Hierarchical Gradient-Based Optimization with B-Splines on Sparse Grids

Julian Valentin and Dirk Pflüger

Abstract Optimization algorithms typically perform a series of function evaluations to find an approximation of an optimal point of the objective function. Evaluations can be expensive, e.g., if they depend on the results of a complex simulation. When dealing with higher-dimensional functions, the curse of dimensionality increases the difficulty of the problem rapidly and prohibits a regular sampling. Instead of directly optimizing the objective function, we replace it with a sparse grid interpolant, saving valuable function evaluations. We generalize the standard piecewise linear basis to hierarchical B-splines, making the sparse grid surrogate smooth enough to enable gradient-based optimization methods. Also, we use an uncommon refinement criterion due to Novak and Ritter to generate an appropriate sparse grid adaptively. Finally, we evaluate the new method for various artificial and real-world examples.

1 Introduction

In this work, we want to solve optimization problems of the following form: Assume we are given a continuous function $f\colon [0,1]^d \to \mathbb{R}$ (*objective function*). Our goal is to find a minimal point

$$\mathbf{x}_{\text{opt}} = \operatorname*{arg\,min}_{\mathbf{x}\in[0,1]^d} f(\mathbf{x}) \ , \tag{1}$$

i.e., we want to solve a general, bound-constrained optimization problem. Optimization algorithms, whether gradient-free or gradient-based, usually perform a series of evaluations of f, its gradient, or its Hessian (if available) to find an approximation $\mathbf{x}^*_{\text{opt}}$ of $\mathbf{x}_{\text{opt}}$. As each evaluation can be expensive, e.g. by triggering a cascade of

J. Valentin, D. Pflüger
Institute for Parallel and Distributed Systems, Universität Stuttgart,
Universitätsstr. 38, 70569 Stuttgart, Germany
e-mail: {julian.valentin,dirk.pflueger}@ipvs.uni-stuttgart.de

nested simulations, we want to use as few evaluations as possible. Of course, for increasing d, the problem suffers from the curse of dimensionality, which obviously suggests the employment of sparse grids for the solution. Optimization with the aid of sparse grids was studied before, e.g. with additional constraints and piecewise linear functions [11] or with sparse grid surrogates defined via Lagrange polynomials on Chebyshev points [9, 10]. However, 1D Lagrange polynomials are asymmetrical, have global support $[0, 1]$, and their degree 2^n is not tunable. In addition, polynomial interpolation prevents us from using equidistant grid points. We want to use B-splines as basis functions instead, as they do not have these drawbacks, but additionally feature many nice properties. B-splines have already been used in the context of sparse grids, e.g. for the purpose of data mining [27, 28] or quasi-interpolation [19]. The sufficient smoothness of B-splines allows us to use gradient-based optimization methods on the sparse grid interpolant efficiently, even if the gradient or Hessian of f are not available or costly to evaluate. Our optimization approach will be as follows:

1. Generate a spatially adaptive sparse grid $X = \{\mathbf{x}_k\}_k$ adapting to the peculiarities of f.
2. Interpolate f at X by an interpolant $\tilde{f}$ defined by a linear combination of B-splines on sparse grids.
3. Apply gradient-based optimization techniques to $\tilde{f}$ to get $\mathbf{x}^*_{\text{opt}}$.

In Sec. 2, we will define hierarchical B-splines and prove their linear independence in the univariate case, which generalizes to higher dimensionalities d. The B-splines will be modified to allow good approximations near the boundary of the domain $[0, 1]^d$. We will explain in Sec. 3 the refinement criterion by Novak and Ritter [26] we use to construct spatially adaptive sparse grids. A description of implementational details follows in Sec. 4. Finally, we evaluate our algorithm and compare it to established methods by studying various artificial and real-world examples in Sec. 5.

2 B-Splines on Sparse Grids

Conventional basis functions for sparse grids, including the piecewise polynomial functions by Bungartz [3], all share the shortcoming of not having globally continuous derivatives, hindering the use of gradient-based optimization. B-splines, which generalize the well-known hat functions, can tackle this problem. They were first studied by Schoenberg [34], who claimed that they were already known to Laplace [8]. But it was not until the 1960s when Schoenberg's results were rediscovered and the potential of B-splines for the emerging finite element method (FEM) was recognized. Important work was done by de Boor, who found simple B-spline algorithms [7]. B-splines have found application in a number of fields, e.g., for the aforementioned FEM [14], as non-uniform rational B-splines (NURBS) for geometric modeling [4, 15], for atomic and molecular physics [1, 23], and for financial mathematics [28], to name just a few examples. We will now repeat the definition of hierarchical B-Splines [28] and then prove their linear independence.

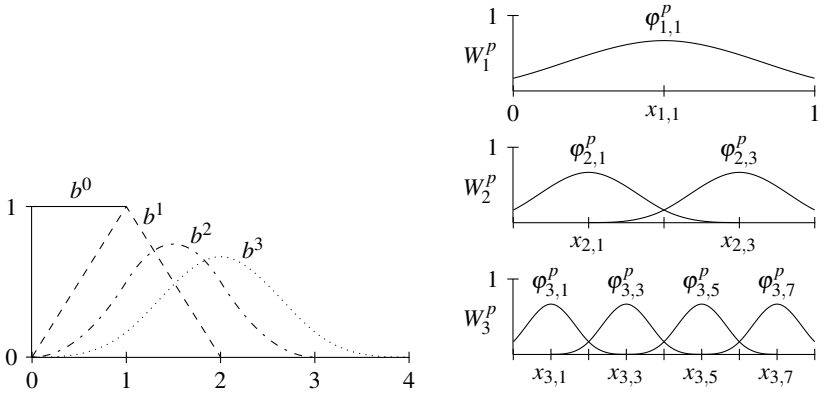

Fig. 1: *Left*: cardinal B-splines of degree $p = 0, 1, 2, 3$. *Right*: hierarchical B-splines of degree $p = 3$ and level $l = 1, 2, 3$.

2.1 Cardinal B-Splines

The *cardinal B-spline* $b^p : \mathbb{R} \to \mathbb{R}$ of degree $p \in \mathbb{N}_0$ is defined by

$$\begin{aligned} b^0(x) &= \chi_{[0,1)}(x) \ , \\ b^p(x) &= \int_0^1 b^{p-1}(x-y)\mathrm{d}y \ , \quad p \geq 1 \ , \end{aligned} \tag{2}$$

with the indicator function χ_A of $A \subset \mathbb{R}$, i.e., b^p is the convolution of b^{p-1} and b^0. This definition implies the following simple properties [15] (see Fig. 1, left). The support of b^p is $[0, p+1]$. On every interval $[k, k+1)$, $k = 0, \ldots, p$ (*knot interval*), b^p is a non-negative polynomial of degree p. The B-spline is bounded by 1 and symmetric with respect to $x = \frac{p+1}{2}$. b^p is $(p-1)$ times continuously differentiable at $x = 0, \ldots, p+1$ (*knots*). By differentiation of (2) we get the simple identity

$$\frac{\mathrm{d}}{\mathrm{d}x} b^p(x) = b^{p-1}(x) - b^{p-1}(x-1) \ . \tag{3}$$

2.2 Hierarchical B-Splines

The *hierarchical B-spline* $\varphi^p_{l,i} \colon [0,1] \to \mathbb{R}$ of level $l \in \mathbb{N}$ and index $i \in I_l := \{1, 3, 5, \ldots, 2^l - 1\}$ is defined by an affine parameter transformation [28],

$$\varphi^p_{l,i}(x) := b^p\left(\frac{x}{h_l} + \frac{p+1}{2} - i\right) \ , \quad h_l := 2^{-l} \ .$$

$\varphi^p_{l,i}(x)$ has support $[0,1] \cap (h_l \cdot [i \pm (p+1)/2])$ (see Fig. 1, right). For $p = 1$, we obtain the well-known piecewise linear hierarchical basis (hat functions). To simplify the next considerations, we only consider odd degree p, as the knots (where the B-spline is not infinitely many times differentiable) of $\varphi^p_{l,i}$ then coincide with the grid points

$$x_{l,i-(p+1)/2}\,, \quad \ldots\,, \quad x_{l,i}\,, \quad \ldots\,, \quad x_{l,i+(p+1)/2}$$

with $x_{l,i} := ih_l$. For even degree, the knots lie between the grid points, i.e.

$$x_{l,i-p/2} - \frac{h_l}{2}\,, \quad \ldots\,, \quad x_{l,i} - \frac{h_l}{2}\,, \quad x_{l,i} + \frac{h_l}{2}\,, \quad \ldots\,, \quad x_{l,i+p/2} + \frac{h_l}{2}\,,$$

leading to slightly different, but related arguments. We can define the *nodal B-spline space* V^p_l and the *hierarchical B-spline subspace* W^p_l of level l by

$$V^p_l := \operatorname{span}\{\varphi^p_{l,i} \mid i = 1, \ldots, 2^l - 1\}\,, \quad W^p_l := \operatorname{span}\{\varphi^p_{l,i} \mid i \in I_l\}\,.$$

2.3 Linear Independence of Hierarchical B-Splines

In the piecewise linear case ($p = 1$), the relationship $V^1_n = \bigoplus_{l=1}^n W^1_l$ can be seen easily. We prove that a similar relationship also holds for higher B-spline degrees. To this end, we first show the linear independence of the union $\{\varphi^p_{l,i} \mid l \le n, i \in I_l\}$ of the hierarchical functions up to level n with the aid of B-splines on general knots.

