A MULTIGRID METHOD FOR ADAPTIVE SPARSE GRIDS

BENJAMIN PEHERSTORFER*, STEFAN ZIMMER†, CHRISTOPH ZENGER‡, AND HANS-JOACHIM BUNGARTZ‡

Preprint December 17, 2014

Abstract. Sparse grids have become an important tool to reduce the number of degrees of freedom of discretizations of moderately high-dimensional partial differential equations; however, the reduction in degrees of freedom comes at the cost of an almost dense and unconventionally structured system of linear equations. To guarantee overall efficiency of the sparse grid approach, special linear solvers are required. We present a multigrid method that exploits the sparse grid structure to achieve an optimal runtime that scales linearly with the number of sparse grid points. Our approach is based on a novel decomposition of the right-hand sides of the coarse grid equations that leads to a reformulation in so-called auxiliary coefficients. With these auxiliary coefficients, the right-hand sides can be represented in a nodal point basis on low-dimensional full grids. Our proposed multigrid method directly operates in this auxiliary coefficient representation, circumventing most of the computationally cumbersome sparse grid structure. Numerical results on non-adaptive and spatially adaptive sparse grids confirm that the runtime of our method scales linearly with the number of sparse grid points and they indicate that the obtained convergence factors are bounded independently of the mesh width.

Key words. adaptive sparse grids, multigrid, ANOVA, Q-cycle, multi-dimensional problems

AMS subject classifications. 65F10, 65N22, 65N55

1. Introduction.

Many problems in computational science and engineering, e.g., in financial engineering, quantum mechanics, and data-driven prediction, lead to high-dimensional function approximations [23]. We consider here the case where the function is not given explicitly at selected sample points but implicitly as the solution of a high-dimensional partial differential equation (PDE). Standard discretizations on isotropic uniform grids, so-called full grids, become computationally infeasible because the number of degrees of freedom grows exponentially with the dimension. Sparse grid discretizations are a versatile way to circumvent this exponential growth [4]. If the function satisfies certain smoothness assumptions [4, Section 3.1] sparse grids reduce the number of degrees of freedom to depend only logarithmically on the dimension; however, the discretization of a PDE on a sparse grid with the (piecewise linear) hierarchical basis results in an almost dense and unconventionally structured system of linear equations. To be computationally efficient solvers must take this structure into account. We develop a multigrid method that exploits the structure of sparse grid discretizations to achieve an optimal runtime that scales linearly with the number of sparse grids points.

Multigrid methods have been extensively studied in the context of sparse grids. They are used either as preconditioners [1, 11, 12, 15, 16] or within real multigrid schemes [2, 3, 6, 26, 33]. We first note that certain problems allow for sparse grid discretizations based on pre-wavelets or multilevel frames, leading to very well-conditioned systems and thus inherently to fast solvers; see, e.g., [18, 30, 33]. We also note that there are multigrid approaches [2] that do not operate directly in the sparse grid space. They rely on the combination technique [17], where ordinary finite element spaces are combined to a sparse grid space; however, the resulting solution

*Department of Aeronautics & Astronautics, MIT, Cambridge, MA 02139
†Institute for Parallel and Distributed Systems, University of Stuttgart,70569 Stuttgart, Germany
‡Department of Informatics, Technische Universität München, 85748 Garching, Germany

is in general not guaranteed to be the sparse grid finite element solution [8, 19, 25]. Moreover, this technique is not applicable to spatially adaptive sparse grids. Adaptivity can, however, significantly extend the scope of sparse grid discretizations; see, e.g., [24, 28]. In the following, we therefore only consider direct and spatially adaptive multigrid approaches. We build on the method introduced in [26]. There, standard multigrid ingredients are assembled to a multigrid method that can cope with the sparse grid structure, but it is computationally inefficient. In its general form, which includes PDEs with variable coefficients, the coarse grid equations have right-hand sides that are expensive to compute, i.e., costs that are quadratic in the number of sparse grid points. It is proposed to approximate the operator of the PDE such that most of the right-hand sides vanish; however, convergence to the sparse grid solution is only shown for two-dimensional Helmholtz problems [26].

We pursue a different approach that relies on a novel representation of the right-hand sides of the coarse grid equations. Our representation hides most of the cumbersome sparse grid structure and is thus computationally more convenient. It is based on an analysis of variance (ANOVA) decomposition [7] of the sparse grid solution function. This decomposition leads to what we call auxiliary coefficients, which allow us to transform the right-hand sides into an ordinary nodal point basis representation on low-dimensional full grids. It is important that this is only a transformation and does not introduce any additional approximations. The key idea is that the right-hand sides in this representation are cheap to compute if the auxiliary coefficients are given. We therefore show that the auxiliary coefficients have a structure that can be exploited to avoid extensive computations. In particular, the coefficients do not need to be recomputed from scratch after a grid transfer. We derive algorithms to compute the auxiliary coefficient representation and then develop a multigrid method that directly operates on the auxiliary coefficients and thus achieves linear runtime costs in the number of sparse grid points.

In Section 2, we discuss basic properties of sparse grid discretizations and multigrid methods, and then show why the sparse grid structure leads to computationally expensive coarse grid equations. We then introduce, in Section 3, the auxiliary coefficients to transform the coarse grid equations into a computationally more convenient representation. In Section 4, we discuss some properties of the auxiliary coefficients, including their hierarchical structure, and then develop an algorithm to compute the auxiliary coefficients during grid transfers. This, finally, leads to our sparse grid multigrid method with linear runtime and storage complexity in the number of sparse grid points for arbitrary dimensional sparse grid discretizations. We demonstrate our approach numerically on non-adaptive and adaptive sparse grids in Section 5.

2. Sparse grids and multigrid.

We briefly introduce the hierarchical basis and sparse grid spaces in Section 2.1 as well as the discretization of PDEs on sparse grids in Section 2.2. We discuss how to solve the corresponding systems of linear equations with multigrid methods in Section 2.3. In Section 2.4, we give a detailed problem formulation where we show that common multigrid methods lead to computationally inefficient coarse grid equations due to the hierarchical basis and the sparse grid structure.

2.1. Sparse grid spaces.

Let $\phi : \mathbb{R} \to \mathbb{R}$, with $\phi(x) = \max\{1 - |x|, 0\}$, be the standard hat function. We derive the one-dimensional hierarchical basis function centered at grid point $x_{l,i} = i \cdot 3^{-l} \in [0, 1] \subset \mathbb{R}$ as $\phi_{l,i}(x) = \phi(3^l x - i)$, where $l \in \mathbb{N}$ is the level and $i \in \mathbb{N}$ the index. We use coarsening by a factor of three purely for technical reasons [21]. All of the following is, however, applicable to other hierarchical

schemes, including classical coarsening by a factor of two. With a tensor product, we derive the d-dimensional hierarchical basis function

$$\phi_{\boldsymbol{l},\boldsymbol{i}}(\boldsymbol{x}) = \phi_{\boldsymbol{l},\boldsymbol{i}}(x_1,\ldots,x_d) = \prod_{j=1}^{d} \phi_{l_j,i_j}(x_j) \tag{2.1}$$

centered at grid point $\boldsymbol{x}_{\boldsymbol{l},\boldsymbol{i}} = [x_{l_1,i_1},\ldots,x_{l_d,i_d}] \in [0,1]^d \subset \mathbb{R}^d$, with level $\boldsymbol{l} = [l_1,\ldots,l_d] \in \mathbb{N}^d$ and index $\boldsymbol{i} = [i,\ldots,i_d] \in \mathbb{N}^d$. For the sake of exposition, we often simply write $\phi_{\boldsymbol{l},\boldsymbol{i}} = \prod_j \phi_{l_j,i_j}$.

