems with the curse of dimension on one hand and with the cubic scaling in the number of parameters on the other hand, we suggest to employ the so-called sparse grid approach for the discretization of f.

4.1 Construction and properties

Sparse grid spaces were originally developed for the efficient discretization of d-dimensional elliptic problems of second order. They are based on tensor products of one-dimensional multiscale functions. The coefficients of a sufficiently smooth solution in the resulting multivariate series representation then exhibit a specific decay with the number of levels involved. For certain function classes, i.e. for functions with dominating r-th mixed derivatives, truncation of the associated series expansion results in sparse grid spaces which need only $O(m\log(m)^{d-1})$ degrees of freedom instead of $O(m^d)$ degrees of freedom for the case of uniform full grids, see [18] and the references cited therein. Here, m denotes the number of grid points in one coordinate direction. With, $h \sim 1/m$, the achieved accuracy, however, is only slightly reduced from $O(h^r)$ to $O(h^r(\log h^{-1})^{d-1})$ in the L^2-norm if piecewise polynomials of degree $r-1$ are used in the basic one-dimensional multilevel basis. With respect to the energy norm even the same order of accuracy can be obtained for both cases. Furthermore there are so-called energy-norm based sparse grids which only need $O(m)$ degrees of freedom but result in $O(h^{r-1})$ accuracy with respect to the energy norm. This approach completely eliminates the dependence of the dimension d in the complexities at least for the m-asymptotics, the order constants however still depend exponentially on d. The sparse grid method has successfully

been applied to problems from quantum mechanics [30], to stochastic differential equations [57], to high-dimensional integration problems from physics and finance [12, 32, 55] and to the solution of moderately higher-dimensional partial differential equations, mainly of elliptic type [6, 7, 16]. For a survey, see [18].

These properties make the sparse grid technique a good candidate for manifold reconstruction problems. To this end, a vector-valued version of the sparse grid approach is employed for the finite-dimensional approximation of f in (3.2), where each coordinate function is represented in the same sparse grid basis. In this subsection, we first present the construction principle of sparse grids and their properties for the case of scalar functions for reasons of simplicity, i.e. for the case $n = 1$. We will carry the sparse grid approach over to vector-valued functions and manifolds with $n > 1$ in the following subsection.

First, we restrict ourselves to the case of piecewise d-linear functions in the sparse grid construction. We proceed as follows: In a piecewise linear setting, the simplest choice of a 1 D basis function is the standard hat function $\phi(x)$,

$$\phi(x) \;:=\; \begin{cases} 1-|x|, & \text{if } x \;\in\; [-1,1]\,, \\ 0, & \text{else.} \end{cases} \tag{4.1}$$

This function can be used to generate an arbitrary $\phi_{l_j,i_j}(x_j)$ with associated support $[x_{l_j,i_j} - h_{l_j}, x_{l_j,i_j} + h_{l_j}] = [(i_j-1)h_{l_j}, (i_j+1)h_{l_j}]$ by dilation and translation, that is

$$\phi_{l_j,i_j}(x_j) \;:=\; \phi\left(\frac{x_j - i_j \cdot h_{l_j}}{h_{l_j}}\right). \tag{4.2}$$

The resulting 1 D basis functions are the input of the tensor product construction which provides a suitable piecewise d-linear basis function in each grid point $\mathbf{x}_{\mathbf{l},\mathbf{i}} := \mathbf{i} \cdot \mathbf{h}_{\mathbf{l}}$, $\mathbf{0} \le \mathbf{i} \le 2^{\mathbf{l}}$, see Figure 4.1:

$$\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) \;:=\; \prod_{j=1}^{d} \phi_{l_j,i_j}(x_j)\,. \tag{4.3}$$

Here, $\mathbf{l} = (l_1, \ldots, l_d) \in \mathbb{N}^d$ denotes a multi-index which indicates the multivariate level of refinement, $\mathbf{i} = (i_1, \ldots, i_d) \in \mathbb{N}^d$ denotes a multi-index which indicates the multivariate position, $\mathbf{0} := (0, \ldots, 0)$, and the inequalities in $\mathbf{0} \le \mathbf{i} \le 2^{\mathbf{l}}$ are to be understood componentwise. We thus consider the family of d-dimensional standard rectangular grids

$$\left\{T_{\mathbf{l}}, \mathbf{l} \in \mathbb{N}^d\right\} \tag{4.4}$$

on $T = [0,1]^d$ with multivariate mesh size $\mathbf{h}_{\mathbf{l}} \;:=\; (h_{l_1}, \ldots, h_{l_d}) \;:=\; 2^{-\mathbf{l}}$. That is, the grid $T_{\mathbf{l}}$ is equidistant with respect to each individual coordinate direction, but, in general, may have different mesh sizes in the different coordinate directions. The grid points $\mathbf{x}_{\mathbf{l},\mathbf{i}}$ of grid $T_{\mathbf{l}}$ are just the points

$$\mathbf{x}_{\mathbf{l},\mathbf{i}} \;:=\; (x_{l_1,i_1}, \ldots, x_{l_d,i_d}) \;:=\; \mathbf{i} \cdot \mathbf{h}_{\mathbf{l}}\,, \qquad \mathbf{0} \le \mathbf{i} \le 2^{\mathbf{l}}\,. \tag{4.5}$$

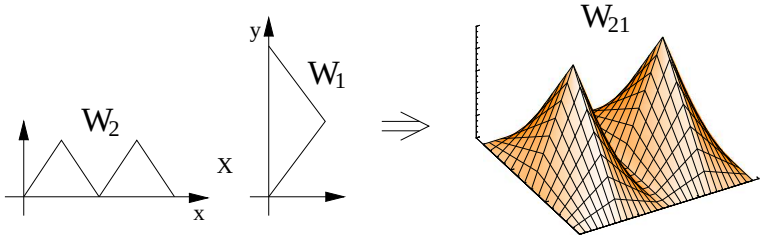

Figure 4.1: Tensor product approach for piecewise bilinear basis functions.

Clearly, the functions $\phi_{\mathbf{l},\mathbf{i}}$ (with obvious modification at the boundary of T) span the space $V_{\mathbf{l}}$ of piecewise d-linear functions on T on grid $T_{\mathbf{l}}$, i.e.

$$V_{\mathbf{l}} \;:=\; \operatorname{span}\left\{\phi_{\mathbf{l},\mathbf{i}} :\; \mathbf{0} \le \mathbf{i} \le 2^{\mathbf{l}}\right\}, \tag{4.6}$$

and form a basis of $V_{\mathbf{l}}$.

Additionally, we introduce the hierarchical increments $W_{\mathbf{l}}$,

$$W_{\mathbf{l}} \;:=\; \operatorname{span}\left\{\phi_{\mathbf{l},\mathbf{i}} : \begin{array}{ll} 1 \le i_j \le 2^{l_j}-1,\ i_j \text{ odd}, & \text{if } l_j > 0, \\ 0 \le i_j \le 1, & \text{if } l_j = 0, \end{array} \; 1 \le j \le d\right\}, \tag{4.7}$$

for which the relation

$$V_{\mathbf{l}} \;=\; \bigoplus_{\mathbf{t} \le \mathbf{l}} W_{\mathbf{t}} \tag{4.8}$$

can be seen easily. Note that the supports of all basis functions $\phi_{\mathbf{l},\mathbf{i}}$ spanning $W_{\mathbf{l}}$ are mutually disjoint for $\mathbf{l} > \mathbf{0}$. Thus, with the index set

$$\mathbf{I}_{\mathbf{l}} \;:=\; \left\{\mathbf{i} \in \mathbb{N}^d : \begin{array}{ll} 1 \le i_j \le 2^{l_j}-1,\ i_j \text{ odd}, & \text{if } l_j > 0, \\ 0 \le i_j \le 1, & \text{if } l_j = 0, \end{array} \; 1 \le j \le d\right\}, \tag{4.9}$$

we get another basis of $V_{\mathbf{l}}$, the *hierarchical basis*

$$\{\phi_{\mathbf{k},\mathbf{i}} : \mathbf{i} \in \mathbf{I}_{\mathbf{k}}, \mathbf{k} \le \mathbf{l}\} \tag{4.10}$$

which generalizes the well-known 1 D basis shown in Figure 4.2 to the d-dimensional case by means of a tensor product approach.

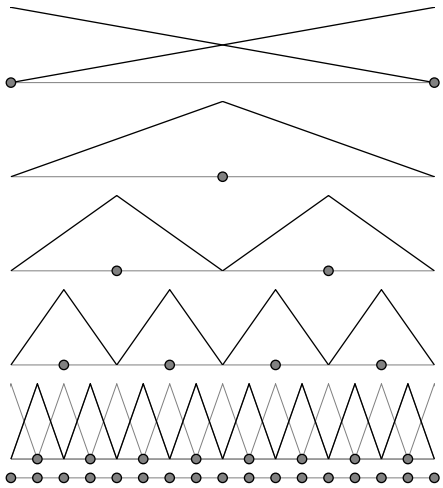

Figure 4.2: Piecewise linear hierarchical basis, $l = 4$.

With these hierarchical difference spaces $W_{\mathbf{l}}$, we can define

$$V^{(d)} \;:=\; \sum_{l_1=0}^{\infty} \cdots \sum_{l_d=0}^{\infty} W_{(l_1,\ldots,l_d)} \;=\; \bigoplus_{\mathbf{l} \in \mathbb{N}_0^d} W_{\mathbf{l}} \tag{4.11}$$

with its natural *hierarchical basis*

$$\left\{\phi_{\mathbf{l},\mathbf{i}} : \mathbf{i} \in \mathbf{I}_{\mathbf{l}}, \mathbf{l} \in \mathbb{N}_0^d\right\}. \tag{4.12}$$

Now it is easy to see that any function $f \in V^{(d)}$ can be uniquely split by

$$f(\mathbf{x}) \;=\; \sum_{\mathbf{l}} f_{\mathbf{l}}(\mathbf{x}), \qquad f_{\mathbf{l}}(\mathbf{x}) \;=\; \sum_{\mathbf{i} \in \mathbf{I}_{\mathbf{l}}} v_{\mathbf{l},\mathbf{i}} \cdot \phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) \;\in\; W_{\mathbf{l}}\,, \tag{4.13}$$

where the $v_{\mathbf{l},\mathbf{i}} \in \mathbb{R}$ are the coefficient values of the hierarchical product basis representation of f.