Let $m, p \in \mathbb{N}_0$ and $\boldsymbol{\xi} = (\xi_0, \ldots, \xi_{m+p})$ be an increasing sequence of real numbers (knot sequence). Then for $k = 0, \ldots, m-1$ the B-splines $b^p_{k,\boldsymbol{\xi}}$ of degree p with knots ξ are defined by the Cox-de Boor recurrence [5, 7, 15]

$$\begin{aligned} b^0_{k,\boldsymbol{\xi}} &:= \chi_{[\xi_k, \xi_{k+1})}\,, \\ b^p_{k,\boldsymbol{\xi}} &:= \gamma^p_{k,\boldsymbol{\xi}} b^{p-1}_{k,\boldsymbol{\xi}} + (1 - \gamma^p_{k+1,\boldsymbol{\xi}}) b^{p-1}_{k+1,\boldsymbol{\xi}}\,, \quad \gamma^p_{k,\boldsymbol{\xi}}(x) := \frac{x - \xi_k}{\xi_{k+p} - \xi_k}, \quad p \ge 1\,. \end{aligned}$$

For the special case of $\boldsymbol{\xi} = (0, 1, \ldots, p+1)$ and $k = 0$, we obtain the cardinal B-spline $b^p(x)$.

Proposition 1. *Let $\boldsymbol{\xi} = (\xi_0, \ldots, \xi_{m+p})$ be a knot sequence. Then the B-splines $b^p_{k,\boldsymbol{\xi}}$, $k = 0, \ldots, m-1$, form a basis of the spline space*

$$S^p_{\boldsymbol{\xi}} := \operatorname{span}\{b^p_{k,\boldsymbol{\xi}} \mid k = 0, \ldots, m-1\}. \tag{4}$$

$S^p_{\boldsymbol{\xi}}$ contains exactly those functions which are continuous on $D := [\xi_p, \xi_m]$, polynomials of degree $\le p$ on every knot interval $[\xi_k, \xi_{k+1}]$ in D and at least $(p-1)$ times continuously differentiable at every knot ξ_k in the interior of D.

The proposition, a proof of which can be found in [15], implies linear independence of the nodal B-splines $\{\varphi^p_{n,i} \mid i = 1,\dots,2^n-1\}$ of level $n \in \mathbb{N}$ by choosing

$$\xi_k := \left(k + 1 - \frac{p+1}{2}\right) h_n \ , \quad k = 0,\dots,m+p \ , \quad m := 2^n - 1 \ , \tag{5}$$

which leads to $\varphi^p_{n,i} = b^p_{i-1,\boldsymbol{\xi}}$ for $i = 1,\dots,m$, i.e. $S^p_{\boldsymbol{\xi}} = V^p_n$ when restricting all B-splines to $D = [\xi_p, \xi_m]$. In particular, this means $\{\varphi^p_{n,i} \mid i \in I_n\}$ is a basis of W^p_n.

Proposition 2. *For every $n \in \mathbb{N}$, the hierarchical B-splines $\{\varphi^p_{l,i} \mid l \le n,\ i \in I_l\}$ are linearly independent, i.e., the sum $\bigoplus_{l=1}^n W^p_l$ is indeed direct.*

Proof. We prove the assertion by induction over n for the most common degrees $p \in \{1,3,5,7\}$. For rather uncommon higher degrees, the proof can be viewed as a sketch. For $n = 1$, only one function exists. To proceed from $n-1$ to n, we assume that $\{\varphi^p_{l,i} \mid l \le n-1,\ i \in I_l\}$ is linearly independent, so its span $\bigoplus_{l=1}^{n-1} W^p_l$ is a direct sum of hierarchical subspaces. Because both sets $\{\varphi^p_{n,i} \mid i \in I_n\}$ and $\{\varphi^p_{l,i} \mid l \le n-1,\ i \in I_l\}$ are linearly independent, it is necessary and sufficient to show that $\operatorname{span}\{\varphi^p_{n,i} \mid i \in I_n\} \cap \bigoplus_{l=1}^{n-1} W^p_l = \{0\}$. Let $f_1 \in \operatorname{span}\{\varphi^p_{n,i} \mid i \in I_n\}$ and $f_2 \in \bigoplus_{l=1}^{n-1} W^p_l$ with $f_1 = f_2$. Then coefficients $c_{n,i}, c_{l,i} \in \mathbb{R}$ exist such that

$$\sum_{i \in I_n} c_{n,i} \varphi^p_{n,i} = f_1 = f_2 = \sum_{l=1}^{n-1} \sum_{i \in I_l} c_{l,i} \varphi^p_{l,i} \ .$$

The right-hand side is smooth in every grid point $x_{n,j}$, $j \in I_n$, of level n, as these grid points are not knots of the B-splines of level $< n$. So the left-hand side must be smooth there as well, i.e.

$$\partial^p_- f_1(x_{n,j}) = \partial^p_+ f_1(x_{n,j}) \ , \tag{6}$$

denoting with ∂^p_- and ∂^p_+ the left and right derivative of order p, respectively. Now we use the combinatorial identity

$$\partial^p_+ b^p(k) = (-1)^k \binom{p}{k} = \partial^p_- b^p(k+1) \ , \quad k \in \mathbb{Z} \ ,$$

setting $\binom{p}{k} := 0$ for $k < 0$ or $k > p$. The identity stems from the repeated application of relation (3) (cf. [15]). Calculating the left and the right derivative in $x_{n,j}$ of each summand of f_1 respectively, we obtain from (6)

$$\sum_{i \in I_n} c_{n,i} (-1)^{k-1} \binom{p}{k-1} = \sum_{i \in I_n} c_{n,i} (-1)^k \binom{p}{k} \ , \quad k := k(i,j) = j - i + \frac{p+1}{2} \ ,$$

due to $\varphi^p_{n,i}(x_{n,j}) = b^p(k)$. The inner derivative $1/h^p_l$ canceled out from both sides. Using the relation $\binom{p}{k-1} + \binom{p}{k} = \binom{p+1}{k}$, we get

$$\sum_{i \in I_n} c_{n,i} (-1)^k \binom{p+1}{k} = 0 \,, \quad j \in I_n \,. \tag{7}$$

As k is always odd or always even (for fixed j), we get

$$\sum_{i \in I_n} c_{n,i} \binom{p+1}{j - i + \frac{p+1}{2}} = 0 \,, \quad j \in I_n \,,$$

by multiplying (7) by -1 if k is odd. This is a linear system with variables $c_{n,i}$, whose sparsity pattern depends on p. The corresponding matrix $A = A(p)$ is a symmetric $(2^{n-1} \times 2^{n-1})$ Toeplitz matrix with bandwidth $\lceil \frac{p-1}{4} \rceil$. For example, we obtain tridiagonal matrices for $p = 3$ or $p = 5$:

$$A(3) = \begin{pmatrix} 6 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 6 \end{pmatrix}, \quad A(5) = \begin{pmatrix} 20 & 6 & & \\ 6 & \ddots & \ddots & \\ & \ddots & \ddots & 6 \\ & & 6 & 20 \end{pmatrix}.$$

$A(p)$ is strictly diagonally dominant for $p = 1, 3, 5, 7$ and therefore invertible. For higher degrees, the regularity of $A(p)$ must be shown differently. If $A(p)$ is regular, we infer $c_{n,i} = 0$ for all $i \in I_n$, implying $f_2 = f_1 = 0$, which completes the proof for the common cases $p \in \{1, 3, 5, 7\}$. □

Proposition 3. *Let $n \in \mathbb{N}$. If we choose ξ as in* (5) *and restrict all functions involved to $D = [\xi_p, \xi_m]$, then $\bigoplus_{l=1}^{n} W_l^p = S_{\boldsymbol{\xi}}^p = V_n^p$.*