With a given $\boldsymbol{k} \in \mathbb{N}^d$ and index set

$$\tilde{\mathcal{I}}_{\boldsymbol{k}} = \left\{ \boldsymbol{i} \in \mathbb{N}^d \,|\, 1 \le i_j \le 3^{k_j},\, 1 \le j \le d \right\} , \tag{2.2}$$

we define the basis $\tilde{\Phi}_{\boldsymbol{k}} = \{\phi_{\boldsymbol{k},\boldsymbol{i}} \,|\, \boldsymbol{i} \in \tilde{\mathcal{I}}_{\boldsymbol{k}}\}$, which is a basis for the space of piecewise d-linear functions $\tilde{\mathcal{H}}_{\boldsymbol{k}}$. A function $u \in \tilde{\mathcal{H}}_{\boldsymbol{k}}$ can be represented as the unique linear combination

$$u = \sum_{\boldsymbol{i} \in \tilde{\mathcal{I}}_{\boldsymbol{k}}} \tilde{\alpha}_{\boldsymbol{k},\boldsymbol{i}} \phi_{\boldsymbol{k},\boldsymbol{i}} , \tag{2.3}$$

with coefficients $\tilde{\alpha}_{\boldsymbol{k},\boldsymbol{i}} \in \mathbb{R}$ for $\boldsymbol{i} \in \tilde{\mathcal{I}}_{\boldsymbol{k}}$. All basis functions in the sum (2.3) are on level $\boldsymbol{k}$, thus (2.3) is a nodal point representation, where $u(\boldsymbol{x}_{\boldsymbol{k},\boldsymbol{i}}) = \tilde{\alpha}_{\boldsymbol{k},\boldsymbol{i}}$ holds. Note that we have $\phi_{\boldsymbol{l},\boldsymbol{i}} \in \tilde{\mathcal{H}}_{\boldsymbol{k}}$ with $\boldsymbol{l} \le \boldsymbol{k}$ and $\boldsymbol{i} \in \tilde{\mathcal{I}}_{\boldsymbol{l}}$, i.e.,

$$\phi_{\boldsymbol{l},\boldsymbol{i}} = \sum_{\boldsymbol{t} \in \tilde{\mathcal{I}}_{\boldsymbol{k}}} \phi_{\boldsymbol{l},\boldsymbol{i}}(\boldsymbol{x}_{\boldsymbol{k},\boldsymbol{t}}) \phi_{\boldsymbol{k},\boldsymbol{t}} , \tag{2.4}$$

with the componentwise relational operator

$$\boldsymbol{l} \le \boldsymbol{k} : \Longleftrightarrow\ \forall j \in \{1,\ldots,d\} : l_j \le k_j .$$

We also define the index set $\hat{\mathcal{I}}_{\boldsymbol{k}} = \{\boldsymbol{i} \in \tilde{\mathcal{I}}_{\boldsymbol{k}} \,|\, i_j \notin 3\mathbb{N}, 1 \le j \le d\}$ and the hierarchical increments $\mathcal{W}_{\boldsymbol{k}}$ spanned by $\hat{\Phi}_{\boldsymbol{k}} = \{\phi_{\boldsymbol{k},\boldsymbol{i}} \,|\, \boldsymbol{i} \in \hat{\mathcal{I}}_{\boldsymbol{k}}\}$. The space $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ can be represented as direct sum

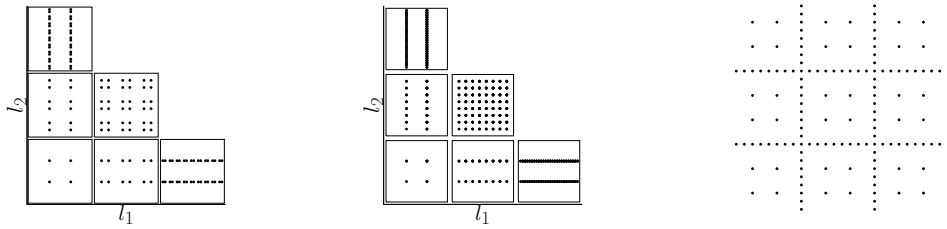
$$\tilde{\mathcal{H}}_{\boldsymbol{k}} = \bigoplus_{\boldsymbol{l} \le \boldsymbol{k}} \mathcal{W}_{\boldsymbol{l}}$$

of hierarchical increments; see Figure 2.1. Similarly to (2.3), we can represent a function $u \in \tilde{\mathcal{H}}_{\boldsymbol{k}}$ as sum

$$u = \sum_{\boldsymbol{l} \le \boldsymbol{k}} \sum_{\boldsymbol{i} \in \hat{\mathcal{I}}_{\boldsymbol{l}}} \alpha_{\boldsymbol{l},\boldsymbol{i}} \phi_{\boldsymbol{l},\boldsymbol{i}} , \tag{2.5}$$

where we now have a hierarchical representation with the so-called hierarchical coefficients. We call $\bigcup_{\boldsymbol{l} \le \boldsymbol{k}} \hat{\Phi}_{\boldsymbol{k}}$ the hierarchical basis of the space $\tilde{\mathcal{H}}_{\boldsymbol{k}}$. Algorithms to transform between the nodal point (2.3) and the hierarchical (2.5) basis representation are available; see, e.g., [4]. The space $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ as well as the hierarchical increment $\mathcal{W}_{\boldsymbol{k}}$ have full tensor product structure, i.e.,

$$\tilde{\mathcal{H}}_{\boldsymbol{k}} = \bigotimes_{j=1}^{d} \tilde{\mathcal{H}}_{k_j} , \qquad \mathcal{W}_{\boldsymbol{k}} = \bigotimes_{j=1}^{d} \mathcal{W}_{k_j} ,$$



(a) hierarchical increments (b) hierarchical subspaces (c) sparse grid

FIG. 2.1. *In (a), the grid points corresponding to the hierarchical increments $\mathcal{W}_{\boldsymbol{l}}$ of a sparse grid of level three arranged in the hierarchical scheme are plotted and, in (b), the grid points of the hierarchical subspaces $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ are shown. The corresponding sparse grid is given in (c).*

and we thus call them full grid spaces.

We define the sparse grid space $\mathcal{V}_\ell^{(1)}$ of dimension $d \in \mathbb{N}$ and level $\ell \in \mathbb{N}$ as

$$\mathcal{V}_\ell^{(1)} = \bigoplus_{\boldsymbol{l} \in \mathcal{L}} \mathcal{W}_{\boldsymbol{l}} \tag{2.6}$$

with the selection

$$\mathcal{L} = \{\boldsymbol{l} \in \mathbb{N}^d \,|\, \mathbf{0} < \boldsymbol{l} \text{ and } |\boldsymbol{l}|_1 \leq \ell + d - 1\}\,, \tag{2.7}$$

where $|\boldsymbol{l}|_1 = \sum_i l_i$; see Figure 2.1. The space $\mathcal{V}_\ell^{(1)}$ has the basis $\Phi_{\mathcal{L}} = \bigcup_{\boldsymbol{l} \in \mathcal{L}} \hat{\Phi}_{\boldsymbol{l}}$ with $\mathcal{O}(3^\ell \ell^{d-1})$ basis functions. We denote the number of basis functions with $N \in \mathbb{N}$. Roughly speaking, the selection $\mathcal{L}$ of hierarchical increments leads to an optimal approximation space with respect to the interpolation error in the L^2 norm for functions with bounded mixed derivatives up to order two. We refer to [4] for detailed proofs, properties of sparse grid spaces, and further references.

2.2. Sparse grid discretization. With correspondence to a linear PDE[1], let $\Omega = (0,1)^d$ be a domain, $\mathcal{V}$ a Hilbert space, $\mathcal{V}_\ell^{(1)}$ a sparse grid space with $\mathcal{V}_\ell^{(1)} \subset \mathcal{V}$, $a : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ a bilinear form, and $b : \mathcal{V} \to \mathbb{R}$ a linear form. For the sake of exposition, and without loss of generality, we set $b = 0$. Note, however, that the following multigrid method is also applicable if $b \neq 0$. We obtain with Ritz-Galerkin and the sparse grid space $\mathcal{V}_\ell^{(1)}$ the system of linear equations

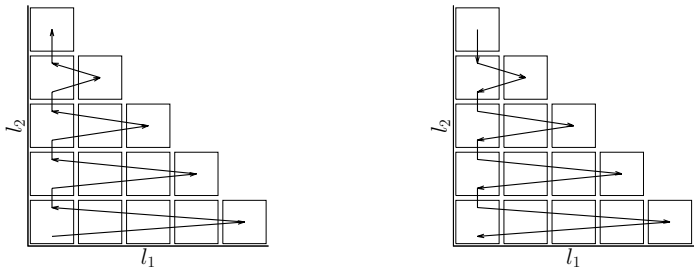
$$a(u, \psi_{\boldsymbol{k},\boldsymbol{t}}) = \sum_{\boldsymbol{l} \in \mathcal{L}} \sum_{\boldsymbol{i} \in \hat{\mathcal{I}}_{\boldsymbol{l}}} \alpha_{\boldsymbol{l},\boldsymbol{i}} a(\phi_{\boldsymbol{l},\boldsymbol{i}}, \psi_{\boldsymbol{k},\boldsymbol{t}}) = 0 \qquad \forall \psi_{\boldsymbol{k},\boldsymbol{t}} \in \Psi_{\mathcal{L}}\,, \tag{2.8}$$

with the sparse grid function $u \in \mathcal{V}_\ell^{(1)}$ and the test functions in $\Psi_{\mathcal{L}} = \Phi_{\mathcal{L}}$. In the following, we only consider bilinear forms a with a tensor product structure in the sense that

$$a(\phi_{\boldsymbol{l},\boldsymbol{i}}, \psi_{\boldsymbol{k},\boldsymbol{t}}) = \prod_{j=1}^{d} a_j(\phi_{l_j,i_j}, \psi_{k_j,t_j}) \tag{2.9}$$

holds, where $a_1, \ldots, a_d$ are the corresponding one-dimensional bilinear forms; see, e.g., [4]. The tensor product structure (2.9) is a key prerequisite for most sparse grid

[1] Examples will follow in Section 5.