The main observation is now as follows: The coefficients $v_{\mathbf{l},\mathbf{i}}$ with respect to the hierarchical basis possess a specific decay with the level $\mathbf{l}$ if f possesses bounded second mixed derivatives, i.e. if

$$f : T \to \mathbb{R} : D^{\boldsymbol{\alpha}} u \in L_q(T), |\boldsymbol{\alpha}|_\infty \le r,$$

with $r = 2$, where

$$D^{\boldsymbol{\alpha}} f \;:=\; \frac{\partial^{|\boldsymbol{\alpha}|_1} f}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}. \tag{4.14}$$

Here, $\boldsymbol{\alpha} \in \mathbb{N}_0^d$ with the norms $|\boldsymbol{\alpha}|_1 := \sum_{j=1}^d \alpha_j$ and $|\boldsymbol{\alpha}|_\infty := \max_{1 \le j \le d} \alpha_j$.

A straightforward calculation using partial integration twice and the product structure, see [18] for details, gives the integral representation[8]

$$v_{\mathbf{l},\mathbf{i}} \;=\; \int_\Omega \psi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) \cdot D^{\mathbf{2}} f(\mathbf{x}) \; \mathrm{d}\mathbf{x} \tag{4.15}$$

for any coefficient value $v_{\mathbf{l},\mathbf{i}}$ of the hierarchical representation (4.13) of f with $\mathbf{l} > \mathbf{0}$. Here $\psi_{l_j,i_j}(x_j) := -2^{-(l_j+1)} \cdot \phi_{l_j,i_j}(x_j)$, and furthermore $\psi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) := \prod_{j=1}^d \psi_{l_j,i_j}(x_j)$. We then can derive the estimate

$$|v_{\mathbf{l},\mathbf{i}}| \;\le\; 2^{-d} \cdot 2^{-2\cdot|\mathbf{l}|_1} \cdot |u|_{\mathbf{2},\infty} = O(2^{-2\cdot|\mathbf{l}|_1}), \quad \mathbf{l} > \mathbf{0}, \tag{4.16}$$

with respect to the semi-norm $|f|_{\boldsymbol{\alpha},\infty} \;:=\; \|D^{\boldsymbol{\alpha}} f\|_\infty$. In other words, if f belongs to the space of functions with second bounded mixed derivatives, then its hierarchical coefficients possess a decay like $2^{-2\cdot|\mathbf{l}|_1}$. For the detailed proof see e.g. [18].

Depending on the norm of the error we are interested in, this justifies various truncation schemes of the series expansion of f. For a given $k \in \mathbb{N}$, the regular sparse grid space is defined as

$$V_k^{(1)} := \bigoplus_{q(\mathbf{l}) \le k} W_{\mathbf{l}} \tag{4.17}$$

with

$$q(\mathbf{l}) := 1 + \sum_{\substack{m=1,..,d \\ l_m \neq 0}} (l_m - 1) \quad \text{and} \quad q(\mathbf{0}) = 0,$$

see also [18, 66]. The associated truncated series, i.e. the interpolant of f in $V_k^{(1)}$ reads

$$f_k^{(1)} := \sum_{q(\mathbf{l}) \le k} \sum_{\mathbf{i}} v_{\mathbf{l},\mathbf{i}} \phi_{\mathbf{l},\mathbf{i}}.$$

Note that this is the finite element analogon of the well-known hyperbolic cross or Korobov spaces which are based on the Fourier series expansion instead of the hierarchical Faber basis. An example of a regular sparse grid is given for the two- and three-dimensional case in Figure 4.3. The basic concept can be traced back to [5, 61], see also [23, 26, 34].

The dimension of the space $V_k^{(1)}$ fulfills

$$|V_k^{(1)}| \;=\; O(h_k^{-1} \cdot |\log_2 h_k|^{d-1}) \tag{4.18}$$

with $h_k = 2^{-k}$, whereas for the interpolation error of a function f in the sparse grid space $V_k^{(1)}$ there holds

$$||f - f_k^{(1)}||_{L_p} \;=\; O(h_k^2 \cdot k^{d-1}), \tag{4.19}$$

for the L_p-norms, and

$$||f - f_k^{(1)}||_E = O(h_k), \tag{4.20}$$

[8] For coefficients associated to the boundary, i.e. for $l_j = 0$, partial integration is not applied for the respective j-th coordinate directions but the respective coordinate x_j is just set to zero or one, depending on the value of i_j. This leads to the general formula

$$v_{\mathbf{l},\mathbf{i}} \;=\; \left[\int \cdots \int \prod_{\substack{j=1 \\ l_j \neq 0}}^d \psi_{l_j,i_j}(x_j) \left(\prod_{\substack{j=1 \\ l_j \neq 0}}^d \frac{\partial^2}{\partial x_j} \right) f(\mathbf{x}) \quad d\left(\prod_{\substack{j=1 \\ l_j \neq 0}}^d x_j \right) \right]_{\mathbf{x}|_{(\mathbf{l}=\mathbf{0})} := \mathbf{x}_{\mathbf{l},\mathbf{i}}|_{(\mathbf{l}=\mathbf{0})}}$$

where $\mathbf{x}|_{(\mathbf{l}=\mathbf{0})}$ denotes the tuple of coordinates from $\mathbf{x}$ with $l_j = 0$. In this case, estimate (4.15) involves $|f|_{\boldsymbol{\alpha},\infty}$ with $\alpha_j = 2$ if $l_j > 0$ and $\alpha_j = 0$ if $l_j = 0$. Furthermore, the term 2^{-d} gets replaced by $2^{-|\mathrm{sgn}(\mathbf{l})|_1}$ but the term $2^{-2|\mathbf{l}|_1}$ stays the same.

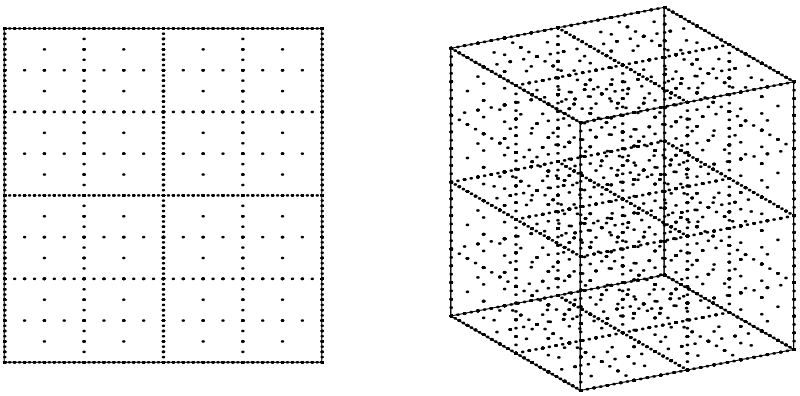

Figure 4.3: Regular sparse grids: Two-dimensional example (left) and three-dimensional example (right).

for the energy (semi-)norm $\|f\|_E = (\nabla f, \nabla f)^{1/2}$ induced by the Laplacian, see for example [18] for detailed proofs. Note that the conventional full grid space

$$V_k^{(\infty)} := \bigoplus_{|\mathbf{l}|_\infty \leq k} W_\mathbf{l}$$

results in an error in the L_p-norm of the order $O(h_k^2)$ and an error in the energy-norm of the order $O(h_k)$ (albeit for functions with just bounded conventional second derivative). It however possesses a dimension $|V_k^{(\infty)}| = O(h_k^{-d})$ and thus exhibits the curse of dimensionality with respect to h_k. In comparison to that we now see a crucial improvement for $V_k^{(1)}$: The number of degrees of freedom is significantly reduced, whereas the accuracy deteriorates only slightly for the L_p-norm and stays of the same order for the energy-norm. The curse of dimensionality is now present in the $\log(h_k)$-term only. Note that this result is optimal for function with second bounded mixed derivative with respect to the L_p-norms.

This basic concept of sparse grids has been generalized in various ways: First, there are special sparse grids which are optimized with respect to the energy semi-norm [17]. These energy-based sparse grids are further sparsified and thus possess a cost complexity of order $O(h_k^{-1})$ and result in an accuracy of order $O(h_k)$. Thus, the exponential dependence of the logarithmic terms on d is completely removed (but is still present in the constants). A thorough discussion of the constants can be found in [35]. A generalization to sparse grids which are optimal with respect to other Sobolev norms can be found in [37]. Then, there are generalized sparse grids [32], dimension-adaptive sparse grids [33] and locally adaptive sparse grids [36], all with favorable properties and specific applications. Note finally that also basis functions of higher order, or prewavelets and wavelets can be used straightforwardly in the sparse grid construction process instead of the common hat function ϕ from (4.1). For further details, see [18].

4.2 Sparse grids and full grids for manifolds

The sparse grid construction for the approximation of scalar functions can now easily be carried over to the case of a vector of functions, i.e. to functions $f : T \to X$, $f = (f_{(1)}(t), \ldots, f_{(n)}(t))$ which represent manifolds in a parametric way. To this end, each coordinate function $f_{(i)}$ of f is represented in the *same* multivariate hierarchical basis and is approximated on the same sparse grid. Then all properties of sparse grids for scalar functions carry over to the vector valued case in a straightforward way. The representation of a manifold f in the sparse grid space $(V_k^{(1)})^n$ reads

$$f_k^{(1)}(t) := \sum_{q(\mathbf{l}) \leq k} \sum_{\mathbf{i}} v_{\mathbf{l},\mathbf{i}} \phi_{\mathbf{l},\mathbf{i}}(t) \tag{4.21}$$

now with just vector-valued hierarchical coefficients $v_{\mathbf{l},\mathbf{i}} \in \mathbb{R}^n$. Analogously, the representation of a manifold f in the full grid space $(V_k^{(\infty)})^n$ reads

$$f_k^{(\infty)}(t) \;:=\; \sum_{|\mathbf{l}|_\infty \le k} \sum_{\mathbf{i}} v_{\mathbf{l},\mathbf{i}} \phi_{\mathbf{l},\mathbf{i}}(t). \tag{4.22}$$

Now if we plug these expansions into the minimization problem (3.1) we obtain the associated discrete minimization problems (3.3). In the gradient descent algorithm we obtain then corresponding minimization problems (3.4) in the projection step and linear systems (3.5) in the adaption step. The switch in notation from index j, value α_j and number M of (3.2) to the multi-index $(\mathbf{l},\mathbf{i})$, the value $v_{\mathbf{l},\mathbf{i}}$ and numbers $|V_k^{(1)}|$ and $|V_k^{(\infty)}|$ in (4.21) and (4.22) is obvious: All we need is an enumeration of the multi-indices $(\mathbf{l},\mathbf{i})$ involved in the respective sums, i.e. a unique mapping $(\mathbf{l},\mathbf{i}) \to j$. We leave this to the reader and refrain here from explicitly giving (3.3), (3.4) and (3.5) in $(\mathbf{l},\mathbf{i})$-notation for the ease of presentation.