Proof. We already mentioned that $W_n^p \subseteq S_{\boldsymbol{\xi}}^p = V_n^p$ holds. When restricting all of the basis functions $\varphi_{l,i}^p$ to $D = [\xi_p, \xi_m]$, $W_l^p \subseteq S_{\boldsymbol{\xi}}^p$ also holds for smaller levels $l < n$: Each basis function $\varphi_{l,i}^p$, $i \in I_l$, is continuous on D, a polynomial of degree $\leq p$ on every knot interval of $\boldsymbol{\xi}$ (due to p odd) and at the knots themselves at least $(p-1)$ times continuously differentiable. From proposition 1 it follows $\varphi_{l,i}^p \in S_{\boldsymbol{\xi}}^p$ and hence $W_l^p \subseteq S_{\boldsymbol{\xi}}^p$ for $l \leq n$. Consequently, $\bigoplus_{l=1}^{n} W_l^p \subseteq S_{\boldsymbol{\xi}}^p$ and with a dimension argument,

$$\dim \bigoplus_{l=1}^{n} W_l^p = \sum_{l=1}^{n} |I_l| = \sum_{l=1}^{n} 2^{l-1} = 2^n - 1 = m = \dim S_{\boldsymbol{\xi}}^p \,,$$

we obtain $\bigoplus_{l=1}^{n} W_l^p = S_{\boldsymbol{\xi}}^p = V_n^p$, which proves the proposition. □

2.4 Modified and Multivariate Hierarchical B-Splines

In $[0, 1] \setminus D$, linear combinations of hierarchical B-splines experience an unnatural decay towards the boundary of $[0, 1]$. As a side effect, this can result in overshoots

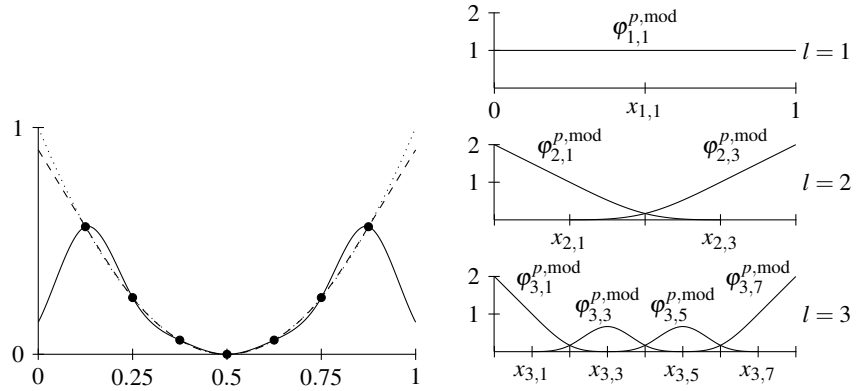

Fig. 2: *Left*: interpolation of the parabola $y = 4(x-0.5)^2$ (*dotted line*) with unmodified (*solid*) and modified B-splines (*dashed*) for $p = 3$. *Right*: modified hierarchical B-splines of degree $p = 3$ and level $l = 1, 2, 3$.

of the linear combinations even when interpolating simple polynomials (see Fig. 2, left). Note that Fig. 2 does not contradict Prop. 3, as the $(p-1)/2$ leftmost and the $(p-1)/2$ rightmost grid points, where compliance with the interpolation condition is enforced, do not lie in $D = [\xi_p, \xi_m]$.

Grids with boundary points can help, but they spend proportionally too few points in the interior, most notably in higher dimensions. To overcome this difficulty, we modified the B-spline of level 1 and the first and last B-splines of higher levels [28],

$$\varphi_{l,i}^{p,\mathrm{mod}}(x) := \begin{cases} 1 & \text{if } l = 1, i = 1, \\ \psi_l^p(x) & \text{if } l > 1, i = 1, \\ \psi_l^p(1-x) & \text{if } l > 1, i = 2^l - 1, \\ \varphi_{l,i}^p(x) & \text{otherwise,} \end{cases} \qquad \psi_l^p := \sum_{k=0}^{\lceil (p+1)/2 \rceil} (k+1)\varphi_{l,1-k}^p \ ,$$

adding B-splines which have their maximum outside of $[0, 1]$ (see Fig. 2, right). Due to the relation

$$x = \sum_{k \in \mathbb{Z}} \left(k + \frac{p+1}{2} \right) b^p(x-k) \ , \quad x \in \mathbb{R} \ ,$$

which can be proven with Marsden's identity [15], we infer for degree $1 \le p \le 4$

$$\psi_l^p(x) = 2 - \frac{x}{h_l} \ , \quad x \in \left[0, \frac{5-p}{2} h_l \right] \ .$$

In other words, modified B-splines with index $i \in \{1, 2^l - 1\}$ extrapolate linearly towards the boundary of $[0, 1]$, providing meaningful values for linear combinations

near the boundary. For higher degrees $p > 4$, the deviation from $2 - x/h_l$ is hardly visible, as the second derivative at the boundary is numerically small.

Hierarchical B-splines of one dimension are generalized to the d-dimensional case as usual by a tensor product approach,

$$\varphi^{\mathbf{p}}_{\mathbf{l},\mathbf{i}}(\mathbf{x}) := \prod_{t=1}^{d} \varphi^{p_t}_{l_t,i_t}(x_t) \,, \quad \mathbf{x}_{\mathbf{l},\mathbf{i}} := (x_{l_1,i_1}, \ldots, x_{l_d,i_d}) \,,$$

using multi-indices $\mathbf{l}, \mathbf{i} \in \mathbb{N}^d$, $\mathbf{p} \in \mathbb{N}_0^d$, and $\mathbf{x} \in [0,1]^d$. We define d-variate nodal and hierarchical subspaces by

$$\begin{aligned} V^{\mathbf{p}}_{\mathbf{l}} &:= \operatorname{span}\{\varphi^{\mathbf{p}}_{\mathbf{l},\mathbf{i}} \mid \forall_{t=1,\ldots,d} \colon i_t = 1, \ldots, 2^{l_t} - 1\} \,, \\ W^{\mathbf{p}}_{\mathbf{l}} &:= \operatorname{span}\{\varphi^{\mathbf{p}}_{\mathbf{l},\mathbf{i}} \mid \mathbf{i} \in I_{\mathbf{l}}\} \,, \quad I_{\mathbf{l}} := I_{l_1} \times \cdots \times I_{l_d} \,, \end{aligned}$$

Tensor products of linearly independent functions are linearly independent, i.e. the generating sets of $V^{\mathbf{p}}_{\mathbf{l}}$ and $W^{\mathbf{p}}_{\mathbf{l}}$ are their bases, respectively. By using an analogous d-variate formulation of Prop. 1 (defining $S^{\mathbf{p}}_{\boldsymbol{\xi}}$ appropriately), it follows as above that

$$V^{\mathbf{p}}_{\mathbf{n}} = S^{\mathbf{p}}_{\boldsymbol{\xi}} = \bigoplus_{\mathbf{l} \leq \mathbf{n}} W^{\mathbf{p}}_{\mathbf{l}} \,,$$

if we choose the d knot sequences $\boldsymbol{\xi} = (\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_d)$, $\boldsymbol{\xi}_t = (\xi_{t,0}, \ldots, \xi_{t,m_t+p_t})$, accordingly to (5) and restrict all functions to $D = [\xi_{1,p_1}, \xi_{1,m_1}] \times \cdots \times [\xi_{d,p_d}, \xi_{d,m_d}]$. The sparse grid space $V^{\mathbf{p},\mathrm{s}}_n$ of level n can now be constructed as usual by

$$V^{\mathbf{p},\mathrm{s}}_n := \bigoplus_{\|\mathbf{l}\|_1 \leq n+d-1} W^{\mathbf{p}}_{\mathbf{l}} \,.$$

We get the familiar piecewise linear sparse grid space with $\mathbf{p} = \mathbf{1} := (1, \ldots, 1)$. Sparse grid spaces consisting of modified B-splines are defined similarly.

3 Adaptive Grid Generation

The surrogate, which replaces the objective function f to be minimized, is defined as the interpolant on an adaptively generated sparse grid. The most widespread method is the refinement of the grid points whose hierarchical surpluses $\alpha_{\mathbf{l},\mathbf{i}}$ (in the piecewise linear basis) have the highest absolute value [29]. However, this approach does not generate more points close to minima than elsewhere. Instead we want to use a slightly modified version of the Novak-Ritter refinement criterion [11, 16, 26] which was specifically made for optimization (initially for hyperbolic cross points, which are closely related to sparse grids).