(a) forward direction (b) backward direction

FIG. 2.2. *The figure shows the hierarchical scheme of a two-dimensional sparse grid of level five and the path of the Q-cycle through the subspaces. Note that the sparse grid points corresponding to the hierarchical subspaces are not plotted here, cf. Figure 2.1b.*

algorithms because it allows to apply one-dimensional algorithms in each dimension [21, 27]. We also exploit the tensor structure to obtain a computationally efficient method. We can also consider sums of such bilinear forms, for example, the summands of the Laplace operator. We also note that the bilinear forms do not need to be symmetric. This is in contrast to other sparse grid algorithms; see [27, Section 2.6]. We also emphasize that the method introduced in [26] does not exploit the product structure (2.9) if present. It thus can naturally cope with PDEs with variable coefficients that do not have such tensor product structure; however, only in up to two dimensions.

2.3. Multigrid and sparse grids. Multigrid methods have been shown to be optimal solvers for many systems of linear equations stemming from the discretization of a wide class of PDEs [31]. Optimal means that one iteration has runtime costs that are bounded linearly in the number of degrees of freedom and that the residual is reduced by a (convergence) factor that is bounded independently of the mesh width. We therefore solve our sparse grid system (2.8) with a multigrid method. We specify suitable multigrid ingredients following [26].

Let us first consider the grid hierarchy. A sparse grid does not have a natural grid hierarchy where coarsening takes place in each direction simultaneously. We therefore define the grid hierarchy of a sparse grid based on the hierarchical subspaces $\tilde{\mathcal{H}}_{\boldsymbol{k}} \subset \mathcal{V}_\ell^{(1)}, \boldsymbol{k} \in \mathcal{L}$, of the sparse grid space $\mathcal{V}_\ell^{(1)}$. There, the grids are coarsened in each direction separately, i.e., semi-coarsening; see Figure 2.1. We therefore employ the so-called Q-cycle, which traverses all hierarchical subspaces in $\{\tilde{\mathcal{H}}_{\boldsymbol{k}} \,|\, \boldsymbol{k} \in \mathcal{L}\}$ of a sparse grid space by recursively nesting standard V-cycles; see [26] and Figure 2.2. An algorithmic formulation and an extension to more than two dimensions can be found in [21].

We select standard grid transfer operators. The prolongation operator $P : \tilde{\mathcal{H}}_{\boldsymbol{k}'} \to \tilde{\mathcal{H}}_{\boldsymbol{k}}$, with $\boldsymbol{k} = [k_1, \ldots, k_q, \ldots, k_d] \in \mathbb{N}^d$ and $\boldsymbol{k}' = [k_1, \ldots, k_q - 1, \ldots, k_d] \in \mathbb{N}^d$, corresponds to linear interpolation. The restriction operator $R : \tilde{\mathcal{H}}_{\boldsymbol{k}} \to \tilde{\mathcal{H}}_{\boldsymbol{k}'}$ corresponds to the trivial restriction. In the context of sparse grids and the hierarchical basis, these prolongation and restriction operators correspond to hierarchisation and dehierarchisation, respectively [4, 21, 26].

We then define the systems of linear equations on the coarse grids as proposed in [26]. For each hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}}$, i.e., for each coarse grid, the sparse grid

space $\mathcal{V}_\ell^{(1)}$ is split into

$$\mathcal{V}_\ell^{(1)} = \tilde{\mathcal{H}}_{\boldsymbol{k}} \oplus \hat{\mathcal{H}}_{\boldsymbol{k}}\,, \tag{2.10}$$

where $\hat{\mathcal{H}}_{\boldsymbol{k}}$ is the direct sum of all hierarchical increments of the sparse grid space $\mathcal{V}_\ell^{(1)}$ that are not contained in $\tilde{\mathcal{H}}_{\boldsymbol{k}}$; see (2.6). We then obtain the coarse grid equations corresponding to $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ as

$$a(\tilde{u}_{\boldsymbol{k}}, \psi_{\boldsymbol{k},\boldsymbol{t}}) = -a(\hat{u}_{\boldsymbol{k}}, \psi_{\boldsymbol{k},\boldsymbol{t}})\,, \qquad \forall \psi_{\boldsymbol{k},\boldsymbol{t}} \in \Psi_{\boldsymbol{k}}\,, \tag{2.11}$$

with the decomposition $u = \tilde{u}_{\boldsymbol{k}} + \hat{u}_{\boldsymbol{k}}$ and the components $\tilde{u}_{\boldsymbol{k}} \in \tilde{\mathcal{H}}_{\boldsymbol{k}}$ and $\hat{u}_{\boldsymbol{k}} \in \hat{\mathcal{H}}_{\boldsymbol{k}}$. We set $\Psi_{\boldsymbol{k}} = \tilde{\Phi}_{\boldsymbol{k}}$.

2.4. Problem formulation. The three multigrid ingredients introduced in the previous section define a multigrid method that is applicable to systems of linear equations stemming from sparse grid discretizations. On each coarse grid corresponding to a hierarchical subspace, a relaxation sweep has to be performed for the corresponding system (2.11). Since the function $\tilde{u}_{\boldsymbol{k}}$ is an element of the full grid space $\tilde{\mathcal{H}}_{\boldsymbol{k}}$, it can be represented in a nodal point basis (2.3). Thus, standard relaxation schemes are applicable if the right-hand side $a(\hat{u}_{\boldsymbol{k}}, \psi_{\boldsymbol{k},\boldsymbol{t}})$ is provided; however, the right-hand side is still represented in the hierarchical basis, i.e.,

$$a(\hat{u}_{\boldsymbol{k}}, \psi_{\boldsymbol{k},\boldsymbol{t}}) = \sum_{\substack{\boldsymbol{l} \in \mathcal{L} \\ \boldsymbol{l} \not\leq \boldsymbol{k}}} \sum_{\boldsymbol{i} \in \hat{\mathcal{I}}_{\boldsymbol{l}}} \alpha_{\boldsymbol{l},\boldsymbol{i}} a(\phi_{\boldsymbol{l},\boldsymbol{i}}, \psi_{\boldsymbol{k},\boldsymbol{t}})\,. \tag{2.12}$$

If we also had a nodal point basis representation for the right-hand side, the support of the test function $\psi_{\boldsymbol{k},\boldsymbol{t}}$ would intersect with the support of a few basis functions in its neighborhood only. Thus, it would be cheap to compute with stencils; however, since we have a hierarchical basis, the test function $\psi_{\boldsymbol{k},\boldsymbol{t}}$ reaches to many basis functions centered all over the spatial domain Ω. A straightforward implementation to evaluate the sum (2.12) would lead to a multigrid algorithm with runtime costs scaling quadratically with the number of sparse grid points N. We therefore require a different representation of (2.12) that allows us to derive algorithms to perform a multigrid cycle with costs bounded linearly in the number of grid points N.

3. Hierarchical ANOVA-like decomposition. We propose to represent the right-hand sides of the coarse grid equations as linear combinations of a nodal point basis with so-called auxiliary coefficients. The representation is based on a hierarchical dimensionwise decomposition of the sparse grid solution function, which is developed in Section 3.1. The decomposition is applied to the system of linear equations in Section 3.2 and the linear combination of the nodal point basis with the auxiliary coefficients is derived in Section 3.3.

3.1. Decomposition of the sparse grid space and sparse grid function. Let $\mathcal{L}$ contain the levels of the hierarchical increments of the d-dimensional sparse grid space $\mathcal{V}_\ell^{(1)}$ of level ℓ; see (2.7). We consider the hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}} \subset \mathcal{V}_\ell^{(1)}$ with $\boldsymbol{k} \in \mathcal{L}$ and define the index sets

$$\mathcal{L}_{\boldsymbol{k}}^{\mathcal{N}} = \{\boldsymbol{l} \in \mathcal{L} \,|\, \forall i \in \mathcal{N} : l_i > k_i \text{ and } \forall i \in \mathcal{D} \setminus \mathcal{N} : l_i \leq k_i\} \tag{3.1}$$

for all $\mathcal{N} \subseteq \mathcal{D}$, where $\mathcal{D} = \{1, \ldots, d\} \subset \mathbb{N}$. With the sets (3.1), we define the spaces

$$\tilde{\mathcal{H}}_{\boldsymbol{k}}^{\mathcal{N}} = \bigoplus_{\boldsymbol{l} \in \mathcal{L}_{\boldsymbol{k}}^{\mathcal{N}}} \mathcal{W}_{\boldsymbol{l}}\,. \tag{3.2}$$

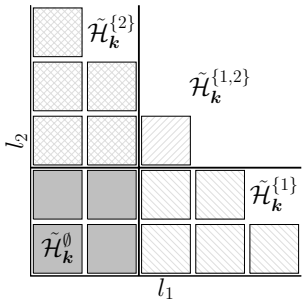


FIG. 3.1. *The figure shows the hierarchical scheme of a two-dimensional sparse grid space of level five. The black cross indicates which hierarchical increments belong to the spaces of the hierarchical ANOVA-like decomposition corresponding to the hierarchical subspace of level* $\boldsymbol{k} = [2, 2]$.