For the regularization term we employ $S(f) = \|Gf\|^2_{L_2(T,X)}$ with $G = \nabla$ or more general vector-valued differential operators.[9] Then, the matrix C in (3.6) resembles the discrete Laplacian or the associated corresponding discrete differential operators. Alternatively, we may use the squared generalized variation of Hardy and Krause $S(f) = \sum_{i=1}^d V_{HK}^{(U,V,W)}(f_{(i)})$. Depending on the specific choice of U and V, this relates to a fixed (squared) length of the boundary curves, a fixed area of the surface of the boundary sides, a fixed volume, etc. of the manifold.

The overall costs for the sparse grid approach are then as follows: The number of degrees of freedom is $M = O(2^k k^{d-1})$. The projection step involves in a naive implementation $O(NM)$ operations whereas a more sophisticated piecewise newton method allows to employ the compactly supported basis functions and needs $O(Nk^{d-1})$ operations with exponential d-dependent order constant. For the adaption step we now do not assemble the matrices but merely program the action of the matrix-vector multiplication which is needed in an iterative solver like the preconditioned CG method. The matrix-vector multiplication[10] then costs $O(M + Nk^{d-1})$ operations. The cost for the solution of the linear systems involves the number of iterations needed to reach a prescribed accuracy which depends on the condition number of the system matrix. In the best case, a multigrid or multilevel method may be envisioned here, for which the number of iterations is independent of M. Then, assuming a constant number of EM iterations, the overall solution costs behave like $O(M+Nk^{d-1})$. Here, we presently employ a multilevel preconditioned conjugate gradient method which is based on prewavelets, see [29, 39]. Alternative multigrid-like solvers may be designed along the lines of [38, 40]. Altogether we see that the sparse grid approach scales linear in the amount M of data and, up to logarithmic factors, also linear in the number of grid points employed. This has to be compared to the kernel-based approach which scales cubic in M due to the globality of the kernel. Furthermore, the number of degrees of freedom is now $M = O(2^k k^{d-1})$ for level k and thus depends exponentially on d only with respect to the logarithmic term k (albeit the constants in the order notation still may scale exponentially with d).

The costs for the full grid approach are as follows: The number of degrees of freedom is now $M = O(2^{kd})$. The projection step in its update version involves now $O(N)$ operations, the adaption step involves in the matrix-vector multiplication $O(M + N)$ operations and, in the best case of a multigrid method, the overall solution costs behave like $O(M + N)$.

Altogether, we see that the sparse grid approach is with $M = O(2^k k^{d-1})$ substantially more cost effective and thus allows to deal with problems involving large set of data points and dimensions larger than three. This is in contrast to the full grid approach where $M = O(2^{kd})$ and in contrast to the kernel-based method with global kernel where the cost scales cubic in its M.[11]

[9] Note that for an appropriate wavelet basis we also can make use of norm-equivalences of the type $\|f\|^2_{H^s} \approx \sum_{\mathbf{l}} 2^{2s|\mathbf{l}|_\infty} \sum_{\mathbf{i}} f^2_{\mathbf{l},\mathbf{i}} \|\phi_{\mathbf{l},\mathbf{i}}\|^2_{L_2}$ to replace a Sobolev-type norm based on a differential operator of degree s by a diagonally weighted sum of wavelet coefficients. Then, the matrix C in (3.6) resembles just a diagonal matrix with the weights $2^{2s|\mathbf{l}|_\infty} \|\phi_{\mathbf{l},\mathbf{i}}\|^2_{L_2}$. This gives the possibility to implement more involved Sobolev-type regularization terms in an easy way.

[10] The implementation via the matrix-vector multiplication avoids the assembly of the matrix. Therefore we have a storage complexity of only $O(N+M)$ which allows to deal with much larger problems than for the kernel approach where the associated full matrix (global kernel) is usually assembled explicitly.

[11] If the kernel functions $k(q_j, t)$ are chosen data centered, i.e. $q_j = x_j$, then $M = N$ here.

5 Numerical experiments

We now consider the results of numerical experiments.

5.1 Principal curves, $d = 1$

First, we consider the behavior of our approach for the most simple case of curves f, i.e. $f : T \to X$, with $T = [0, 1]^1$ and $X = \mathbb{R}^n$, where $n = 2$. Here, a sparse grid is not invoked yet, only a simple one-dimensional grid on T is used. We employ data points sampled from a circle with noise, compare [3]. Figure 5.1 (left) shows the result obtained on a coarse grid with the PCA as initial value. Using this solution as initial value for a refined grid, we get Figure 5.1 (second picture) and a further grid refinement of this solution yields Figure 5.1 (third picture). The circle structure is learned with increasing accuracy. Note that the result of the EM algorithm depends sensitively on the respective start value: Figure 5.1 (right) shows a substantially worse result obtained by a directly solution on level 4 with PCA as initial value.

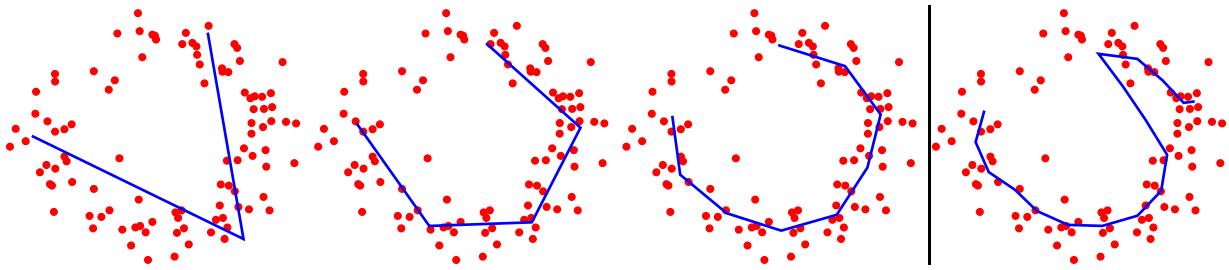

Figure 5.1: Coarse solution of a circle-like problem (left) and successive grid refinements with starting values as solution from corresponding next coarser levels (second and third) and finally a direct solution on level 4 with 1st eigenvector of PCA as starting value (right).

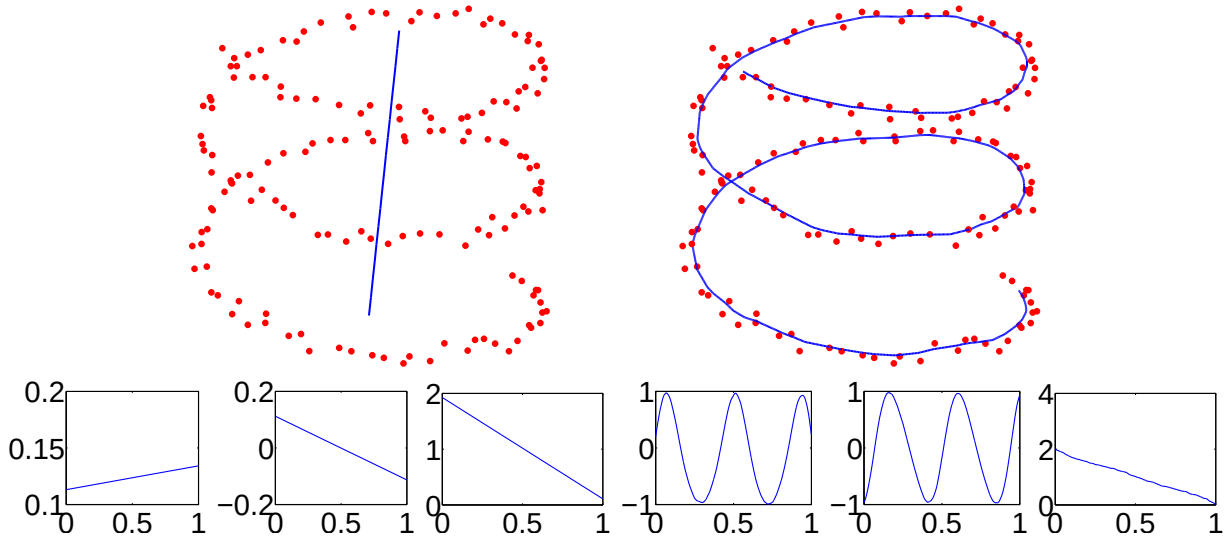

Figure 5.2: Starting situation with third eigenvector of the PCA and reconstructed helix after 32 iterations on level 9 (top) and their corresponding three component functions (bottom).

We now consider a principal curve problem in three-dimensional space. To this end, we randomly sample 160 points of the helix

$$f(t) = \big(\sin t, \cos t, 2t/(5\pi)\big)^T, \quad t \in T = [0, 5\pi] \tag{5.1}$$

with white noise of variance 0.03, i.e. the points may be off the helix. Figure 5.2 shows the resulting curve

together with the three component functions $f_{(1)}(t), f_{(2)}(t), f_{(3)}(t)$. We see that the reconstruction was perfectly successful, the structure of the helix was indeed learned.

Before we return to the helix example to analyse the approximation properties of our approach at the end of this section, we treat a principal curve problem with two clearly separated point clusters shown in Figure 5.3 (left). If we use a conventional principal component analysis, we obtain a quite bad reconstruction for such a data set. The first and the second eigenvector of the PCA of the data is shown in Figure 5.3 (middle) and (right), respectively. We clearly see that such data leads to bad linear reconstructions with an error proportional to the respective diameter of the data. Here, the first eigenvector is the optimal solution with respect to the empirical quantization error (2.1), but a projection onto it destroys the previously existing data separation. A projection onto the second eigenvector maintains the separation, but its empirical quantization error is substantially larger.

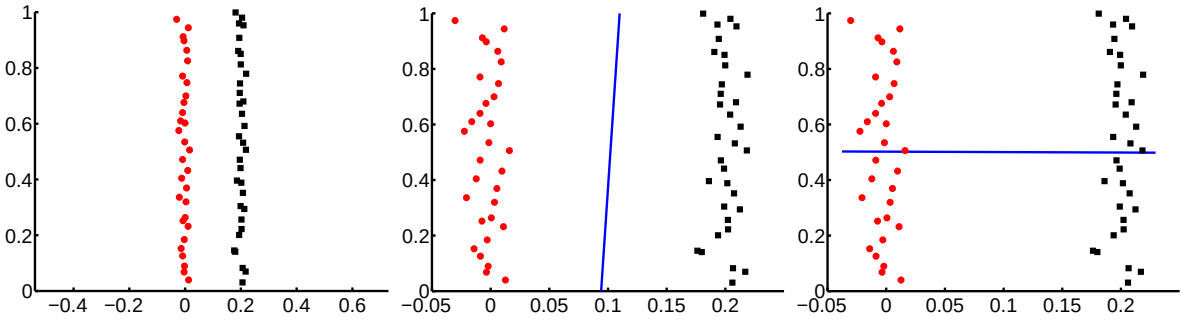

Figure 5.3: Data points (left), data points and first eigenvector, rescaled x-axis (middle), data points and second eigenvector, rescaled x-axis (right).