The method works iteratively: We start with an initial regular sparse grid, e.g. of level 3. Let $X = \{\mathbf{x}_k := \mathbf{x}_{\mathbf{l}_k,\mathbf{i}_k} \mid k = 1, \ldots, N\} \subset \mathbb{R}^d$ be the current sparse grid at

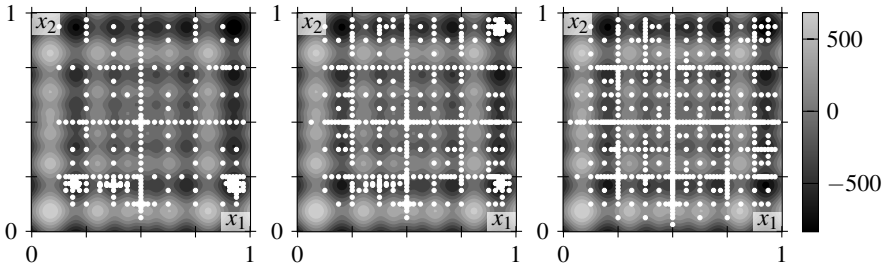

Fig. 3: Adaptive grid generation of $N = 500$ points with Novak-Ritter's refinement criterion for the Schwefel 2D test function with $\gamma = 0.6$ (*left*), $\gamma = 0.8$ (*center*), and $\gamma = 0.95$ (*right*). The global minimum lies in the upper right corner.

the beginning of an iteration. The Novak-Ritter criterion selects one point $\mathbf{x}_{k^*}$ of X, which is then refined by inserting the $2d$ neighbors into the grid. The neighbors of $\mathbf{x}_{\mathbf{l},\mathbf{i}}$ in the t-th dimension have level $l_t + 1$ and index $2i_t \pm 1$ in dimension t and the same level and index in all other dimensions. If one of the neighbors already exists in X, then the first higher-order neighbor, which is not in X, is inserted instead. The neighbor of order m has level $l_t + m$ and index $2^m i_t \pm 1$ in the t-th dimension. Therefore, in each iteration exactly $2d$ points are inserted. The grid generation is completed when a specific number $N \in \mathbb{N}$ of grid points, which is due to the overall effort that can be invested, has been reached.

The Novak-Ritter criterion determines $\mathbf{x}_{k^*}$ as follows: Associate with each grid point $\mathbf{x}_k = \mathbf{x}_{\mathbf{l}_k,\mathbf{i}_k}$ three scalars $\|\mathbf{l}_k\|_1$, d_k and r_k. $\|\mathbf{l}_k\|_1$ is the sum of the levels, as usual. d_k represents the *degree* of $\mathbf{x}_k$, the number of times the point was already selected (initially 0). r_k is the *rank* of $\mathbf{x}_k$ defined by $r_k := |\{k' \mid f(\mathbf{x}_{k'}) \leq f(\mathbf{x}_k)\}|$, i.e., the point with the smallest objective function value gets rank 1, the next bigger one gets rank 2 etc. Now, k^* is selected as the index for which the *quality* β_{k^*} is minimal:

$$\beta_k := (\|\mathbf{l}_k\|_1 + d_k + 1)^\gamma \cdot r_k^{1-\gamma} \ .$$

We added 1 to the base of the first factor to prevent ambiguities if levels and degree sum up to zero (possible when working with boundary grids). $\gamma \in [0,1]$ is the adaptivity parameter with $\gamma = 0$ meaning pure adaptivity and $\gamma = 1$ leading to an unadaptive algorithm with the function values being irrelevant. γ must be chosen carefully to allow the algorithm to explore the whole domain $[0,1]^d$, while refining in promising regions sufficiently well to increase the accuracy of the sparse grid interpolant (see Fig. 3 for an example). Its best choice depends a lot on the characteristics of the objective function at hand. As a compromise, we choose a priori $\gamma = 0.85$ for all applications. Note that for γ large enough, the set X of generated grid points gets dense in $[0,1]^d$ in the limit $N \to \infty$, implying that for arbitrary objective functions f, a global optimum will be found eventually (if N and γ are chosen large enough).

4 Implementation

After adaptively generating the grid as the first step, we replace the objective function f by the sparse grid interpolant $\tilde{f}$ and then apply existing optimization algorithms to $\tilde{f}$. In this section, we want to elaborate on the two remaining steps.

4.1 Hierarchization

The interpolant $\tilde{f}$ on the sparse grid $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^d$ is defined by the linear combination of the basis functions $\varphi_k := \varphi^{\mathbf{p}}_{\mathbf{l}_k,\mathbf{i}_k}$ (either modified or not) interpolating f in the grid points $\mathbf{x}_k := \mathbf{x}_{\mathbf{l}_k,\mathbf{i}_k}$. This leads to a linear system with the variables $\alpha_1, \ldots, \alpha_N \in \mathbb{R}$ (*hierarchical surpluses*):

$$\tilde{f}(\mathbf{x}) := \sum_{k=1}^{N} \alpha_k \varphi_k(\mathbf{x}) \,, \quad \tilde{f}(\mathbf{x}_j) = f_j := f(\mathbf{x}_j) \,, \quad j = 1, \ldots, N \,. \tag{8}$$

The basis transform $\mathbf{f} \mapsto \boldsymbol{\alpha}$ is usually called *hierarchization*. For $\mathbf{p} = \mathbf{1}$, the linear system can efficiently be solved via the unidirectional principle [28]: It suffices to apply one-dimensional hierarchization operators to all one-dimensional subgrids (so-called poles) of X in each dimension, working with updated values. However, the principle only works if every pole is a proper 1D sparse grid: Every hierarchical ancestor of a grid point of X must be in X, too. This requirement has severe effects, because every grid point insertion by Novak-Ritter's algorithm in the grid generation phase implies the recursive insertion of all (indirect) hierarchical ancestors. The number of the ancestors to be inserted grows rapidly with the number d of dimensions. For example, performing Novak-Ritter's grid generation for the well-known Rosenbrock function and $\gamma = 0.8$ leads to 1128, 223, 61, 33, 16 refinement iterations for $d = 2$, 3, 4, 5, 10 respectively, stopping when $N = 10000$ points have been generated. As a result, for $d = 2$ only 45% of the maximum possible number $N/(2d)$ of iterations has been exploited, for $d = 3$ only 13% and for $d \geq 4$ less than 5%. The ancestors often lie at uninteresting places, wasting valuable evaluations of the objective function.

For higher B-spline degrees $\mathbf{p} > \mathbf{1}$, the unidirectional principle is in general not applicable anyway. This is due to the fact that in this case basis functions do not vanish at all grid points of coarser levels (unlike in the piecewise linear case). For our purposes with limited overall effort N, it is sufficient to solve the linear system (8) directly or iteratively. We thus do not have to generate additional hierarchical ancestors, allowing to exhaust the full number $N/(2d)$ of iterations in the grid generation phase. In general, the linear system is asymmetric and its sparsity structure depends on how many grid points are contained in the supports of the basis

functions. For lower numbers d of dimensions and lower B-spline degrees $\mathbf{p}$, the system is sparse, which allows a solution by adequate solvers in reasonable time[1].

4.2 Global Optimization

The constructed sparse grid B-spline interpolant $\tilde{f}$ is $(p_t - 1)$ times partially continuously differentiable in dimension $t = 1, \ldots, d$. For $p_t > 1$, we can apply gradient-based optimization methods to $\tilde{f}$ without having to evaluate f additional times. We used local gradient-based algorithms [25], particularly the gradient descent method, the nonlinear conjugate gradient method with Polak-Ribière coefficients (NLCG, [30]), Newton's method, BFGS, and Rprop [32], in addition to the local gradient-free Nelder-Mead algorithm (NM, [24]). We also used Storn's and Price's Differential Evolution (DE, [35]), using a population size of $10d$, as a non-local gradient-free method. To prevent being stuck in local minima, we globalized all mentioned local algorithms by using a multi-start approach with $m := \min(10d, 100)$ uniformly random starting points (i.e. m parallel calls of the local algorithm, each with $1/m$ of the permitted function evaluations). The gradient-free techniques NM and DE are not only used for the global optimization of the surrogate $\tilde{f}$, but also to directly optimize the objective function and the standard piecewise linear interpolant (case of $\mathbf{p} = \mathbf{1}$), as we will explain later.