LEMMA 3.1. *The direct sum of the spaces* (3.2) *yields the sparse grid space* $\mathcal{V}_\ell^{(1)}$, *i.e.,*

$$\mathcal{V}_\ell^{(1)} = \bigoplus_{\mathcal{N} \subseteq \mathcal{D}} \tilde{\mathcal{H}}_{\boldsymbol{k}}^{\mathcal{N}}\,, \tag{3.3}$$

for a hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}} \subset \mathcal{V}_\ell^{(1)}$.

Proof. It follows from definition (3.1) that the sets $\mathcal{L}_{\boldsymbol{k}}^{\mathcal{N}}$ with $\mathcal{N} \subseteq \mathcal{D}$ are a partition of $\mathcal{L}$. Because of this and since $\mathcal{V}_\ell^{(1)}$ is a direct sum of hierarchical increments (2.6) with level in $\mathcal{L}$, the sum (3.3) of the spaces (3.2) has to be equal to the sparse grid space as in (3.3). ∎

Note that, for the empty set $\emptyset \subset \mathcal{D}$, we have $\tilde{\mathcal{H}}_{\boldsymbol{k}}^{\emptyset} = \tilde{\mathcal{H}}_{\boldsymbol{k}}$. Figure 3.1 visualizes the spaces (3.1) for the two-dimensional sparse grid space of level five. The decomposition (3.3) allows us to represent a sparse grid function $u \in \mathcal{V}_\ell^{(1)}$ as

$$u = \sum_{\mathcal{N} \subseteq \mathcal{D}} u_{\boldsymbol{k}}^{\mathcal{N}}\,, \tag{3.4}$$

with the components $u_{\boldsymbol{k}}^{\mathcal{N}} \in \tilde{\mathcal{H}}_{\boldsymbol{k}}^{\mathcal{N}}$ given as

$$u_{\boldsymbol{k}}^{\mathcal{N}} = \sum_{\boldsymbol{l} \in \mathcal{L}_{\boldsymbol{k}}^{\mathcal{N}}} \sum_{\boldsymbol{i} \in \hat{\mathcal{I}}_{\boldsymbol{l}}} \alpha_{\boldsymbol{l},\boldsymbol{i}} \phi_{\boldsymbol{l},\boldsymbol{i}}\,. \tag{3.5}$$

Let us compare our decomposition (3.3) with the decomposition (2.10) used for the multigrid methods in, e.g., [3, 22, 26]. The first summand in (3.3) is equal to $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ and the sum of all other terms is equal to $\hat{\mathcal{H}}_{\boldsymbol{k}}$ in the decomposition (2.10). Thus, our decomposition (3.3) can be seen as a more fine-grained decomposition than (2.10). Furthermore, the corresponding sparse grid function representation (3.4) is similar to high-dimensional model representations (HDMR) or ANOVA decompositions [5, 20, 29, 32]. There is a close connection between ANOVA and sparse grids that has been thoroughly studied in, e.g., [7, 9, 20]. In contrast to these previous works, however, we have a hierarchical decomposition that depends on the hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}}$. This dependence lets us control the representation of the component functions (3.5).

Consider the component $u_{\boldsymbol{k}}^{\mathcal{N}}$ corresponding to the hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ and the subset $\mathcal{N} \subseteq \mathcal{D}$. Even though $u_{\boldsymbol{k}}^{\mathcal{N}}$ is represented in the hierarchical basis in (3.5),

only basis functions of level $l_j > k_j$ for $j \in \mathcal{N}$ are taken into account. All basis functions in directions $j \in \mathcal{D} \setminus \mathcal{N}$ are of level $l_j \leq k_j$. Thus, we can represent $u_{\boldsymbol{k}}^{\mathcal{N}}$ with basis functions of level k_j in directions $j \in \mathcal{D} \setminus \mathcal{N}$. First, recall that $\tilde{\mathcal{I}}_{\boldsymbol{k}}$ contains all indices of the basis functions spanning $\tilde{\mathcal{H}}_{\boldsymbol{k}}$. We then define, for a grid point $\boldsymbol{x}_{\boldsymbol{k},\boldsymbol{t}}$, the index set

$$\tilde{\mathcal{I}}_{\boldsymbol{k},\boldsymbol{t}}^{\mathcal{D}\setminus\mathcal{N}} = \left\{ \boldsymbol{i} \in \tilde{\mathcal{I}}_{\boldsymbol{k}} \,|\, \forall j \in \mathcal{N} : i_j = t_j \right\}, \tag{3.6}$$

which contains the indices of the basis functions spanning $\tilde{\mathcal{H}}_{\boldsymbol{k}}$, where all components in directions $j \in \mathcal{N}$ are fixed to the corresponding components in $\boldsymbol{t}$. We might think of this as slicing a d-dimensional full grid at grid point $\boldsymbol{x}_{\boldsymbol{k},\boldsymbol{t}}$ along the directions in $\mathcal{N}$. With this definition, we state the following lemma.

LEMMA 3.2. *A component $u_{\boldsymbol{k}}^{\mathcal{N}}$ as defined in* (3.5) *can be represented with basis functions of level k_j in directions $j \in \mathcal{D} \setminus \mathcal{N}$. We thus obtain*

$$u_{\boldsymbol{k}}^{\mathcal{N}} = \sum_{\boldsymbol{\xi} \in \tilde{\mathcal{I}}_{\boldsymbol{k},\boldsymbol{1}}^{\mathcal{D}\setminus\mathcal{N}}} \underbrace{\prod_{j \in \mathcal{D}\setminus\mathcal{N}} \phi_{k_j,\xi_j}}_{\substack{\text{nodal point basis} \\ \text{in directions in } \mathcal{D}\setminus\mathcal{N}}} \sum_{\boldsymbol{l} \in \mathcal{L}_{\boldsymbol{k}}^{\mathcal{N}}} \sum_{\boldsymbol{i} \in \hat{\mathcal{I}}_{\boldsymbol{l}}} \underbrace{\alpha_{\boldsymbol{l},\boldsymbol{i}} \prod_{j \in \mathcal{D}\setminus\mathcal{N}} \phi_{l_j,i_j}(x_{k_j,\xi_j})}_{\text{new coefficients}} \underbrace{\prod_{j \in \mathcal{N}} \phi_{l_j,i_j}}_{\substack{\text{hierarchical basis} \\ \text{in directions in } \mathcal{N}}} \tag{3.7}$$

with the grid points $\boldsymbol{x}_{\boldsymbol{k},\boldsymbol{\xi}} = [x_{k_1,\xi_1}, \ldots, x_{k_d,\xi_d}] \in \Omega$ for $\boldsymbol{\xi} \in \tilde{\mathcal{I}}_{\boldsymbol{k},\boldsymbol{1}}^{\mathcal{D}\setminus\mathcal{N}}$.

Proof. We first note that $\boldsymbol{t}$ of (3.6) is set to $\boldsymbol{1}$ in (3.7) without any loss of generality because equation (3.7) is independent of ξ_j with $j \in \mathcal{N}$. The transformation from the definition of $u_{\boldsymbol{k}}^{\mathcal{N}}$ in (3.5) into

$$u_{\boldsymbol{k}}^{\mathcal{N}} = \sum_{\boldsymbol{l} \in \mathcal{L}_{\boldsymbol{k}}^{\mathcal{N}}} \sum_{\boldsymbol{i} \in \hat{\mathcal{I}}_{\boldsymbol{l}}} \sum_{\boldsymbol{\xi} \in \tilde{\mathcal{I}}_{\boldsymbol{k},\boldsymbol{1}}^{\mathcal{D}\setminus\mathcal{N}}} \underbrace{\alpha_{\boldsymbol{l},\boldsymbol{i}} \prod_{j \in \mathcal{D}\setminus\mathcal{N}} \phi_{l_j,i_j}(x_{k_j,\xi_j})}_{\text{new coefficients}} \underbrace{\prod_{j \in \mathcal{N}} \phi_{l_j,i_j}}_{\substack{\text{hierarchical basis} \\ \text{in directions in } \mathcal{N}}} \underbrace{\prod_{j \in \mathcal{D}\setminus\mathcal{N}} \phi_{k_j,\xi_j}}_{\substack{\text{nodal point basis} \\ \text{in directions in } \mathcal{D}\setminus\mathcal{N}}}$$

follows with the tensor product structure (2.1) and property (2.4) of the basis functions. The reordering to (3.7) holds because the basis functions of level k_j for $j \in \mathcal{D}\setminus\mathcal{N}$ do not depend on the level $\boldsymbol{l} \in \mathcal{L}_{\boldsymbol{k}}^{\mathcal{N}}$ anymore. ∎

3.2. Decomposition of the bilinear form. We now consider the system of linear equations (2.11) corresponding to the hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ of the sparse grid space $\mathcal{V}_\ell^{(1)}$. We can evaluate $a(u, \psi_{\boldsymbol{k},\boldsymbol{t}})$ at $u \in \mathcal{V}_\ell^{(1)}$ and at a test function $\psi_{\boldsymbol{k},\boldsymbol{t}} \in \Psi_{\boldsymbol{k}}$ by evaluating the bilinear form at each component $u_{\boldsymbol{k}}^{\mathcal{N}}$ of u. We therefore have

$$a(u, \psi_{\boldsymbol{k},\boldsymbol{t}}) = \sum_{\mathcal{N} \subseteq \mathcal{D}} a(u_{\boldsymbol{k}}^{\mathcal{N}}, \psi_{\boldsymbol{k},\boldsymbol{t}}). \tag{3.8}$$

Similar to the representation (3.7) for a component function $u_{\boldsymbol{k}}^{\mathcal{N}}$, we find a representation of one component $a(u_{\boldsymbol{k}}^{\mathcal{N}}, \psi_{\boldsymbol{k},\boldsymbol{t}})$ of (3.8) with basis functions of level k_j in directions $j \in \mathcal{D} \setminus \mathcal{N}$.