Now, we consider the results obtained with our grid-based manifold reconstruction approach. To this end, we start the EM algorithm with the 2nd eigenvector of the PCA. Figure 5.4 shows the results obtained for successively finer levels of discretization, i.e. mesh widths.

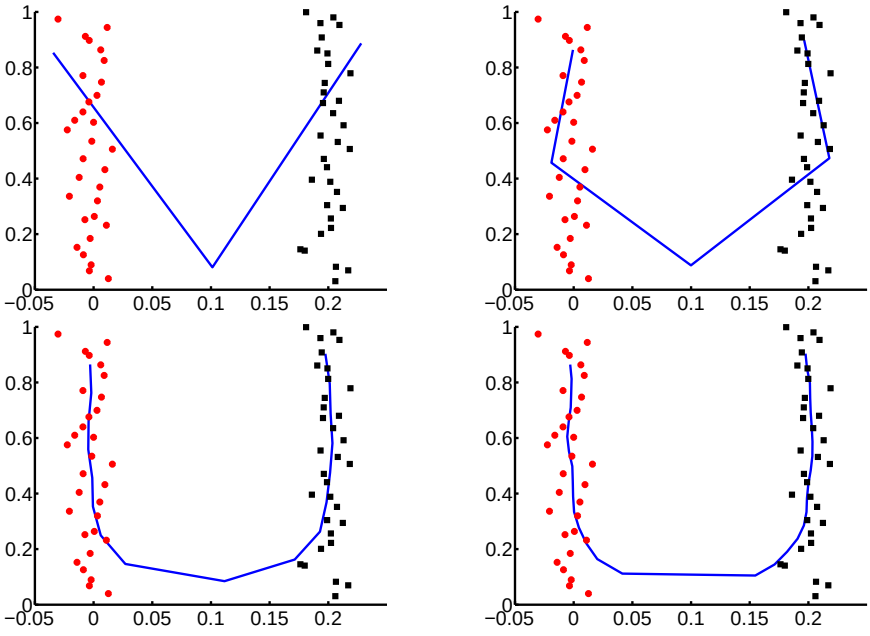

Figure 5.4: Solution of EM algorithm on four successive levels of refinement, second eigenvector of the linear PCA as starting value, rescaled x-axis.

Again, the result of the EM algorithm depends sensitively on the respective starting value. If we start the procedure with the first eigenvector a substantially worse solution results. This is depicted in Figure 5.5.

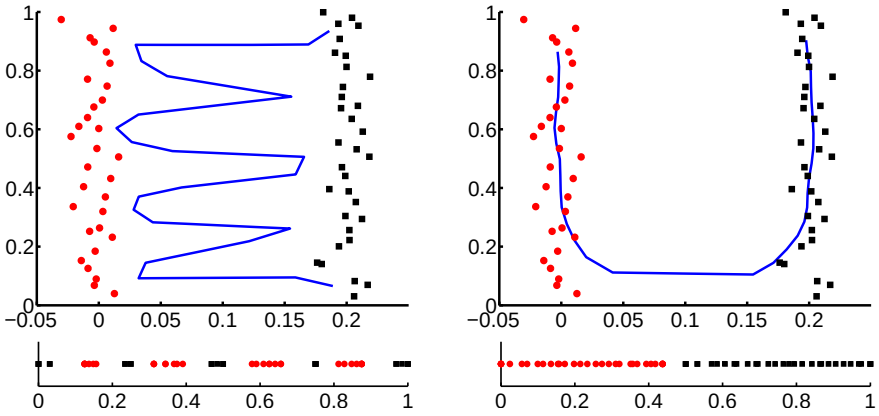

Figure 5.5: Solution of EM algorithm: first eigenvector of linear PCA as starting value (left) and second eigenvector of linear PCA (right). Associated latent variables in T-space (bottom left and right).

Here, the result (top) with the first eigenvector as starting value together with the latent variables, i.e. the data points projected onto T (bottom) is shown on the left side, whereas the result with the second eigenvector as starting value is shown on the right side. We clearly see that the solution on the left side is substantially worse than the one on the right side. Further experiments with additional points at the bottom, between the two clusters, exhibited the same qualitative results.

Finally, we consider the approximation properties of our approach in more detail using the helix (5.1) as model curve. To this end, we are interested in the convergence rate of the root mean square error

$$RMSE_k := \sqrt{\frac{1}{|X(f_K)|} \sum_{i=1}^{|X(f_K)|} d(x_i, f_k)}. \tag{5.2}$$

For this, we compute the solution f_K of the problem (3.3) on a uniform grid with K levels of refinement, i.e. $(2^K+1)^d$ points x_i in parameters space T, then we sample f_K randomly using 50˙000 points which gives the test data set $X(f_K)$ and compute for each point in $X(f_K)$ the squared distance to f_k as $d(x_i, f_k) = \inf_{t\in T} \|x_i - f_k(t)\|_2^2$ involving orthogonal projection, compare the projection step in the above-mentioned descent method. We also consider the convergence of the maximum error

$$\max_k := \max_{x_i \in X(f_K)} \sqrt{d(x_i, f_k)} \tag{5.3}$$

which is closely related to the Hausdorff distance of two curves.

In Table 5.1 we give the results for 163˙840 training data points x_i which were sampled randomly but equally distributed, this time *without* noise, from the helix. We clearly see a convergence order of two for the $RMSE$- and the max-error. Note that the number of data points is still larger than the number of grid

k	$n \cdot M$	RMSE_k	$\frac{\mathrm{RMSE}_k}{\mathrm{RMSE}_{k+1}}$	$\sqrt{\max}_k$	$\frac{\sqrt{\max}_k}{\sqrt{\max}_{k+1}}$
4	51	3.52_{-2}	3.9	8.17_{-2}	4.0
5	99	8.94_{-3}	4.0	2.06_{-2}	4.0
6	195	2.25_{-3}	4.0	5.20_{-3}	3.8
7	387	5.61_{-4}	4.0	1.37_{-3}	4.0
8	771	1.41_{-4}	4.0	3.45_{-4}	2.8
9	1˙539	3.55_{-5}	3.9	1.24_{-4}	2.0
10	3˙075	8.98_{-6}	–	6.08_{-5}	–

Table 5.1: Error and convergence rate for the helix problem measured against the discrete solution on level $K = 17$, 163˙840 training points without noise, $\lambda = 1.95 \cdot 10^{-6}$, $S = \nabla$.

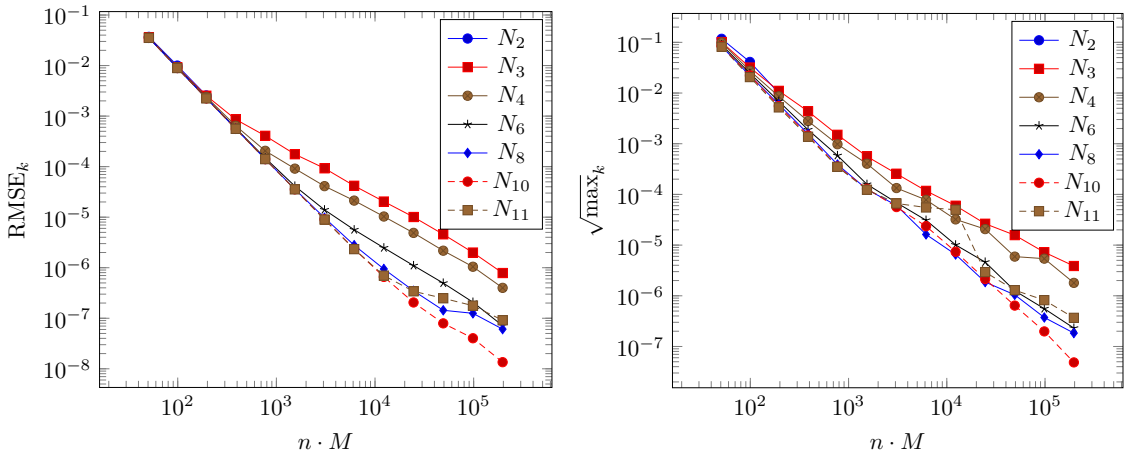

Figure 5.6: Error for the helix problem measured against the discrete solution on level $K = 17$ versus the degrees of freedom $n \cdot M$ for varying training data sets of sizes $N_i \in \{320, 640, 1\dot{2}80, ...\}$, $\lambda = 1.95 \cdot 10^{-6}$, $S = \nabla$.

points employed. Note furthermore that we consider here convergence towards the reconstructed curve on a fine level K only and not towards the sampled interpolated true helix yet.

In further experiments we observed that if the number of grid points exceeds the number of data points then the rates deteriorate somewhat into roughly first order which either may indicate overfitting effects or may reflect the $H^{3/2}$-regularity of the solution of the adaption step problems due to the Dirac right hand sides, compare also [31]. This can be seen in Figure 5.6.

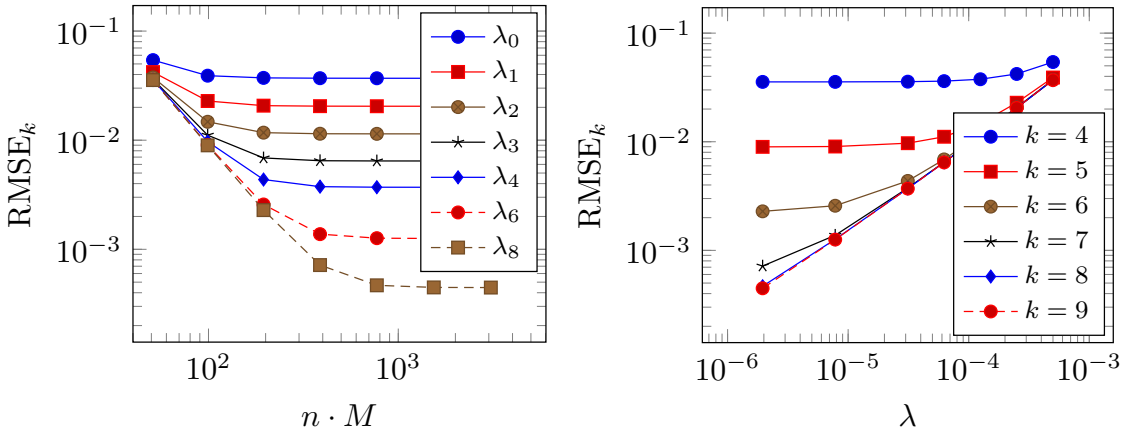

Figure 5.7: RMSE measured against the true helix versus degrees of freedom for varying $\lambda_i = 5 \cdot 10^{-4} \cdot 2^{-i}$ (left) and RMSE versus λ for varying levels k (right) for the helix problem, $S = \nabla$, training data set of size 163˙840.