Our optimization algorithm to solve problem (1) for a given objective function $f\colon [0,1]^d \to \mathbb{R}$ works as follows, assuming that the adaptivity parameter $\gamma \in [0,1]$ of the grid generation and the maximal number $N \in \mathbb{N}$ of evaluations of f is given:

1. Generate the grid $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $n \leq N$, using the adaptive Novak-Ritter method. This requires to evaluate the objective function n times, obtaining $f_j = f(\mathbf{x}_j)$.
2. Solve the linear system (8) to get the interpolant $\tilde{f}\colon [0,1]^d \to \mathbb{R}$.
3. Optimize the interpolant: First, find $\mathbf{y}_0 := \mathbf{x}_{j^*}$ with $j^* := \arg\min_j f_j$. Then apply all gradient-based methods to $\tilde{f}$ with $\mathbf{y}_0$ as starting point. Let $\mathbf{y}_1$ be the resulting point with the minimal objective function value. Now use the globalized local algorithms and DE applied to $\tilde{f}$; let $\mathbf{y}_2$ be the best (i.e. in terms of the f value) point of the results. Take the point of $\{\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2\}$ with the smallest f value as approximation $\mathbf{x}^*_{\text{opt}}$ to the optimum $\mathbf{x}_{\text{opt}}$ of f.

The third step requires (beyond the n evaluations during grid generation) some, say c, additional evaluations of f. Thus, a total of up to $N + c$ evaluations have to be performed during the algorithm. To keep the overall effort to at most N one can enforce $n \leq N - c$ in step 1. Because $\mathbf{y}_0$ is taken into account when determining $\mathbf{x}^*_{\text{opt}}$, the returned optimum is the point with the smallest objective function value of all points where f was evaluated during the algorithm.

[1] We used Gmm++ ([31], GMRES) and UMFPACK ([6], LU factorization) for sparse systems and Armadillo ([33], LU factorization) and Eigen ([13], QR Householder factorization) for full systems.

Table 1: Employed test functions in two and arbitrary dimensions with abbreviations in bold.

Name	Domain	$\mathbf{x}_{\text{opt}}$	$f(\mathbf{x}_{\text{opt}})$	Reference
Branin	$[-5, 10] \times [0, 15]$	$(-\pi, 12.275)$, $(\pi, 2.275)$, $(9.42478, 2.475)$	0.397887	[18, Branin RCOS]
Eggholder	$[-512, 512]^2$	$(512, 404.2319)$	-959.6407	[37, $F101$]
Rosenbrock	$[-5, 10]^2$	$(1, 1)$	0	[38]
Ackley	$[-1, 9]^d$	$\mathbf{0}$	0	[38]
Rastrigin	$[-2, 8]^d$	$\mathbf{0}$	0	[38]
Schwefel	$[-500, 500]^d$	$420.9687 \cdot \mathbf{1}$	$-418.9829d$	[38]

We compared our optimization algorithm to the following common optimization techniques:

- Optimization of the piecewise linear sparse grid interpolant. Therefore we proceed as above with B-spline degree $\mathbf{p} = \mathbf{1}$, using only the gradient-free methods NM (globalized) and DE to optimize $\tilde{f}$. The best of the two results is called $\mathbf{x}'_{\text{opt}}$.
- Direct optimization of the objective function f (with globalized NM) without using sparse grids, but with only N evaluations permitted. The resulting optimum is called $\mathbf{x}''_{\text{opt}}$.

5 Numerical Results and Applications

In this section, we want to review our optimization method with the aid of artificial test functions and real-world applications. As standard parameters, we used modified B-splines of degree $\mathbf{p} = \mathbf{5}$ as basis functions and $\gamma = 0.85$ as adaptivity.

5.1 Test Functions

We studied a wide variety of test functions for different dimensionalities [36]. In the following, we present three functions for two dimensions and three functions defined in arbitrary dimensions. The domain of each function is transformed to the unit hypercube $[0, 1]^d$ by an affine transformation. Additionally, some of the domains were translated and/or scaled first (when compared to the literature) to make sure that the optimum does not lie at the center of the domain. Otherwise sparse grid approaches would have been in advantage, because they spend proportionally few points near the corners of $[0, 1]^d$. In Table 1 we give the domains, minimal points, and corresponding function values (all before parameter scaling and translation). The two-dimensional test functions are shown in Fig. 4. All functions were perturbed in the parameter domain by a small pseudo-random normally distributed translation

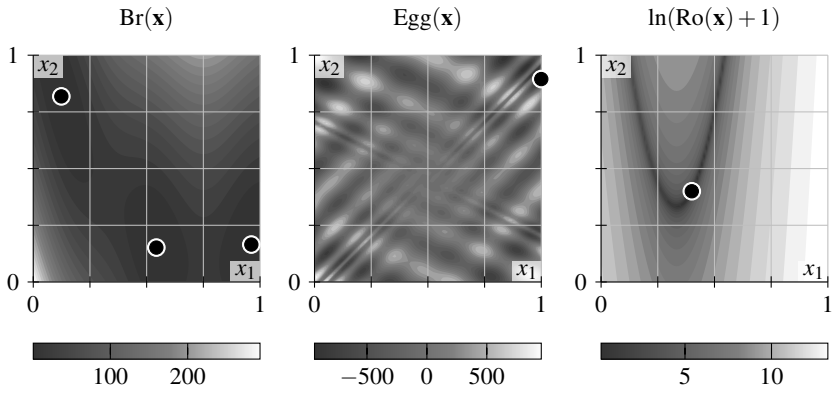

Fig. 4: Bivariate test functions with location of the minimal points (after normalization of the domain to $[0,1]^2$).

(standard deviation 0.01), while making sure that the optima of the perturbed functions still lie in the original domains. To increase the validity of our results, all results shown are the mean of five passes with different perturbations.

The plots depicted in Figs. 5, 6, and 8 show the difference between approximated and true minimal value of the objective function over the number N of evaluations of f. Each test function is associated with three lines: The solid lines represent the performance of our optimization algorithm with result $\mathbf{x}^*_{\text{opt}}$, the dotted lines display the performance of the optimization of the piecewise linear sparse grid interpolant with Nelder-Mead (NM) and Differential Evolution with result $\mathbf{x}'_{\text{opt}}$, and the dashed lines show the optimization of the objective function using globalized NM with result $\mathbf{x}''_{\text{opt}}$. Note that in the notation of the last section, we have $f(\mathbf{x}'_{\text{opt}}) \leq f(\mathbf{y}_0)$, implying that the gain of our method compared to the best Ritter-Novak grid point $\mathbf{y}_0$ is at least $f(\mathbf{x}'_{\text{opt}}) - f(\mathbf{x}^*_{\text{opt}})$ (difference between solid and dotted lines).

As can be seen in Fig. 5, functions like Branin and Rosenbrock are generally easier to optimize since they feature few local minima and few oscillations. Such functions can be approximated by B-spline linear combinations very well, which leads to a considerable advantage for the B-splines compared to the standard piecewise linear basis. Our method even beats the globalized NM method (dashed lines) for most of the test functions. For the Eggholder function, fast convergence of all methods is impeded not only by high oscillations and many local minima, but also by the fact that the global optimum lies on the boundary of the domain (before perturbing).

Figure 6 shows that for higher-dimensional functions, the problem of optimization rapidly becomes very difficult. With increasing d, the rate of convergence becomes substantially slower. We note that for moderate dimensions $d \in \{3,4\}$, B-splines can provide a significant boost in the performance compared to the piecewise linear basis. For higher dimensions $d \geq 6$, both sparse grid approaches (B-splines and piecewise linear) perform better in general in comparison to the globalized NM optimization technique. It can be seen that Rastrigin is a very tough function as it

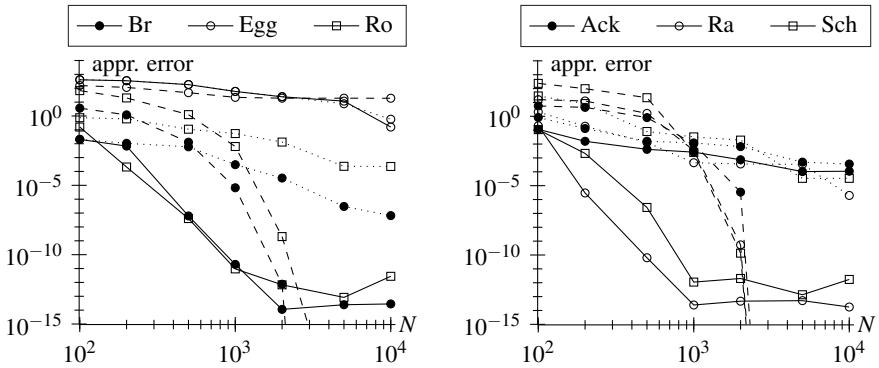

Fig. 5: Approximation errors $f(\mathbf{x}^*_{\text{opt}}) - f(\mathbf{x}_{\text{opt}})$ (*solid lines*), $f(\mathbf{x}'_{\text{opt}}) - f(\mathbf{x}_{\text{opt}})$ (*dotted*), and $f(\mathbf{x}''_{\text{opt}}) - f(\mathbf{x}_{\text{opt}})$ (*dashed*) over the number N of evaluations for different test functions with $d = 2$ variables.

exhibits numerous local minima in a neighborhood of the global minimum, all with a similar function value. This can lead to a non-monotonous error decay of our method as seen in the plots for $d \in \{3,4,6\}$, since the global minimum $\mathbf{x}^*_{\text{opt}}$ of the interpolant occasionally does not match with the actual optimum $\mathbf{x}_{\text{opt}}$ of the objective function.