LEMMA 3.3. *A summand $a(u_{\boldsymbol{k}}^{\mathcal{N}}, \psi_{\boldsymbol{k},\boldsymbol{t}})$ of* (3.8) *can be represented as*

$$\sum_{\boldsymbol{\xi} \in \tilde{\mathcal{I}}_{\boldsymbol{k},\boldsymbol{1}}^{\mathcal{D}\setminus\mathcal{N}}} \prod_{j \in \mathcal{D}\setminus\mathcal{N}} a_j(\phi_{k_j,\xi_j}, \psi_{k_j,t_j}) \sum_{\boldsymbol{l} \in \mathcal{L}_{\boldsymbol{k}}^{\mathcal{N}}} \sum_{\boldsymbol{i} \in \hat{\mathcal{I}}_{\boldsymbol{l}}} \alpha_{\boldsymbol{l},\boldsymbol{i}} \prod_{j \in \mathcal{D}\setminus\mathcal{N}} \phi_{l_j,i_j}(x_{k_j,\xi_j}) \prod_{j \in \mathcal{N}} a_j(\phi_{l_j,i_j}, \psi_{k_j,t_j}), \tag{3.9}$$

where $a(\phi_{\boldsymbol{l},\boldsymbol{i}}, \psi_{\boldsymbol{k},\boldsymbol{t}}) = \prod_j a_j(\phi_{l_j,i_j}, \psi_{k_j,t_j})$.

Proof. The transformation follows from Lemma 3.2 and the tensor product structure of the bilinear form (2.9). ∎

The representation (3.9) allows us to evaluate $a(u_{\boldsymbol{k}}^{\mathcal{N}}, \psi_{\boldsymbol{k},\boldsymbol{t}})$ by invoking the hierarchical basis only in directions $j \in \mathcal{N}$. In all other directions $j \in \mathcal{D} \setminus \mathcal{N}$, basis functions of level k_j are sufficient. In particular, in all directions $j \in \mathcal{D} \setminus \mathcal{N}$, the basis functions and the test functions are of the same level k_j. We can thus think of evaluating the bilinear form on a low-dimensional full grid of dimension $d - |\mathcal{N}|$ in the directions $j \in \mathcal{D} \setminus \mathcal{N}$.

3.3. Auxiliary coefficients. We now derive auxiliary coefficients that allow us to represent a component $a(u_{\boldsymbol{k}}^{\mathcal{N}}, \psi_{\boldsymbol{k},\boldsymbol{t}})$ of the bilinear form (3.8) in the nodal point basis of level $\boldsymbol{k}$ in directions $\mathcal{D} \setminus \mathcal{N}$. For that, we define the auxiliary coefficients

$$r_{\boldsymbol{k},\boldsymbol{\xi}}^{\mathcal{N}} = \sum_{\boldsymbol{l} \in \mathcal{L}_{\boldsymbol{k}}^{\mathcal{N}}} \sum_{\boldsymbol{i} \in \hat{\mathcal{I}}_{\boldsymbol{l}}} \alpha_{\boldsymbol{l},\boldsymbol{i}} \prod_{j \in \mathcal{D} \setminus \mathcal{N}} \phi_{l_j,i_j}(x_{k_j,\xi_j}) \prod_{j \in \mathcal{N}} a_j(\phi_{l_j,i_j}, \psi_{k_j,t_j}), \tag{3.10}$$

for all $\mathcal{N} \subseteq \mathcal{D}$ and $\boldsymbol{\xi} \in \tilde{\mathcal{I}}_{\boldsymbol{k},\boldsymbol{t}}^{\mathcal{D} \setminus \mathcal{N}}$ corresponding to the hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}}$. We note that $r_{\boldsymbol{k},\boldsymbol{\xi}}^{\emptyset} \in \mathbb{R}$ is the coefficient corresponding to the basis function $\phi_{\boldsymbol{k},\boldsymbol{\xi}}$ of the nodal point representation of $u_{\boldsymbol{k}}^{\emptyset} \in \tilde{\mathcal{H}}_{\boldsymbol{k}}$, i.e., $r_{\boldsymbol{k},\boldsymbol{\xi}}^{\emptyset} = u_{\boldsymbol{k}}^{\emptyset}(\boldsymbol{x}_{\boldsymbol{k},\boldsymbol{\xi}})$. With the auxiliary coefficients (3.10) and Lemma 3.3, we obtain

$$a(u_{\boldsymbol{k}}^{\mathcal{N}}, \psi_{\boldsymbol{k},\boldsymbol{t}}) = \sum_{\boldsymbol{\xi} \in \tilde{\mathcal{I}}_{\boldsymbol{k},\boldsymbol{t}}^{\mathcal{D} \setminus \mathcal{N}}} r_{\boldsymbol{k},\boldsymbol{\xi}}^{\mathcal{N}} \prod_{j \in \mathcal{D} \setminus \mathcal{N}} a_j(\phi_{k_j,\xi_j}, \psi_{k_j,t_j}).$$

This means that the component $a(u_{\boldsymbol{k}}^{\mathcal{N}}, \psi_{\boldsymbol{k},\boldsymbol{t}})$ is represented in a nodal point basis of dimension $d - |\mathcal{N}|$ and of level $\boldsymbol{k}$ with the coefficients (3.10). Thus, if the auxiliary coefficients for each component of the decomposition (3.8) are available, the bilinear form $a(u, \psi_{\boldsymbol{k},\boldsymbol{t}})$ can be evaluated for each test function $\psi_{\boldsymbol{k},\boldsymbol{t}} \in \Psi_{\boldsymbol{k}}$ with

$$a(u, \psi_{\boldsymbol{k},\boldsymbol{t}}) = \sum_{\mathcal{N} \subseteq \mathcal{D}} \sum_{\boldsymbol{\xi} \in \tilde{\mathcal{I}}_{\boldsymbol{k},\boldsymbol{t}}^{\mathcal{D} \setminus \mathcal{N}}} r_{\boldsymbol{k},\boldsymbol{\xi}}^{\mathcal{N}} \prod_{j \in \mathcal{D} \setminus \mathcal{N}} a_j(\phi_{k_j,\xi_j}, \psi_{k_j,t_j}). \tag{3.11}$$

This representation involves only basis functions of level $\boldsymbol{k}$ centered at the full grids corresponding to the index sets $\tilde{\mathcal{I}}_{\boldsymbol{k},\boldsymbol{t}}^{\mathcal{D} \setminus \mathcal{N}}$ with $\mathcal{N} \subseteq \mathcal{D}$. Hence, the sparse grid structure and the hierarchical basis are hidden in the auxiliary coefficients.

4. Hierarchical multigrid method based on auxiliary coefficients. The auxiliary coefficient representation (3.11) of the coarse grid equation (2.11) depends on the hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}}$, i.e., on the current grid level. Thus, the auxiliary coefficients have to be updated after a grid transfer. We discuss in Section 4.1 and 4.2 that the auxiliary coefficients have a hierarchical structure that can be exploited to construct the coarse grid equations without recomputing the coefficients from scratch. We then show in Section 4.3 that this leads to a sparse grid multigrid solver with optimal costs, i.e., runtime and storage costs scaling linearly with the number of sparse grid points.

4.1. Hierarchical structure of auxiliary coefficients. We first show that the hierarchisation of the auxiliary coefficients removes the dependency on hierarchical coefficients corresponding to hierarchically lower basis functions, and then discuss

the relationship between the auxiliary coefficients corresponding to two different hierarchical subspaces, i.e., to two different grids of the grid hierarchy.