Now we are interested in the convergence towards the true helix f. To this end we have to successively use more sample data points, more grid points, i.e. finer levels l, and successively smaller values of λ. To this end, the test data set X of points in (5.2) and (5.3) is sampled from the true f. Figure 5.7 (left) shows the resulting RMSE versus the degrees of freedom[12] to represent f_k for varying values of λ. We clearly see that the error decays first for a rising number of degrees of freedom but then stays constant depending on the respective value of λ. We also see that if we use successively smaller values of λ *and* successively larger numbers of degrees of freedom we obtain convergence with a rate of the order two, compare the slope of the

[12]Please note that the term "degrees of freedom" denotes the number of basis coefficients. There are M basis functions with n components each.

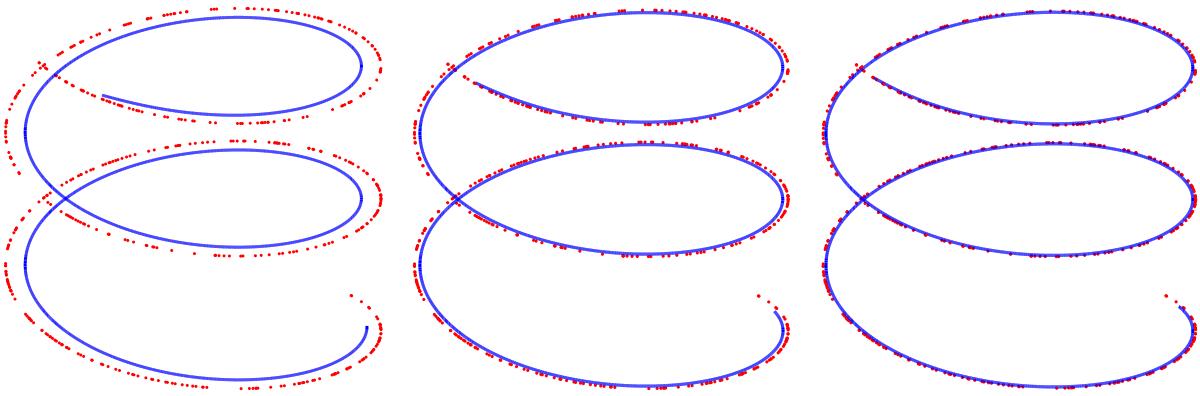

Figure 5.8: Reconstructed curve (bold) and values sampled from the true helix (points) for $\lambda = 2.0 \cdot 10^{-3}, 5.0 \cdot 10^{-4}, 2.5 \cdot 10^{-5}$.

left lower curve in Figure 5.7 (left). In Figure 5.7 (right) we show the RMSE versus λ for varying levels k for the discretization. An analogous behavior can be observed here.

The error is here substantially influenced by the parameter λ. Recall that the regularization term $\|\nabla f\|^2_{L_2(T,X)}$ induces a length constraint on the reconstructed curve. This effect can be seen in Figure 5.8 in more detail. The reconstructed curve (bold) converges towards the values sampled from the true helix (points) for rising values of λ. Due to the imposed length constraint, the computed helix is somewhat shrinked for larger values of λ (Figure 5.8 (left)). But its diameter soon tends to approach the true helix diameter for smaller values of λ (Figure 5.8 (middle)). Nevertheless, for still smaller values, its length is still a bit restricted and the very first and last points of the true helix are thus not reached (Figure 5.8 (right)).

Note finally that in general just two EM iterations were needed to reach the relative discretization error accuracy when starting with the solution on the next coarser level.

5.2 Principal surfaces, $d = 2, n = 3$

So far, we only dealt with curves, i.e. one-dimensional manifolds, which were approximated by simple polygons involving just a one-dimensional grid. Now, we turn to two-dimensional manifolds in three-dimensional space, i.e. principal surfaces, where we may employ regular sparse grids for the three two-dimensional component functions $f_{(1)}(t_1, t_2), f_{(2)}(t_1, t_2), f_{(3)}(t_1, t_2)$. This results in a substantial saving compared to the use of functions which live on a uniform full grid.

As an example, we consider a simple half sphere which we sample randomly using 516˙961 points (without noise). For an illustration see Figure 5.9.

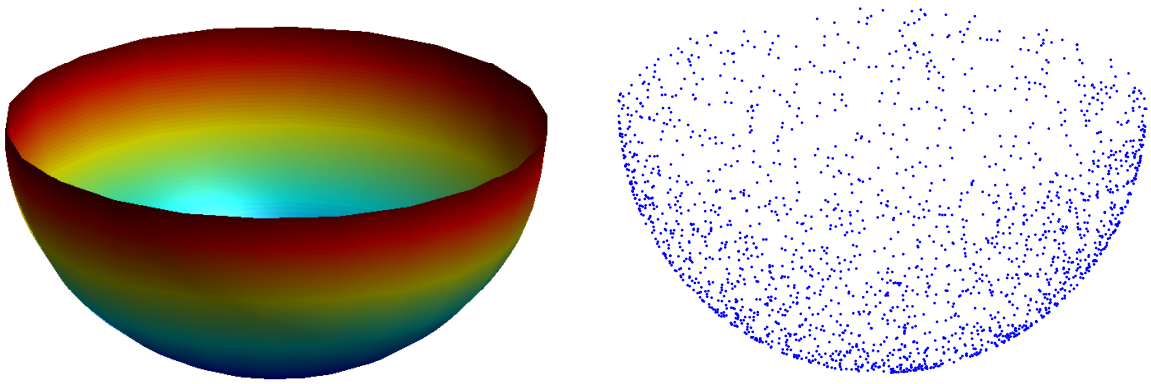

Figure 5.9: Half sphere manifold (left) and sample points (right).

k	$n \cdot M$	RMSE_k	$\frac{\mathrm{RMSE}_k}{\mathrm{RMSE}_{k+1}}$	$\sqrt{\max}_k$	$\frac{\sqrt{\max}_k}{\sqrt{\max}_{k+1}}$
3	243	5.54_{-3}	3.83	2.60_{-2}	3.66
4	867	1.45_{-3}	3.88	7.10_{-3}	3.81
5	3˙267	3.73_{-4}	3.92	1.87_{-3}	3.82
6	12˙675	9.51_{-5}	3.69	4.89_{-4}	3.06
7	49˙923	2.58_{-5}	2.96	1.60_{-4}	2.50
8	198˙147	8.70_{-6}	2.16	6.38_{-5}	2.34
9	789˙507	4.04_{-6}	2.20	2.73_{-5}	2.55
10	3˙151˙875	1.84_{-6}	–	1.07_{-5}	–

Table 5.2: Error and convergence rate for the half sphere problem measured against the discrete solution on level $K = 11$, 1˙034˙289 training points, full uniform grid, $\lambda = 5 \cdot 10^{-4}$, regularization term (5.4).

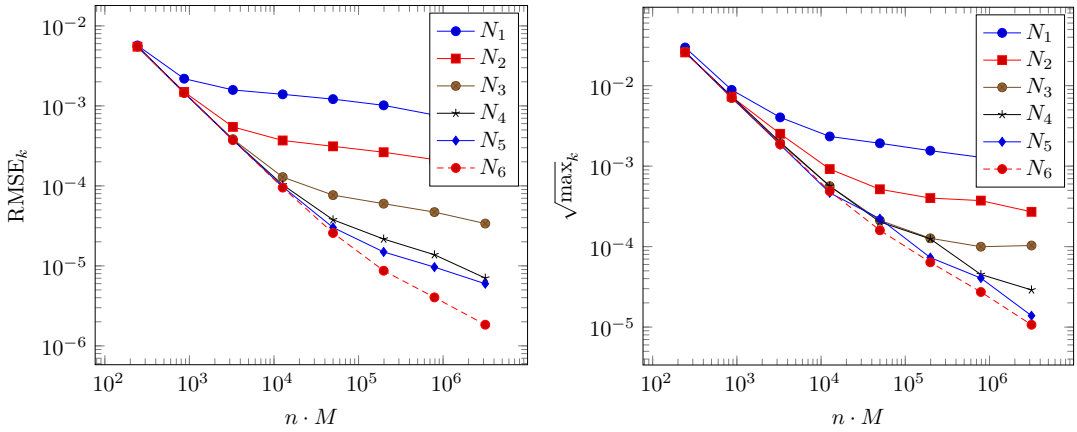

Figure 5.10: Error for the half sphere problem versus the degrees of freedom $n \cdot M$ and varying training data sets of sizes $N_i = \{8˙100, 32˙041, 128˙064, 516˙961, 1˙034˙289\}$, full uniform grid, $\lambda = 5 \cdot 10^{-4}$, regularization term (5.4), measured against the discrete solution on the finest level $K = 11$.

As regularization term we employ

$$
\begin{aligned}
S(f) = \sum_{i=1}^{3} \int_0^1 \int_0^1 \left[\partial_{t_1} f_{(i)}(t_1,t_2)\right]^2 + \left[\partial_{t_2} f_{(i)}(t_1,t_2)\right]^2 \, dt_1 \, dt_2 \\
+ \frac{1}{5} \int_0^1 \left[\partial_{t_1} f_{(i)}(t_1,0)\right]^2 + \left[\partial_{t_1} f_{(i)}(t_1,1)\right]^2 \, dt_1 \\
+ \frac{1}{5} \int_0^1 \left[\partial_{t_2} f_{(i)}(0,t_2)\right]^2 + \left[\partial_{t_2} f_{(i)}(1,t_2)\right]^2 \, dt_2,
\end{aligned}
\tag{5.4}
$$

i.e. we use the example (2.8) with different weights $w_{u,v}$. The first part expresses an area restriction for the surface while the other four terms can be considered as a length restriction for the four boundary curves. The weight factors $1/5$ are a subjective choice which gave good results in our numerical experiments.