5.2 Model of a DC Motor

As an example application we study an inverse problem of a simple DC (direct current) motor. If we denote with θ and ω the angular position and velocity in rad and rad/s, respectively, then an idealized model (with zero disturbance and torque) of the motor can be deduced [22], obtaining the linear state-space representation

$$\dot{\theta}(t) = \omega(t)\,, \quad \dot{\omega}(t) = -\frac{1}{\tau}\omega(t) + \frac{k}{\tau}U(t)\,, \quad \theta(0) = \theta_0\,, \quad \omega(0) = \omega_0\,, \tag{9}$$

with (θ,ω) as both state and output, input voltage U, and motor-dependent constants τ (*time constant*) and k (*steady-state gain*). It can easily be seen that for constant inputs $U \equiv U_0$, ω then satisfies $\omega(t) = (\omega_0 - kU_0)e^{-t/\tau} + kU_0$.

We have generated artificial data for the motor sampled at t_j with 10 Hz over a time span of 60 s (see Fig. 7). The sampled data (θ_j,ω_j) was generated by adding an artificial Gaussian noise with standard deviation 0.1 rad and 0.1 rad/s to the solution (θ,ω) of (9) for the generated voltage data U_j, respectively. Our goal is now to determine $(\tau,k,\theta_0,\omega_0)$ so that the resulting solution (θ,ω) of (9) minimizes the ℓ^2 norm $(\sum_j(\omega(t_j) - \omega_j)^2)^{1/2}$ of the difference of experimental and simulated angular velocity. The error functional does not need to take θ into account, as $\dot{\theta} = \omega$ should imply a good match of θ and θ_j and including θ in the error functional would lead to worse results due to overfitting. In total, this leads to a 4D optimization problem.

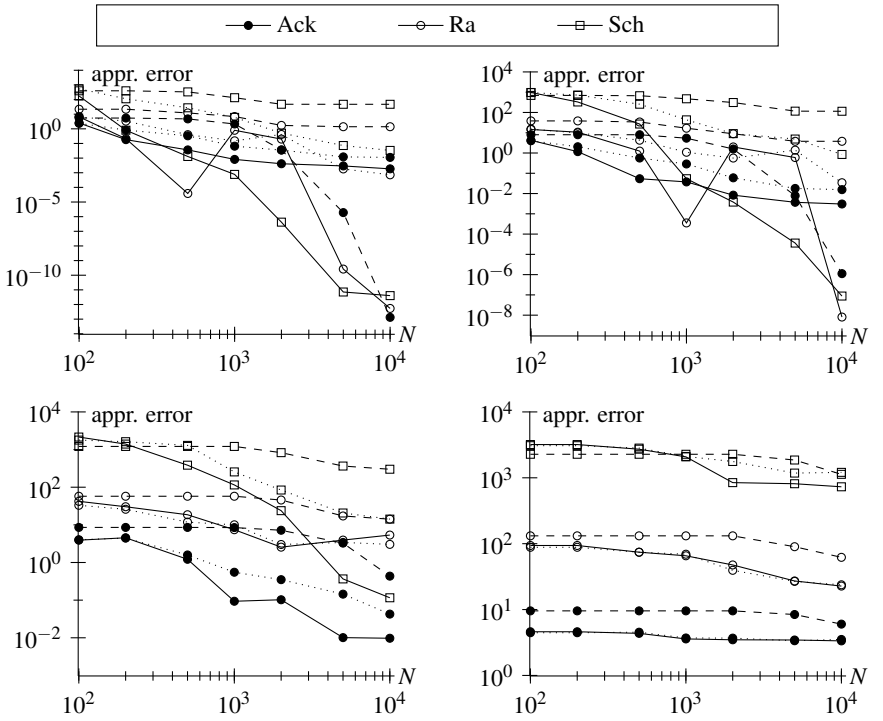

Fig. 6: Approximation errors $f(\mathbf{x}^*_{\text{opt}}) - f(\mathbf{x}_{\text{opt}})$ (*solid lines*), $f(\mathbf{x}'_{\text{opt}}) - f(\mathbf{x}_{\text{opt}})$ (*dotted*), and $f(\mathbf{x}''_{\text{opt}}) - f(\mathbf{x}_{\text{opt}})$ (*dashed*) over the number N of evaluations for different test functions with $d = 3$, $d = 4$, $d = 6$, and $d = 10$ variables (*from top left to bottom right*).

Before we can start optimizing, we need to determine reasonable parameter intervals. Looking at the data, we guess $\theta_0 \in [-2\,\text{rad}, 2\,\text{rad}]$ and $\omega_0 \in [-2\,\text{rad/s}, 0\,\text{rad/s}]$. We guess τ by looking at the half-life period $\tau \ln 2$ of the transient response after a change in input voltage polarity. This period roughly equals 0.3 s which would imply $\tau \approx 0.43\,\text{s}$, justifying the assumption of $\tau \in [0.2\,\text{s}, 0.6\,\text{s}]$. For the interval of k, we look at the steady-state angular velocity kU_0, which is around 1 rad/s, leading with $U_0 = 5\,\text{V}$ to $k \approx 0.2\,\text{rad/(Vs)}$. Therefore, we generously set the interval to $k \in [0.1\,\text{rad/(Vs)}, 0.4\,\text{rad/(Vs)}]$.

Figure 8 (log-log plot as above) shows the performance of our optimization method as well as the performance of the piecewise linear basis and the direct optimization of the objective function with the globalized Nelder-Mead method (NM). We see that the objective function is sufficiently smooth to allow good B-spline interpolation, leading to a faster convergence compared to the piecewise linear basis and to the classical NM technique, with convergence beginning at $N = 100$ grid points. Our method exactly finds the optimal parameters $\tau_{\text{opt}} = 0.496\,\text{s}$, $k_{\text{opt}} = 0.214\,\text{rad/(Vs)}$,

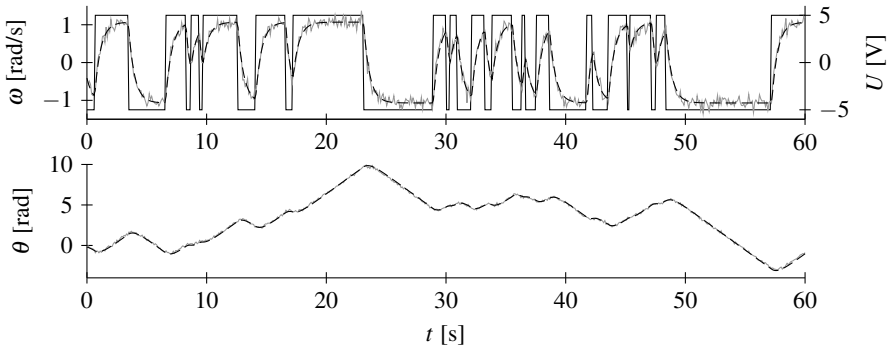

Fig. 7: Input voltage U (*solid black line*), artificial motor data (θ_j, ω_j) (*solid gray*), and simulated data (θ, ω) of optimal model (*dashed*).

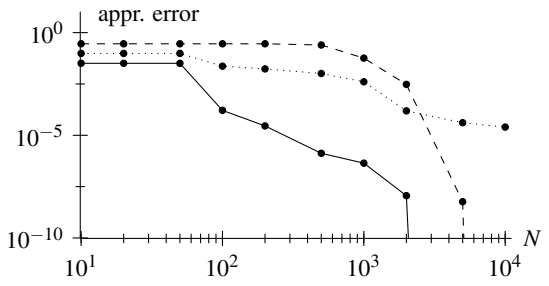

Fig. 8 Approximation errors $f(\mathbf{x}^*_{\text{opt}}) - f(\mathbf{x}_{\text{opt}})$ (*solid line*), $f(\mathbf{x}'_{\text{opt}}) - f(\mathbf{x}_{\text{opt}})$ (*dotted*), and $f(\mathbf{x}''_{\text{opt}}) - f(\mathbf{x}_{\text{opt}})$ (*dashed*) over the number N of evaluations for the objective function resulting from the DC motor.