Consider the auxiliary coefficient $r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}}$ for $\mathcal{N} \subseteq \mathcal{D}$ corresponding to the hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ of the sparse grid space $\mathcal{V}^{(1)}_{\ell}$. The coefficient $r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}}$ is already hierarchised in directions $j \in \mathcal{N}$ where the hierarchical basis is used; however, in all other directions $j \in \mathcal{D} \setminus \mathcal{N}$, the hierarchical basis is evaluated at the grid points such that it becomes dehierarchised; cf. equation (2.4). We hierarchise the auxiliary coefficient $r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}}$ in a direction $q \in \mathcal{D} \setminus \mathcal{N}$ with the standard stencils; see [4, 21] for details. The hierarchised auxiliary coefficients evaluate only hierarchical basis functions of level k_q in direction q. Thus, we obtain the hierarchised auxiliary coefficient

$$\hat{r}^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}} = \sum_{\substack{\boldsymbol{l} \in \mathcal{L}^{\mathcal{N}}_{\boldsymbol{k}} \\ l_q = k_q}} \sum_{\boldsymbol{i} \in \hat{\mathcal{I}}_{\boldsymbol{l}}} \alpha_{\boldsymbol{l},\boldsymbol{i}} \prod_{j \in \mathcal{D} \setminus \mathcal{N}} \phi_{l_j,i_j}(x_{k_j,\xi_j}) \prod_{j \in \mathcal{N}} a_j(\phi_{l_j,i_j}, \psi_{k_j,t_j}) \,, \tag{4.1}$$

where the sum is restricted to levels with $l_q = k_q$. Note that the indices $t_j, j \in \mathcal{N}$, in (4.1) are defined through the index $\boldsymbol{\xi}$; see equation (3.10). The hierarchised coefficient (4.1) is independent of the hierarchical coefficients with $l_q < k_q$. For the sake of exposition, we do not add another index to $\hat{r}^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}}$ to indicate the direction in which it is hierarchised, but instead always explicitly state the direction in the text.

Let now $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ be the hierarchical subspace with level $\boldsymbol{k}$ and let $\tilde{\mathcal{H}}_{\boldsymbol{k}'}$ be the subspace with level $k'_q = k_q - 1$ in direction $q \in \mathcal{D}$. The following theorem shows that the hierarchical structure can be exploited to represent an auxiliary coefficient corresponding to $\tilde{\mathcal{H}}_{\boldsymbol{k}'}$ as linear combination of dehierarchised and hierarchised auxiliary coefficients corresponding to $\tilde{\mathcal{H}}_{\boldsymbol{k}}$. Thus, an auxiliary coefficient at a coarse grid can be computed from the auxiliary coefficients at the fine grid. Corollary 4.2 shows that the same holds for the inverse direction, i.e., from the coarse grid to the fine grid. The following theorem and its corollary are the key ingredients to derive an efficient updating algorithm for the auxiliary coefficients in the following section.

THEOREM 4.1. *Let $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ be a hierarchical subspace of level $\boldsymbol{k} \in \mathcal{L}$ of the d-dimensional sparse grid space $\mathcal{V}^{(1)}_{\ell}$, and let $\tilde{\mathcal{H}}_{\boldsymbol{k}'}$ be the space with $\boldsymbol{k}' = [k_1, \ldots, k_q - 1, \ldots, k_d] \in \mathcal{L}$ and $q \in \mathcal{D}$. Let $\psi_{\boldsymbol{k},\boldsymbol{t}} \in \Psi_{\boldsymbol{k}}$ and $\psi_{\boldsymbol{k}',\boldsymbol{t}'} \in \Psi_{\boldsymbol{k}'}$ be test functions of level $\boldsymbol{k}$ and $\boldsymbol{k}'$, respectively, with $\boldsymbol{t}' = [t_1, \ldots, t_q/3, \ldots, t_d] \in \mathbb{N}^d$. With the index sets $\tilde{\mathcal{I}}^{\mathcal{D}\setminus\mathcal{N}}_{\boldsymbol{k},\boldsymbol{t}}$ and $\tilde{\mathcal{I}}^{\mathcal{D}\setminus\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{t}'}$, see (3.6), and an index $\boldsymbol{\xi} \in \tilde{\mathcal{I}}^{\mathcal{D}\setminus\mathcal{N}}_{\boldsymbol{k},\boldsymbol{t}}$ such that $\boldsymbol{\xi}' = [\xi_1, \ldots, \xi_q/3, \ldots, \xi_d] \in \tilde{\mathcal{I}}^{\mathcal{D}\setminus\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{t}'}$, we consider the auxiliary coefficients $r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}}$ and $r^{\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{\xi}'}$ with $\mathcal{N} \subseteq \mathcal{D}$ corresponding to the subspaces $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ and $\tilde{\mathcal{H}}_{\boldsymbol{k}'}$, respectively. Then $r^{\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{\xi}'} = r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}}$ holds if $q \notin \mathcal{N}$, and*

$$\begin{aligned} r^{\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{\xi}'} = {} & a_q(\phi_{k_q,\xi_q-2}, \psi_{k'_q,\xi'_q}) \hat{r}^{\mathcal{N}\setminus\{q\}}_{\boldsymbol{k},\boldsymbol{\xi}_{-2}} + a_q(\phi_{k_q,\xi_q+2}, \psi_{k'_q,\xi'_q}) \hat{r}^{\mathcal{N}\setminus\{q\}}_{\boldsymbol{k},\boldsymbol{\xi}_{+2}} \\ & + a_q(\phi_{k_q,\xi_q-1}, \psi_{k'_q,\xi'_q}) \hat{r}^{\mathcal{N}\setminus\{q\}}_{\boldsymbol{k},\boldsymbol{\xi}_{-1}} + a_q(\phi_{k_q,\xi_q+1}, \psi_{k'_q,\xi'_q}) \hat{r}^{\mathcal{N}\setminus\{q\}}_{\boldsymbol{k},\boldsymbol{\xi}_{+1}} \\ & + \frac{1}{3}\left(r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}_{-2}} + r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}_{+2}}\right) + \frac{2}{3}\left(r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}_{-1}} + r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}_{+1}}\right) + r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}} \end{aligned} \tag{4.2}$$

holds if $q \in \mathcal{N}$, where $\hat{r}^{\mathcal{N}\setminus\{q\}}_{\boldsymbol{k},\boldsymbol{\xi}_{\pm i}}$ are auxiliary coefficients hierarchised in direction q as defined in (4.1) *and $\boldsymbol{\xi}_{\pm i} = [\xi_1, \ldots, \xi_q \pm i, \ldots, \xi_d]$ for $i \in \{1, 2\}$.*

Proof. First consider the case with $q \notin \mathcal{N}$, where we want to show $r^{\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{\xi}'} = r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}}$. We have $k'_q = k_q - 1$ and $\xi'_q = \xi_q/3$ and thus $\boldsymbol{x}_{\boldsymbol{k}',\boldsymbol{\xi}'} = \boldsymbol{x}_{\boldsymbol{k},\boldsymbol{\xi}}$. We then know from definition (3.10) that the only difference between $r^{\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{\xi}'}$ and $r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}}$ is that $r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}}$ includes

hierarchical basis functions of level $l_q \leq k_q$ and $r^{\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{\xi}'}$ basis functions with level $l_q \leq k_q - 1$; however, these hierarchical basis functions are evaluated at the grid point $x_{k_q,\xi_q} = \xi_q \cdot 3^{-k_q} = \frac{\xi_q}{3} \cdot 3^{-k_q+1} = x_{k_q-1,\xi_q/3}$ of level $k_q - 1$, and thus all basis functions of level k_q evaluate to zero; see Section 2.1. Hence, the basis functions of level k_q do not change the value of the sum and we thus obtain $r^{\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{\xi}'} = r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}}$.

Consider now the case with $q \in \mathcal{N}$. Because $r^{\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{\xi}'}$ includes hierarchical basis functions of level $l_q > k_q - 1$, we can split it into a sum that includes the basis functions of level $l_q = k_q$ and a sum with the basis functions of $l_q > k_q$, i.e.,

$$r^{\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{\xi}'} = \sum_{\substack{\boldsymbol{l}\in\mathcal{L}^{\mathcal{N}}_{\boldsymbol{k}'} \\ l_q = k_q}} \sum_{\boldsymbol{i}\in\hat{\mathcal{I}}_{\boldsymbol{l}}} \alpha_{\boldsymbol{l},\boldsymbol{i}} \prod_{j\in\mathcal{D}\setminus\mathcal{N}} \phi_{l_j,i_j}(x_{k_j,\xi_j}) \prod_{j\in\mathcal{N}\setminus\{q\}} a_j(\phi_{l_j,i_j},\psi_{k_j,t_j}) a_q(\phi_{l_q,i_q},\psi_{k_q-1,\xi_q/3})$$
$$+ \sum_{\substack{\boldsymbol{l}\in\mathcal{L}^{\mathcal{N}}_{\boldsymbol{k}'} \\ l_q > k_q}} \sum_{\boldsymbol{i}\in\hat{\mathcal{I}}_{\boldsymbol{l}}} \alpha_{\boldsymbol{l},\boldsymbol{i}} \prod_{j\in\mathcal{D}\setminus\mathcal{N}} \phi_{l_j,i_j}(x_{k_j,\xi_j}) \prod_{j\in\mathcal{N}\setminus\{q\}} a_j(\phi_{l_j,i_j},\psi_{k_j,t_j}) a_q(\phi_{l_q,i_q},\psi_{k_q-1,\xi_q/3}) \,.$$