In Table 5.2 we show the results for the case of uniform full grids. As regularization parameter we employed $\lambda = 5 \cdot 10^{-4}$. We see a convergence order of roughly two for the $RMSE_k$-error and the $\max_k$-error which somewhat declines on higher levels to a rate of about one. Note that the amount of points used in the discretization scales with 2^{2k} which reflects the use of a full two-dimensional grid.

In further experiments we observed that the onset of the decline of the convergence rate is related to the number of grid points, the number of data points and the choice of λ. This can be seen in Figure 5.10. The deterioration of the convergence rate especially for the smaller number of data points indicates typical

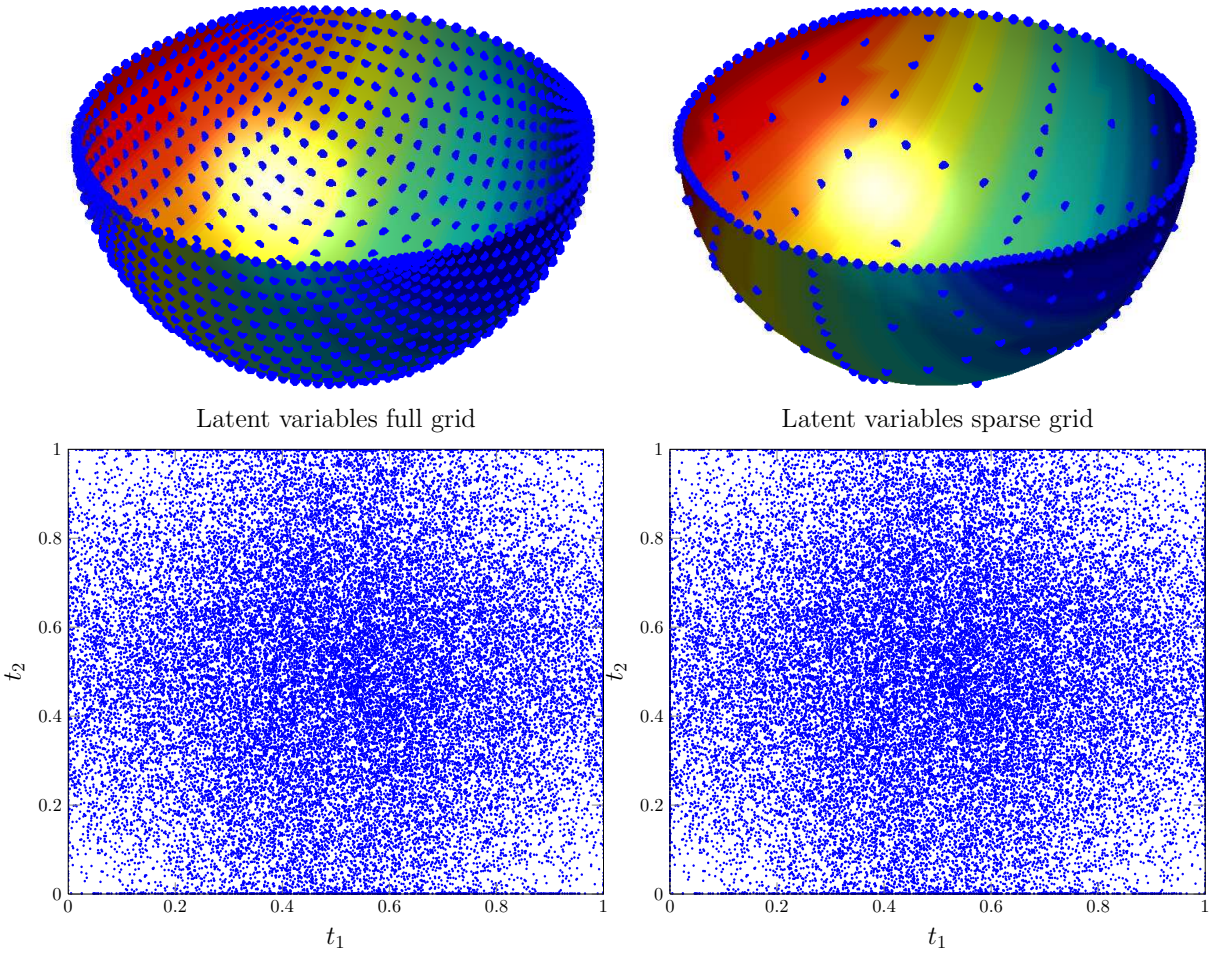

Figure 5.11: Reconstructed half spheres, full grid (left) and sparse grid (right), with associated latent variables, $k = 5$.

overfitting effects. For a proper choice of the amount of data points, of the amount of grid points and of the regularization parameter the rate of two can however be maintained.

The reconstructed half spheres for 516˙961 training points and discretization level $k = 5$ together with their grid points are shown in Figure 5.11 for the full grid case (top left) and the sparse grid case (top right). Figure 5.11 (bottom left and right) provides the corresponding latent variables for a data set with 32˙041 points.

In Table 5.3 we now give for comparison the error and convergence rate for the case of regular sparse grids. We use the value $\lambda = 1.95 \cdot 10^{-6}$ to compensate for the fewer points in the sparse grid.[13] We see a convergence order of roughly two for the $RMSE_k$-error and the $\max_k$-error at coarser levels (maybe with an additional log-factor which is typical for sparse grids) which is substantially reduced on the finer levels due to overfitting. Note that the amount M of points used in the discretization now only scales with $k \cdot 2^k$ which reflects the use of a sparse two-dimensional grid.

Again, the onset of the decline of the convergence rate is related to the number of grid points, the number of data points and the choice of λ. This can be seen in Figure 5.12. The deterioration of the convergence rate especially for the smaller number of data points indicates typical overfitting effects. The rate of about two can however be maintained for a proper choice of the amount of data points, of the amount of grid points and of the regularization parameter. Note furthermore that a study of the convergence of the numerical

[13]Note that besides the regularization due to the smoothing term $S(f)$ in general a further regularization by discretization comes into play.

k	$n \cdot M$	RMSE_k	$\frac{\text{RMSE}_k}{\text{RMSE}_{k+1}}$	$\sqrt{\max}_k$	$\frac{\sqrt{\max}_k}{\sqrt{\max}_{k+1}}$
3	147	5.71_{-3}	3.96	2.64_{-2}	3.64
4	339	1.44_{-3}	3.82	7.25_{-3}	2.72
5	771	3.77_{-4}	3.65	2.66_{-3}	1.40
6	1˙731	1.03_{-4}	3.37	1.90_{-3}	3.35
7	3˙843	3.07_{-5}	2.72	5.68_{-4}	1.63
8	8˙451	1.13_{-5}	1.85	3.49_{-4}	0.96
9	18˙435	6.09_{-6}	1.31	3.63_{-4}	0.98
10	39˙939	4.66_{-6}	1.15	3.71_{-4}	1.03
11	86˙019	4.04_{-6}	1.13	3.59_{-4}	1.04
12	184˙323	3.56_{-6}	–	3.47_{-4}	–

Table 5.3: Error and convergence rate for the half sphere problem measured against the discrete solution on level $K = 14$, 1˙034˙289 training points, sparse grid, $\lambda = 1.95 \cdot 10^{-6}$, regularization term (5.4).

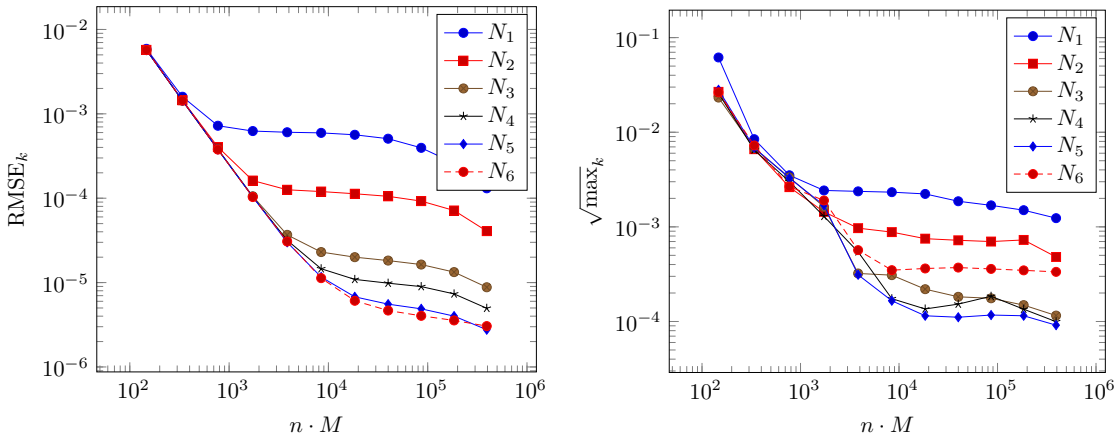

Figure 5.12: Error for the half sphere problem versus the degrees of freedom $n \cdot M$ and varying training data sets of sizes $N_i = \{8˙100, 32˙041, 128˙064, 516˙961, 1˙034˙289\}$, sparse grid, $\lambda = 1.95 \cdot 10^{-6}$, regularization term (5.4), measured against the discrete solution on the finest level $K = 14$.

solution to the true half sphere gave qualitatively similar results as for the helix problem of Figure 5.7.

Altogether, the sparse grid behaves superior to the full grid due to its reduced amount of grid points when it comes to the question of accuracy versus costs involved. This makes the sparse grid approach a good candidate for manifold learning problems in moderately higher dimensions.[14]

5.3 A classification example

Since there is a close relationship of our approach to the GTM we also applied the sparse grid method to the oil flow data set which is used in [10] as a benchmark problem for the GTM. Here, the problem is to determine the fraction of oil in a multi-phase pipeline carrying a mixture of oil, water and gas. Each data point consists of 12 measurements taken from dual-energy gamma densitometers measuring the attenuation of gamma beams passing through the pipe. Synthetically generated data is used which models accurately the attenuation processes in the pipe, as well as the presence of noise (arising from photon statistics), for details see [9]. The three phases in the pipe (oil, water and gas) can belong to one of three different geometrical configurations, corresponding to laminar, homogeneous, and annular flows. The data set consists of 1˙000 points drawn with equal probability from the three configurations.

[14] In practice, problems with up to 12 dimensions can be dealt with the regular sparse grid method. Higher dimensional problems usually can not be handled due to the log-terms involved. Then one can resort to energy-norm based sparse grids, generalized sparse grids or dimension-adaptive sparse grids which may work in higher dimensions when other error measures are considered and/or additional properties of the manifold like weighted, i.e. not equally important dimensions are present.