$\theta_{0,\text{opt}} = -0.198\,\text{rad}$, $\omega_{0,\text{opt}} = -0.411\,\text{rad/s}$ with error functional value 2.693 using just $N = 1000$ evaluations.

5.3 Shape Optimization with Homogenization

As another application, we have also employed B-splines on sparse grids in a two-scale shape optimization setting [17]. Classical approaches in shape optimization share the drawback of severely restricting the set of feasible topologies before starting to optimize, which leads to homeomorphic results. However, one often does not know the topology of the optimal shape beforehand. For example, if we take the cantilever in Fig. 9 (left), which is fixed at one side and deformed by a force $\mathbf{F}$ at the other side, and want to determine the cantilever shape which minimizes displacement, we do not know if we should use one, two, or even more crossbars, if the cantilever is only allowed to take up a specific volume.

In the homogenization approach [2], the whole domain Ω is potentially filled with material with varying density $\varrho(\mathbf{x}) \in [0,1]$. For a fixed force $\mathbf{F}$, we want ϱ to minimize the *compliance function* $J(\mathbf{u}_\varrho)$ with a volume constraint,

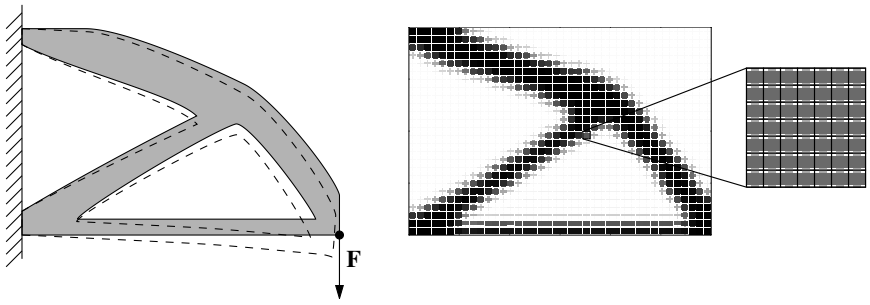

Fig. 9: *Left*: shape optimization of a 2D cantilever fixed at the left side with force $\mathbf{F}$ resulting in a deformation. *Right*: optimized model consisting of macro cells (compounds of micro cells).

$$\min_{\varrho} J(\mathbf{u}_\varrho) \quad \text{s.t.} \ \frac{1}{|\Omega|}\int_\Omega \varrho(\mathbf{x})\mathrm{d}\mathbf{x} \le \varrho_{\max}\,, \quad J(\mathbf{u}_\varrho) := \int_\Omega \mathbf{F}\cdot\mathbf{u}_\varrho(\mathbf{x})\mathrm{d}\mathbf{x}\,, \tag{10}$$

where $\varrho_{\max} \in [0,1]$ is the maximal total density and $\mathbf{u}_\varrho(\mathbf{x})$ is the displacement in $\mathbf{x} \in \Omega$ (depending on the density ϱ), which can be determined by the finite element method (FEM, [14,17]). In the following, we restrict ourselves to the two-dimensional case, but the 3D case could also be handled in an analogous way.

We choose a two-scale approach as Hübner [17]: First, we discretize the domain in $(N_1 \times N_2)$ *macro cells*. Each macro cell k is a compound of tiny periodic and identical *micro cells* (see Fig. 9), which are formed by shearing an axis-parallel cross with thicknesses a_k, b_k by an angle $\varphi_k \in (-\pi/2, \pi/2)$, resulting in parallelogram-shaped micro cells. If a_k, b_k, φ_k are known, one can compute the symmetrical elasticity tensor $E_k = (E_{k,i,j})_{1\le i,j\le 3} \in \mathbb{R}^{3\times 3}$ of macro cell k by the FEM (*micro problem*). When we know all the E_k, we can compute $\mathbf{u}_\varrho$ with the density ϱ given by the $3N_1N_2$ parameters a_k, b_k, φ_k, $k = 1,\dots,N_1N_2$, again by solving a FEM problem (*macro problem*). Our goal is finding a combination of the $3N_1N_2$ parameters which solves (10). Because every evaluation of $J(\mathbf{u}_p)$ triggers the solution of a macro problem (which depends on N_1N_2 micro problems), a single evaluation is very expensive. As in [17], we use the FEM solver CFS++ [20] which provides interfaces to established optimizers like SNOPT [12], requiring gradients that have to be approximated by finite differences, since they are not available explicitly.

To increase performance, Hübner [17] precomputes values of the elasticity tensor $E\colon [0,1]^3 \to \mathbb{R}^{3\times 3}$ for different combinations of normalized parameters a, b, φ and replaces the task of solving micro problems by the evaluation of an interpolant $\tilde{E}\colon [0,1]^3 \to \mathbb{R}^{3\times 3}$ of E. Full grid interpolation approaches are possible, but more complex micro cell models (e.g. in 3D) will feature more parameters, which would imply a prohibitively large precomputational effort in terms of both computing time and storage space. In [17], also the suitability of sparse grid interpolation with piecewise linear functions was studied. However, these lead to problems because of the discontinuous derivatives calculated by the gradient-based optimizer. It seems

Table 2: Results for the two-parameter case (where $\varphi_k = 0$ is fixed) and the three-parameter case with optimal compliance function value, number of iterations, and time needed by SNOPT for optimization without precomputation of the elasticity tensors ($\rho_{\max} = 0.5$).

#Param.	Grid	Basis	Obj. Fcn.	#Iter.	Time
2	full, level 6	piecewise tricubic interpolation	42.85	170	128 s
2	sparse, level 7	modified piecewise linear	43.00	704	546 s
2	sparse, level 7	modified cubic B-splines	42.80	377	299 s
3	full, level 6	piecewise tricubic interpolation	41.86	1307	27 min
3	sparse, level 8	modified piecewise linear	41.95	4203	163 min
3	sparse, level 8	modified cubic B-splines	41.26	1483	139 min

natural to employ B-spline basis functions instead, as the function E to be interpolated is supposedly relatively smooth. Additionally, the optimizer can use exact derivatives of the interpolant.

As an example, we consider the cantilever in Fig. 9 ("example A" in [17]), where the domain consists of $40 \cdot 26$ macro cells. The emerging optimization problem with $3 \cdot 40 \cdot 26 = 3120$ variables is solved by CFS++/SNOPT and visualized as in Fig. 9 (right), where each macro cell is represented by a single micro cell cross. We compare the performance of B-splines to the piecewise linear basis and to piecewise tricubic interpolation [21] on the full grid of level 6.

First, we examine the B-spline sparse grid interpolation method when all $\varphi_k = 0$ are fixed and only a_k and b_k are optimized. In this case, four elasticity tensor entries $E_{1,3} = E_{2,3} = 0 = E_{3,2} = E_{3,1}$ vanish. This makes a significant difference in complexity as there are only four non-trivial entries of E left and only two variables per interpolant, resulting in an optimization problem of $2 \cdot 40 \cdot 26 = 2080$ variables. If we look at the results in Table 2 (top half), we observe that sparse grid interpolation produces similar objective function values as tricubic interpolation on the full grid. However, using a full grid interpolant leads to better convergence and faster termination. With sparse grid B-splines (see Fig. 10, left) we even get a smaller compliance function value than the full grid interpolation.

Second, we look at the general case where all $3N_1N_2$ are to be optimized. The optimization now takes much more time since there are more tensor entries to be interpolated, more partial derivatives to be evaluated, and more unknown optimization parameters. But as we have more degrees of freedom, the obtained objective function values (cf. Table 2, bottom half) are slightly better than in the case of $\varphi_k = 0$. If we compare the visualizations in Fig. 10, we note the exploitation of the additional degrees of freedom as the incline of the crossbar is more gentle in the case of three parameters per micro cell.