The first sum with $l_q = k_q$ corresponds to the hierarchised auxiliary coefficient $\hat{r}^{\mathcal{N}\setminus\{q\}}_{\boldsymbol{k},\boldsymbol{\xi}}$, except that we have to account for the bilinear form $a_q(\phi_{l_q,i_q},\psi_{k_q-1,\xi_q/3})$. The second sum with $l_q > k_q$ corresponds to the auxiliary coefficients $r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}}$, but we have to change the test function in direction q from level k_q to $k_q - 1$. We therefore derive from (2.4) that

$$\psi_{k_q-1,\xi_q/3} = \frac{1}{3}\left(\psi_{k_q,\xi_q-2} + \psi_{k_q,\xi_q+2}\right) + \frac{2}{3}\left(\psi_{k_q,\xi_q-1} + \psi_{k_q,\xi_q+1}\right) + \psi_{k_q,\xi_q}$$

which leads to (4.2). ∎

COROLLARY 4.2. *Let the setting and the auxiliary coefficients $r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}}$ and $r^{\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{\xi}'}$ be given as in Theorem 4.1. We then have, for $q \notin \mathcal{N}$, that $r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}} = r^{\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{\xi}'}$ and, for $q \in \mathcal{N}$, that*

$$\begin{aligned} r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}} = r^{\mathcal{N}}_{\boldsymbol{k}',\boldsymbol{\xi}'} &- \frac{1}{3}\left(r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}_{-2}} + r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}_{+2}}\right) - \frac{2}{3}\left(r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}_{-1}} + r^{\mathcal{N}}_{\boldsymbol{k},\boldsymbol{\xi}_{+1}}\right) \\ &- a_q(\phi_{k_q,\xi_q-2},\psi_{k'_q,\xi'_q})\hat{r}^{\mathcal{N}\setminus\{q\}}_{\boldsymbol{k},\boldsymbol{\xi}_{-2}} - a_q(\phi_{k_q,\xi_q+2},\psi_{k'_q,\xi'_q})\hat{r}^{\mathcal{N}\setminus\{q\}}_{\boldsymbol{k},\boldsymbol{\xi}_{+2}} \\ &- a_q(\phi_{k_q,\xi_q-1},\psi_{k'_q,\xi'_q})\hat{r}^{\mathcal{N}\setminus\{q\}}_{\boldsymbol{k},\boldsymbol{\xi}_{-1}} - a_q(\phi_{k_q,\xi_q+1},\psi_{k'_q,\xi'_q})\hat{r}^{\mathcal{N}\setminus\{q\}}_{\boldsymbol{k},\boldsymbol{\xi}_{+1}} \,. \end{aligned}$$

Proof. This corollary follows immediately from Theorem 4.1 ∎

4.2. Updating algorithm. In this section, we give an algorithmic formulation to compute the auxiliary coefficients corresponding to the hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}'}$, with $\boldsymbol{k}' = [k_1,\ldots,k_q-1,\ldots,k_d] \in \mathcal{L}$ from the coefficients of space $\tilde{\mathcal{H}}_{\boldsymbol{k}}$. The following algorithm only considers coarsening the grid in direction q. The grid can be refined by performing the inverse operations as shown in Corollary 4.2.

We consider an in-place storage scheme for the auxiliary coefficients. Thus, we have only one array $[r]^{\mathcal{N}}$ for each $\mathcal{N} \subseteq \mathcal{D}$ and do not store the auxiliary coefficients of different grid levels in separate arrays. We then formulate Algorithm 1, which updates the auxiliary coefficients stored in the in-place storage arrays from $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ to $\tilde{\mathcal{H}}_{\boldsymbol{k}'}$. The key ingredient is Theorem 4.1, which is applied in each direction separately.

In direction q, the auxiliary coefficients only change on level $k_q - 1$. We only need a for loop over the auxiliary coefficients. In all other directions $\tilde{q} \in \mathcal{D} \setminus \{q\}$, the

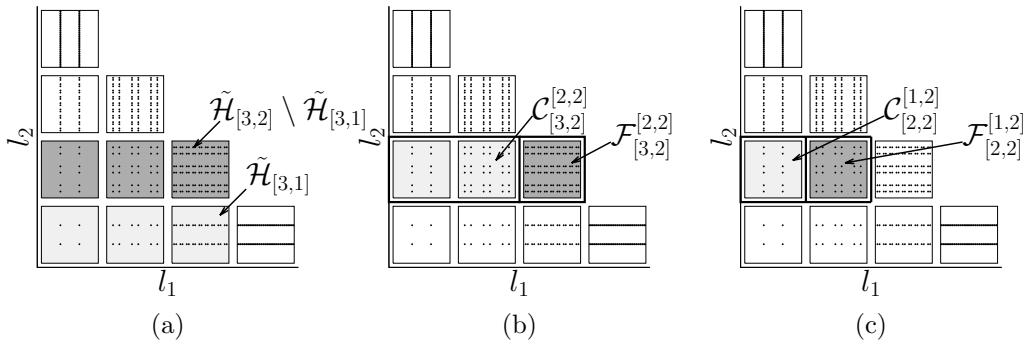


FIG. 4.1. *The plots show the coarse and fine grid points for the coarsening step from $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ to $\tilde{\mathcal{H}}_{\boldsymbol{k}'}$, with $\boldsymbol{k} = [3,2]$ and coarsening direction $q = 2$. In (a), the fine (dark) and coarse (light) grid points for Algorithm 1 operating in direction two (first for loop, lines 4-10) are shown. If Algorithm 1 operators in direction $\tilde{q} = 1$ (second for loop, lines 12-25), then we further partition the fine grid points as shown in (b) and (c).*

coefficients change on all coarse grid levels $k_{\tilde{q}} - 1, \ldots, 1$ and thus the update has to be performed on each coarse grid level separately. This requires an additional for loop that processes one level at a time.

In each direction, the auxiliary coefficients at the coarse grid points are updated with (4.2). Note that we do not have to change or copy the coefficients if $q \notin \mathcal{N}$ because of Theorem 4.1 and the in-place storage scheme. At the fine grid points, the coefficients are hierarchised to remove the dependency on the hierarchically lower hierarchical coefficients, which might change during a relaxation sweep; see Section 4.1. A visualization of the coarse and fine grid points can be found in Figure 4.1.

4.3. Runtime and storage complexity. We show that the runtime of Algorithm 1 scales linearly with the number of grid points $N_{\boldsymbol{k}}$ of the hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}}$. This result is then used to prove that the runtime costs of one multigrid cycle, i.e., of one Q-cycle, of our multigrid method with the auxiliary coefficient representation scale linearly with the number of sparse grid points N.

LEMMA 4.3. *The runtime of Algorithm 1 is linear in the number of grid points $N_{\boldsymbol{k}}$ of the hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}}$.*

Proof. We first note that the update of one auxiliary coefficient with (4.2) has constant runtime, and that one auxiliary coefficient can be hierarchised by applying a stencil; see [4, 21]. Thus, to analyze the runtime of the algorithm, it is sufficient to count the number of auxiliary coefficients that are either updated or hierarchised.

First consider direction q, where the for loop in line 4 in Algorithm 1 iterates over all 2^d subsets $\mathcal{N} \subseteq \mathcal{D}$ and in each iteration the auxiliary coefficients are either updated or hierarchised. Since there are at most $N_{\boldsymbol{k}} = \prod_{i=1}^{d}(3^{k_i}+1) = |\Psi_{\boldsymbol{k}}| = |\tilde{\mathcal{H}}_{\boldsymbol{k}}|$ auxiliary coefficients, the runtime of this for loop is $\mathcal{O}(2^d \cdot N_{\boldsymbol{k}})$. Consider now any other direction $\tilde{q} \in \mathcal{D} \setminus \{q\}$, where we have an additional loop over all levels from $k_{\tilde{q}}$ to 2. In iteration j of this loop, only grid points up to level j are processed. These are