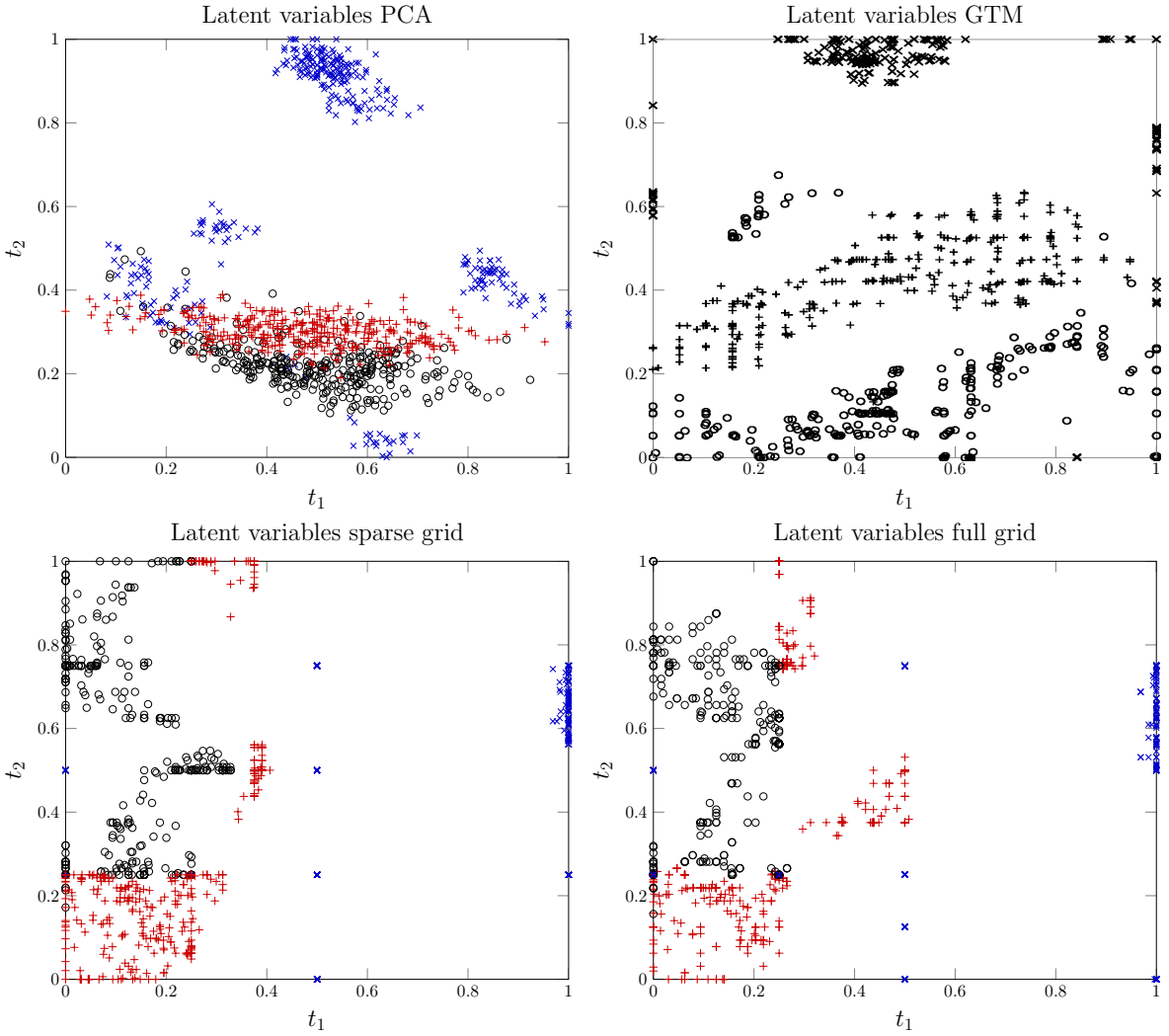

Figure 5.13: Oil flow data: Result for the linear PCA (top left), the original result copied from [10] (top right), the sparse grid approach (lower left) with $k = 6$, $\lambda = 0.01$, regularization term (5.4) and the full grid approach (lower right) with same parameters, $d = 2$.

The main goal is now data visualization and cluster detection. As for the original GTM, our algorithm requires the manifold's intrinsic dimension d as parameter which has to be fixed a-priori. Several methods to estimate the intrinsic dimension have been proposed, see [8, 15, 52] and the references therein.

To this end, the latent-variable space is chosen to be two-dimensional, and the data points are mapped from $\mathbb{R}^{12}$ via the reconstructed manifold into the latent space $T = [0, 1]^2$. Each point is then labelled according to its multi-phase configuration. From the distribution of points one may then get information of the data's intrinsic structure.

Figure 5.13 gives the results (from top left to bottom right) for the linear PCA, the original GTM[15] from [10] and our sparse grid and full grid approaches. Both, sparse grid and full grid approach use the regularization term (5.4). The learning procedure employed successive grid refinement up to level 6. Here, the blue crosses, black circles and red plus signs represent stratified, annular and homogeneous multi-phase configurations, respectively. We see that PCA fails completely, the three classes are not visually separated at all. For our sparse grid approach however a clear spatial separation of the data points is achieved. There, even a separation of the class of red plus signs into 3 further subclasses can be observed. Thus, the result

[15]The original graphics from [10] has been rescaled and mapped into the unit cube for display reasons. Markers have been chosen consistently with those of [10].

obtained with our sparse grid approach reveals much more of the data's intrinsic structure than a simple search for directions with high variance. These findings are comparable to those obtained with the GTM approach [10], shown in figure 5.13 (top right) and the kernel based method from [60]. However, the costs of the computation is now substantially reduced due to the sparse grid method. Our prototype required about 30 seconds on a 2.5GHz workstation for the sparse grid result (total 32 iterations, successive grid refinement).

Finally, we choose the latent-variable space to be three-dimensional, i.e. the data points are mapped from $\mathbb{R}^{12}$ via the reconstructed manifold into $T = [0,1]^3$. We used the regularization term (2.6) with $U = \{\emptyset\} \cup \{u \subset \{1,2,3\} : |u| = 1\} \cup \{u \subset \{1,2,3\} : |u| = 2\}, V(u) = \{v \subset \bar{u} : |v| = 1\}$ and weights $w_{u,v}$ with values 1 if $|u| = 0$, values 0.3 if $|u| = 1$, and values 0.1 if $|u| = 2$. The result is shown in Figure 5.14. Now, an even better separation of classes is obtained: The black, red and blue points are well separated and the subclusters for the red plus signs can be seen as well. Furthermore, the black circles also exhibit subclusters.

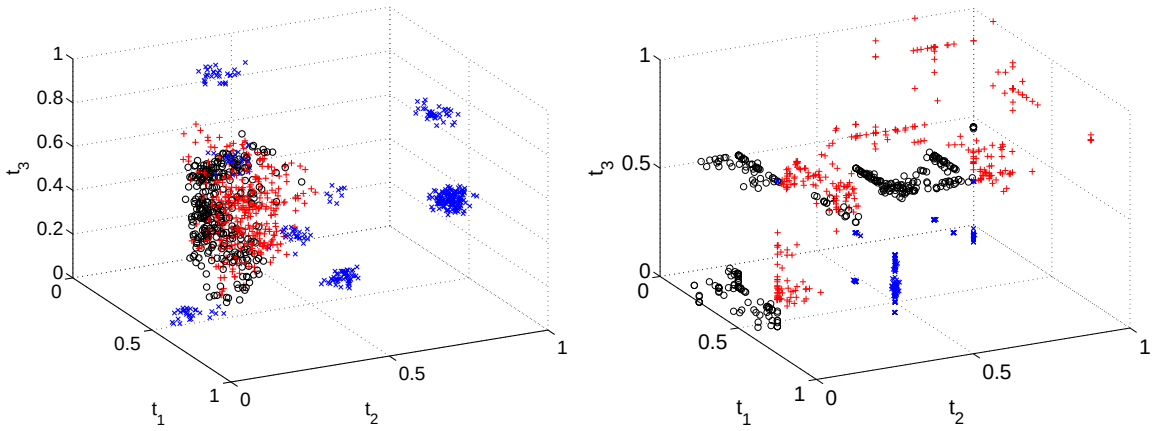

Figure 5.14: Oil flow data: Result for the linear PCA (left) and the sparse grid approach (right) with $k = 5$, $\lambda = 0.01$, regularization term (2.6), $d = 3$.

6 Concluding remarks

In this paper we presented a sparse grid method for the construction of lower-dimensional principal manifolds from high-dimensional data. This approach avoids to some extent the curse of dimension that appears with conventional grids. The arising non-linear problem is solved by a descent method which resembles the expectation maximization algorithm. We discussed the basic ideas and main ingredients of the approach and demonstrated its properties for one-, two- and three-dimensional model problems to give a proof of concept. The method can in principle be applied to problems with manifolds with about 12 intrinsic dimensions, provided that a better and more efficient implementation than the present prototype one is realized. This may be done along the lines of [29]. But for higher dimensions the logarithmic terms pose a practical obstacle here. Then, energy-norm based sparse grids, dimension-adaptive sparse grids or fully adaptive sparse grid versions of our approach may be envisioned.

Presently, the proper choice of the regularization term S for $d > 1$ is an open question and needs further investigation. So far, we made good experiences with a regularization term that is similar in structure to the variation of Hardy and Krause. Its terms directly relate to geometric constraints on the manifold. But further investigations are needed here.

Furthermore, a proper choice of the initial value for the optimization turns out to be important. This is a common problem for most nonlinear learning methods. Here, our successive refinement approach imposes some kind of successively reduced regularization which improves the EM iteration in comparison to the fine initial grid. However, the results are still sensitive to initial values. A further problem is the selection of the final grid level and an appropriate value for λ.

Up to now, the dimension of the intrinsic space, i.e. the manifold, must be chosen a-priori. It would be advantageous to have a method where this dimension is determined automatically from the given data. To this end, there exist certain techniques where dimension is heuristically estimated from local or global PCAs, compare [8, 15, 52] and the references cited therein. A more theoretically founded approach relies on Whitney's famous embedding theorem [65] and its refinement due to Taken's [62]. Here, Broomhead and King suggested to calculate the dimension d of the manifold a-posteriori by first taking the embedding dimension *sufficiently high* and then determining d from the numerical rank of the covariance matrix of the embedded data [13]. The resulting numerical strategy can be seen as a combination of Taken's method of delays with the principal component analysis of the data in some extended space. Actual methods for dimension estimation which draw from these ideas are found in [14, 44, 50]. These techniques may be incorporated into a dimension-adaptive version of our sparse grid method to obtain the necessary dimension of the manifold in an adaptive way, provided proper error estimators can be developed. However, such an approach is future work.