For the trivariate sparse grid interpolants, it is not sufficient to discretize $[0,1]^3$ with a regular sparse grid of level 7 due to too many oscillations of the resulting $\tilde{E}$. Even for a regular sparse grid of level 8 and piecewise linear functions, the optimizer

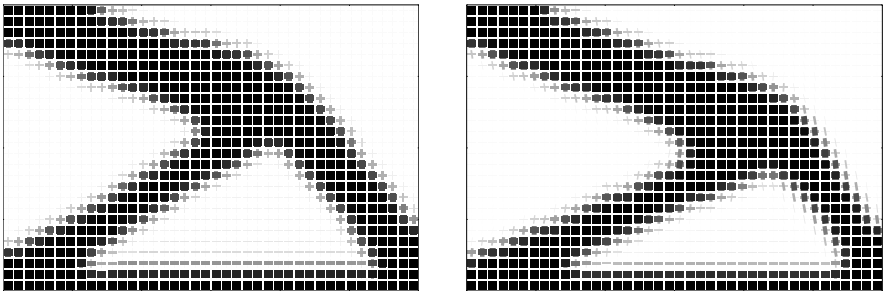

Fig. 10: Results with modified B-splines of degree $\mathbf{p} = \mathbf{3}$ and a material fraction of $\rho_{\max} = 0.5$ for fixed $\varphi_k = 0$ with a regular sparse grid of level 7 with 2815 points (*left*) and optimized φ_k with a refined sparse grid with 4439 points (*right*).

terminates early because of numerical difficulties, showing the problems introduced by approximating discontinuous derivatives by difference quotients. Again, B-splines perform quite well since they find the best parameter combination in terms of compliance function values compared to full tricubic or sparse piecewise linear interpolation, spending a multiple of the computational time of the full grid interpolation, though. This will, however, change as soon as we consider a more complicated micro cell model (e.g. in three dimensions), where we will not be able to employ full grid interpolation anymore.

Starting from the regular sparse grid of level 7, we also generated a spatially adaptive sparse grid, specifically tailored for this interpolation problem [36]. With only 4439 points (cf. 2815 and 7423 points of the regular grids of level 7 and 8, respectively), we get moderately worse results (objective function value of 42.14) for the modified cubic B-splines than for the regular sparse grid of level 8, but the optimization then only takes 84 minutes. Additionally, we have not taken into account the precomputation time to create the elasticity tensor data at the sparse grid interpolation points, which, accordingly to the smaller number of grid points, would be smaller, too (compared to the level 8 regular grid). Of course, the precomputation effort for the full grid of level 6 is much larger as it needs 258,048 data points.

6 Conclusion

We constructed a surrogate-based optimization approach using B-splines on sparse grids. After proving their linear independence and studying the direct sum of hierarchical subspaces, we used an adaptive grid generation method by Novak-Ritter to generate spatially adaptive sparse grids with adjustable adaptivity $\gamma \in [0,1]$. Finally, we successfully employed our new optimization method to various artificial test functions and real-world examples. The new method works well for smooth, moderately dimensioned objective functions without high-frequency oscillations. We

would like to mention we have applied our method to a lot more test functions (e.g. Beale, Goldstein-Price, Griewank) [36], and we picked a somewhat representative subset for this work. We also studied other sparse grid types like grids with boundary points or Clenshaw-Curtis grids with non-uniform Chebyshev points. However, the modified B-splines on the standard grid without boundary points seem to exhibit the best performance for a given number of grid points.

Certainly, there is room for improvement, as we used the same fixed B-spline degree $\mathbf{p} = p \cdot \mathbf{1}$ for all of the dimensions. We could start to use different degrees p_t depending on the dimensions t. Going one step further, we could even choose $\mathbf{p}$ adaptively depending on the objective function f, to adapt to discontinuities of f or its derivatives.

Acknowledgements This work was financially supported by the Juniorprofessurenprogramm of the Landesstiftung Baden-Württemberg.

References

1. H. Bachau, E. Cormier, P. Decleva, J. E. Hansen, and F. Martín. Applications of B-splines in atomic and molecular physics. *Rep. Prog. Phys.*, 64(12):1815–1942, 2001.
2. M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Comput. Methods Appl. Mech. Eng.*, 71(2):197–224, 1988.
3. H.-J. Bungartz. *Finite Elements of Higher Order on Sparse Grids*. Habilitationsschrift, Institut für Informatik, TU München, 1998.
4. E. Cohen, R. F. Riesenfeld, and G. Elber. *Geometric Modeling with Splines: An Introduction*. A K Peters, Natick, 2001.
5. M. G. Cox. The numerical evaluation of B-splines. *IMA J. Appl. Math.*, 10(2):134–149, 1972.
6. T. A. Davis. Algorithm 832: UMFPACK V4.3–an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, 30(2):196–199, 2004.
7. C. de Boor. On calculating with B-splines. *J. Approx. Theory*, 6(1):50–62, 1972.
8. C. de Boor. Splines as linear combinations of B-splines. A survey. In G. G. Lorentz, C. K. Chui, and L. L. Schumaker, editors, *Approximation Theory II*, pages 1–47, New York, 1976. Academic Press.
9. F. Delbos, L. Dumas, and E. Echagüe. Global optimization based on sparse grid surrogate models for black-box expensive functions. http://dumas.perso.math.cnrs.fr/JOGO.pdf.
10. M. M. Donahue, G. T. Buzzard, and A. E. Rundell. Parameter identification with adaptive sparse grid-based optimization for models of cellular processes. In A. Jayaraman and J. Hahn, editors, *Methods in Bioengineering: Systems Analysis of Biological Networks*, pages 211–232. Artech House, Boston/London, 2009.
11. I. Ferenczi. Globale Optimierung unter Nebenbedingungen mit dünnen Gittern. Diploma thesis, Department of Mathematics, TU München, 2005.
12. P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM J. Optim.*, 12(4):979–1006, 2002.
13. G. Guennebaud, B. Jacob, et al. Eigen. http://eigen.tuxfamily.org/.
14. K. Höllig. *Finite Element Methods with B-Splines*. SIAM, Philadelphia, 2003.
15. K. Höllig and J. Hörner. *Approximation and Modeling with B-Splines*. SIAM, Philadelphia, 2013.
16. Y.-K. Hu and Y. P. Hu. Global optimization in clustering using hyperbolic cross points. *Pattern Recognit.*, 40(6):1722–1733, 2007.

17. D. Hübner. Mehrdimensionale Parametrisierung der Mikrozellen in der Zwei-Skalen-Optimierung. Master's thesis, Department of Mathematics, FAU Erlangen-Nürnberg, 2014.
18. M. Jamil and X.-S. Yang. A literature survey of benchmark functions for global optimisation problems. *Int. J. Math. Model. Numer. Optim.*, 4(2):150–194, 2013.
19. Y. Jiang and Y. Xu. B-spline quasi-interpolation on sparse grids. *J. Complex.*, 27(5):466–488, 2011.
20. M. Kaltenbacher. Advanced simulation tool for the design of sensors and actuators. In *Proc. Eurosensors XXIV*, volume 5, pages 597–600, 2010.
21. F. Lekien and J. Marsden. Tricubic interpolation in three dimensions. *Int. J. Numer. Methods Eng.*, 63(3):455–471, 2005.
22. L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, 2 edition, 1999.
23. C. W. McCurdy and F. Martín. Implementation of exterior complex scaling in B-splines to solve atomic and molecular collision problems. *J. Phys. B: At. Mol. Opt. Phys.*, 37(4):917–936, 2004.
24. J. A. Nelder and R. Mead. A simplex method for function minimization. *Comput. J.*, 7(4):308–313, 1965.
25. J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 1999.
26. E. Novak and K. Ritter. Global optimization using hyperbolic cross points. In C. A. Floudas and P. M. Pardalos, editors, *State of the Art in Global Optimization*, pages 19–33. Springer, US, 1996.
27. D. Pandey. Regression with spatially adaptive sparse grids in financial applications. Master's thesis, Department of Informatics, TU München, 2008.
28. D. Pflüger. *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. Verlag Dr. Hut, München, 2010.
29. D. Pflüger. Spatially adaptive refinement. In J. Garcke and M. Griebel, editors, *Sparse Grids and Applications*, Lecture Notes in Computational Science and Engineering, pages 243–262, Berlin Heidelberg, 2012. Springer.
30. E. Polak and G. Ribière. Note sur la convergence de méthodes de directions conjuguées. *Rev. Fr. Inf. Rech. Oper.*, 3(1):35–43, 1969.
31. Y. Renard and J. Pommier. Gmm++. http://download.gna.org/getfem/html/homepage/gmm/index.html.
32. M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. *Proc. 1993 IEEE Int. Conf. Neural Netw.*, 1:586–591, 1993.
33. C. Sanderson. Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments. Technical report, NICTA, Australia, 2010.
34. I. J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. *Q. Appl. Math.*, 4:45–99, 112–141, 1946.
35. R. Storn and K. Price. Differential Evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.*, 11(4):341–359, 1997.
36. J. Valentin. Hierarchische Optimierung mit Gradientenverfahren auf Dünngitterfunktionen. Master's thesis, IPVS, Universität Stuttgart, 2014.
37. D. Whitley, S. Rana, J. Dzubera, and K. E. Mathias. Evaluating evolutionary algorithms. *Artif. Intel.*, 85(1–2):245–276, 1996.
38. X.-S. Yang. *Engineering Optimization*. John Wiley & Sons, Hoboken, 2010.