References

[1] http://www.cse.msu.edu/∼lawhiu/manifold/

[2] http://www.cs.ubc.ca/∼mwill/dimreduct.htm

[3] http://www.iro.umontreal.ca/∼kegl/research/pcurves/implementations/index.html

[4] N. Aronzaijn, *Theory of reproducing kernels*, Trans. Amer. Math. Soc., 68, (1950), pp. 337–404.

[5] K. Babenko, *Approximation by trigonometric polynomials in a certain class of periodic functions of several variables*, Soviet Math. Dokl. 1, (1960), pp. 672–675. Russian original in Dokl. Akad. Nauk SSSR, 132 (1960), pp. 982–985.

[6] R. Balder, *Adaptive Verfahren für elliptische und parabolische Differentialgleichungen*, Dissertation, Technische Universität München (1994).

[7] R. Balder and C. Zenger, *The solution of the multidimensional real Helmholtz equation on sparse grids*, SIAM J. Sci. Comp., 17, (1996), pp. 631–646.

[8] D. Banks and R. Olszewski, *Estimating local dimensionality*, in Proceedings of the Statistical Computing Section of the American Statistical Society, ASA (1997).

[9] C. Bishop and G. James, *Analysis of multiphase flows using dual-energy gamma densitometry and neural networks*, Nuclear Instruments and Methods in Physics Research A327, 580-593 (1993).

[10] C. Bishop, M. Svensen, and C. Williams, *GTM: The generative topographic mapping*, Neural Computations, 10(2), (1998), pp. 215–234.

[11] C. Bishop, M. Svensen, and C. Williams, *Developments of the generative topographic mapping*, Neurocomputing, 21, (1998), pp. 203–224.

[12] T. Bonk, *Ein rekursiver Algorithmus zur adaptiven numerischen Quadratur mehrdimensionaler Funktionen*, Dissertation, Institut für Informatik, Technische Universität München (1994).

[13] D. Broomhead and G. King, *Extracting qualitative dynamics from experimental data*, Physica D, 20, 217 (1986).

[14] D. Broomhead and M. Kirby, *A new approach to dimensionality reduction: Theory and algorithms*, SIAM J. Applied Mathematics. 60(6), (2000), pp. 2114–2142.

[15] J. Bruske and G. Summer, *Intrinsic dimensionality estimation with optimally topology preserving maps*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(5), (1998), pp. 572–575.

[16] H.-J. Bungartz, *An adaptive Poisson solver using hierarchical bases and sparse grids*, in Iterative Methods in Linear Algebra, Elsevier, (1992), pp. 293–310.

[17] H.-J. Bungartz and M. Griebel, *A note on the complexity of solving Poisson's equation for spaces of bounded mixed derivatives*, J. Complexity, 15, (1999), pp. 167–199.

[18] H.-J. Bungartz and M. Griebel, *Sparse grids*, Acta Numerica, 13, (2004), pp. 1–121.

[19] K. Chang and J. Ghosh, *A unified model for probabilistic principal surfaces*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(1), (2001), pp. 22–4

[20] K Chang and J. Ghosh, *Probabilistic principal surfaces classifier*, in L. Wang and Y. Jin (eds.), FSKD 2005, LNAI 3614, (2005), pp. 1236–1244.

[21] M. Carreira-Perpinan, *A review of dimension reduction techniques*, Technical Report CS-96-09, Department of Computer Science, University of Sheffield (1997).

[22] P. Delicado, *Another look at principal curves and surfaces*, Journal of Multivariate Analysis, 77(1), (2001), pp. 84–116

[23] F. Delvos and W. Schempp, *Boolean methods in interpolation and approximation*, Vol. 230 of Pitman Research Notes in Mathematics, Longman Scientific and Technical, Harlow, (1989).

[24] A. Dempster, N. Laird and D. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. Roy. Statist. Soc. B, 39(1), (1977), pp. 1–38.

[25] R. Der, U. Steinmetz, and G. Balzuweit, *Nonlinear principal component analysis*, Technical Report, Institut für Informatik, Universität Leipzig (1998).

[26] R. DeVore, S. Konyagin and V. Temlyakov, *Hyperbolic wavelet approximation*, Constr. Approx. 14, (1998), pp. 1–26.

[27] D. Dong and T. McAvoy, *Nonlinear principal component analysis - based on principal curves and neural networks*, Computers Chem. Engineering, 20(1), (1995), pp. 65–78.

[28] T. Evgeniou, M. Pontil and T. Poggio, *Regularization networks and support vector machines*, Advances in Computational Mathematics, 13, (2000), pp. 1–50.

[29] C. Feuersänger, *Dünngitterverfahren für hochdimensionale elliptische partielle Differentialgleichungen*, Diplomarbeit, Institut für Numerische Simulation, Universität Bonn (2005).

[30] 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), pp. 694–716.

[31] J. Garcke and M. Hegland, *Fitting multidimensional data using gradient penalties and combination techniques*, in Proceedings of HPSC 2006, Hanoi, Vietnam (2006).

[32] T. Gerstner and M. Griebel, *Numerical integration using sparse grids*, Numerical Algorithms, 18, (1998), pp. 209–232.

[33] T. Gerstner and M. Griebel, *Dimension-adaptive tensor-product quadrature*, Computing, 71(1), (2003), pp. 65–87.

[34] W. Gordon, *Blending function methods of bivariate and multivariate interpolation and approximation*, SIAM J. Numer. Anal. 8, (1971), pp. 158–177.

[35] M. Griebel, *Sparse grids and related approximation schemes for higher dimensional problems*, in Proceedings of the Conference on Foundations of Computational Mathematics (FoCM05), Santander, Spain (2005), Foundations of Computational Mathematics (L. Pardo, A. Pinkus, E. Suli and M.J. Todd, eds), LMS 331, Cambridge University Press, Cambridge (2006).

[36] M. Griebel, *Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences*, Computing, 61(2), (1998), pp. 151–179.

[37] M. Griebel and S. Knapek, *Optimized tensor-product approximation spaces*, Constr. Approx. 16(4), (2000), pp. 525–540.

[38] M. Griebel and P. Oswald, *On additive Schwarz preconditioners for sparse grid discretizations*, Numer. Math., 66, (1994), pp. 449–464.

[39] M. Griebel and P. Oswald, *Tensor product type subspace splitting and multilevel iterative methods for anisotropic problems*, Adv. Comput. Math., 4, (1995), pp. 171–206.

[40] M. Griebel, C. Zenger, and S. Zimmer, *Multilevel Gauss-Seidel-algorithms for full and sparse grid problems*, Computing, 50, (1993), pp. 127–148.

[41] T. Hastie, *Principal curves and surfaces*, PhD thesis, Stanford University (1984).

[42] T. Hastie and W. Stuetzle, *Principal curves*, Journal of the American Statistical Association, 84(406), (1989), pp. 502–516.

[43] X. Huo, X. Ni, and A. Smith, *A survey of manifold-based learning methods*, in Mining of enterprise data, emerging nonparametric methodology, chapter 1, Springer, New York (2006).

[44] A. A. Jamshidi, and M. J. Kirby, *Towards a black box algorithm for nonlinear function approximation over high-dimensional domains*, SIAM J. Sci. Comput., 29(3), (2007), pp. 941–963.

[45] I. Jollife, *Principal component analysis*, Springer-Verlag, New York (1986).

[46] J. Jost, *Differentialgeometrie und Minimalflächen*, Springer (1994).

[47] B. Kégl, *Principal curves: learning, design, and applications*, PhD. Thesis, Concordia University, Canada (1999).

[48] B. Kégl, A. Krzyzak, T. Linder, and K. Zeger, *Learning and design of principal curves*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(3), (2000), pp. 281–297.

[49] G. Kimmelsdorf and G. Wahba, *Some results on Tchebycheffian spline functions*, Journal of Mathematical Analysis aand Applications 33, (1971), pp. 82–95.

[50] M. Kirby, *Geometric Data Analysis: An empirical approach to dimensionality reduction and the study of patterns*, John Wiley and Son, New York (2001).

[51] M. Kramer, *Nonlinear principal component analysis using autoassociative neural networks*, AIChE Journal, 37, (1991), pp. 233–243.

[52] T. Minka, *Automatic choice of dimensionality for PCA*, in T. Leen, T. Dietterich, and V. Tresp, editors, Advances in Neural Information Processing Systems 13, pages 598–604. MIT Press (2001).

[53] W. Press, B. Flannery, S. Teukolsky and W.-Vetterling, *Numerical recipes in C*, Cambridge University Press (1992).

[54] A. Owen, *Multidimensional variation for quasi-Monte Carlo*, Technical Report 2004-02, Dep. of. Statistics, Stanford Univ. (2004).

[55] S. Paskov, *Average case complexity of multivariate integration for smooth functions*, J. Complexity, 9(2), (1993), pp. 291–312.

[56] S. Sandilya and S. Kulkarni, *Principal curves with bounded turn*, IEEE Trans. on Information Theory, 48(10), (2000), pp. 2789–2793.

[57] C. Schwab and R. Todor, *Sparse finite elements for stochastic elliptic problems-higher order moments*, Computing, 71, (2003), pp. 43–63.

[58] B. Schoelkopf and A. Smola, *Learning with kernels*, MIT Press (2002).

[59] B. Schölkpf, R. Herbrich, A. Smola, and R. Williamson, *A generalized representer theorem*, Technical Report 200-81, NeuroCOLT 2000, in Proceedings COLT'2001, Lecture Notes on Artificial Intelligence, Springer (2001).

[60] A. Smola, S. Mika, B. Schölkopf, and R. Williamson, *Regularized principal manifolds*, Journal of Machine Learning Research, 1, (2001), pp. 179–209.

[61] S. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Soviet Math. Dokl. 4, 240–243. Russian original in Dokl. Akad. Nauk SSSR, 148, (1963), pp. 1042–1045.

[62] F. Takens, *Detecting strange attractors in turbulence*, in Dynamical Systems and Turbulence, D. Rand and L. Young, eds., Springer, New Yorck, Lecture Notes in Mathematics, 366 (1981).

[63] R. Tibshirani, *Principal curves revisited*, Statistics and Computation, 2, (1992), pp. 183–190.

[64] G. Wahba, *Spline models for observational data*, Volume 59 of CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia (1990).

[65] H. Whitney, *Differentiable manifolds*, Annals of Mathematics, 37, (1936), pp. 645–680.

[66] C. Zenger, *Sparse grids*, Parallel Algorithms for Partial Differential Equations, W. Hackbusch (ed.), NNFM 31, Vieweg, Braunschweig/Wiesbaden (1991).