$$(3^j + 1) \cdot \prod_{\substack{i=1 \\ i \neq \tilde{q}}}^{d} (3^{k_i} + 1)$$

Algorithm 1 Coarsen auxiliary coefficients in direction q

procedure COARSEN(q)
  $\boldsymbol{k}' = [k_1, \dots, k_q - 1, \dots, k_d]$
  for $\mathcal{N} \subseteq \mathcal{D}$ **do** ▷ In direction q
    if $q \in \mathcal{N}$ **then**
      Update $[r]^{\mathcal{N}}$ at coarse grid points in $\Psi_{\boldsymbol{k}'}$ with (4.2)
    else if $q \notin \mathcal{N}$ **then**
      Hierarchise $[r]^{\mathcal{N}}$ in direction q at fine grid points in $\Psi_{\boldsymbol{k}} \setminus \Psi_{\boldsymbol{k}'}$
    end if
  end for
  for $\tilde{q} \in \{1, \dots, d\} \setminus \{q\}$ **do** ▷ In all other directions than q
    for $\tilde{\boldsymbol{k}} = [k_1, \dots, k_{\tilde{q}}, \dots, k_d]$ to $[k_1, \dots, 2, \dots, k_d]$ **do**
      $\tilde{\boldsymbol{k}}' = [\tilde{k}_1, \dots, \tilde{k}_{\tilde{q}} - 1, \dots, \tilde{k}_1]$
      $\mathcal{C}_{\tilde{\boldsymbol{k}}}^{\tilde{\boldsymbol{k}}} = \left(\Psi_{\boldsymbol{k}} \setminus \Psi_{\boldsymbol{k}'} \cap \Psi_{\tilde{\boldsymbol{k}}}\right) \cap \Psi_{\tilde{\boldsymbol{k}}'}$ ▷ Test functions at coarse grid points
      $\mathcal{F}_{\tilde{\boldsymbol{k}}}^{\tilde{\boldsymbol{k}}} = \left(\Psi_{\boldsymbol{k}} \setminus \Psi_{\boldsymbol{k}'} \cap \Psi_{\tilde{\boldsymbol{k}}}\right) \setminus \Psi_{\tilde{\boldsymbol{k}}'}$ ▷ Test functions at fine grid points
      for $\mathcal{N} \subseteq \{1, \dots, d\}$ **do**
        if $\tilde{q} \in \mathcal{N}$ **then**
          Update coefficients $[r]^{\mathcal{N}}$ at coarse grid points in $\mathcal{C}_{\tilde{\boldsymbol{k}}}^{\tilde{\boldsymbol{k}}}$ with (4.2)
        else if $\tilde{q} \notin \mathcal{N}$ **then**
          Hierarchise $[r]^{\mathcal{N}}$ in direction $\tilde{q}$ at fine grid points in $\mathcal{F}_{\tilde{\boldsymbol{k}}}^{\tilde{\boldsymbol{k}}}$
        end if
      end for
    end for
  end for
end procedure

grid points. Thus, the loop iterates over

$$\sum_{j=2}^{k_{\tilde{q}}} \left(3^j + 1\right) \cdot \prod_{\substack{i=1 \\ i \neq \tilde{q}}}^{d} \left(3^{k_i} + 1\right) = \left(\frac{3}{2} \cdot \left(3^{k_{\tilde{q}}} - 1\right) + k_{\tilde{q}} - 4\right) \prod_{\substack{i=1 \\ i \neq \tilde{q}}}^{d} \left(3^{k_i} + 1\right) \leq 3 \cdot N_{\boldsymbol{k}}$$

grid points, and we again have a linear runtime in $N_{\boldsymbol{k}}$. Overall, Algorithm 1 has runtime costs of $\mathcal{O}(d \cdot 2^d \cdot N_{\boldsymbol{k}})$. □

THEOREM 4.4. *Let $\mathcal{V}_\ell^{(1)}$ be the sparse grid space of dimension d and level ℓ with N grid points. The runtime costs of one Q-cycle is in $\mathcal{O}(N)$ if the coarse grid equations are represented in the auxiliary coefficients as in* (3.11).

Proof. First, evaluating (3.11) has runtime costs of $\mathcal{O}(1)$ for one test function $\psi_{\boldsymbol{k},\boldsymbol{t}} \in \Psi_{\boldsymbol{k}}$ because the sum has 2^d components and each component can be evaluated by applying a stencil to the auxiliary coefficients. Thus, one relaxation sweep at the grid corresponding to the hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ is in $\mathcal{O}(N_{\boldsymbol{k}})$ and thus linear in the number of grid points, if the auxiliary coefficients are given. Second, no setup phase is required because the Q-cycle of the first multigrid iteration is started at the coarsest grid corresponding to $\tilde{\mathcal{H}}_{[1,\dots,1]}$ with the auxiliary coefficients set to zero. Third, each hierarchical subspace is passed by the Q-cycle c times where c is a constant independent of the sparse grid level ℓ. We thus only have to count the number of grid

points (coarse and fine points) that are processed by one Q-cycle.

A Q-cycle passes through all hierarchical subspaces of a sparse grid space. The number of grid points $\mathcal{O}(3^{|\boldsymbol{k}|_1})$ of a hierarchical subspace $\tilde{\mathcal{H}}_{\boldsymbol{k}}$ is of the same order as of an hierarchical increment $\mathcal{W}_{\boldsymbol{l}}$, where $|\boldsymbol{k}|_1 = \sum_{i=1}^{d} k_i$. Furthermore, the number of subspaces equals the number of increments. According to (2.6), the union of the grid points of all hierarchical increments is the set of the sparse grid points of size N and thus the union of the grid points of the hierarchical subspace also has to have size $\mathcal{O}(N)$. This shows that the coarse grid points, which are included in the subspaces but not in the increments, are asymptotically negligible.

We thus have shown that a Q-cycle processes $\mathcal{O}(N)$ grid points per iteration. Since we know that a relaxation sweep in the auxiliary coefficient representation as well as the update of the auxiliary coefficients is linear in the number of grid points, it follows that the overall runtime of one Q-cycle iteration of our multigrid method is linear in the number of sparse grid points N.
□

We finally consider the storage requirements of our multigrid method based on the auxiliary coefficients. Our in-place storage scheme introduced in Section 4.2 requires one array $[r]^{\mathcal{N}}$ for each $\mathcal{N} \subseteq \mathcal{D}$ of size N. Thus, the storage costs are $\mathcal{O}(2^d \cdot N)$, linear in the number of sparse grids. We emphasize that this is a very conservative estimate because most auxiliary coefficients have a very small absolute value, so it is not necessary store them [10, 13, 14].

5. Numerical results. We present numerical runtime and convergence results of our multigrid method for non-adaptive and adaptive sparse grids in Section 5.1 and 5.2, respectively. The multigrid performance of our method, i.e., runtime scaling linearly with the number of degrees of freedom and convergence factor bounded independently of mesh width, are shown with multi-dimensional Helmholtz and convection-diffusion problems. We also show that the Q-cycle is well-suited for anisotropic problems.

In all examples, two Gauss-Seidel relaxation sweeps are performed on each grid level. Following the comment in Section 4.3, auxiliary quantities are only stored if their absolute value is above 10^{-14}. This does not impact the overall accuracy behavior of the sparse grid solution; see Figure 5.4a. We finally note that we employ sparse grids with coarsening by a factor of three and emphasize that a grid with coarsening by a factor of three on level ℓ has distinctly more grid points than a grid with coarsening by a factor of two on the same level ℓ. Thus, for a better comparison, we not only report the level ℓ in our numerical examples, but also the number of sparse grid points.

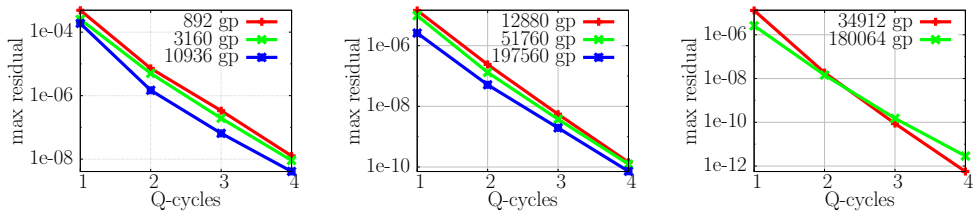
5.1. Non-adaptive sparse grids. In this section, we consider sparse grids without adaptive refinement. Let $\Omega = (0,1)^d$ be the spatial domain. We then define the multi-dimensional Helmholtz problem

$$\begin{aligned} \Delta u + \lambda u &= 0\,, \qquad \text{in } \Omega, \\ u_{|\partial\Omega} &= g_h, \end{aligned} \tag{5.1}$$

where we set $\lambda = 2\pi\sqrt{d-1} + 1$ and

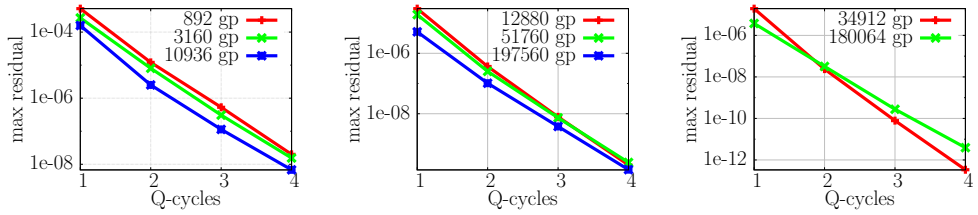
$$g_h(\boldsymbol{x}) = \exp\left(\left(-\sqrt{d-1}\pi + \lambda\right) x_d\right) \prod_{j=1}^{d-1} \sin\left(x_j \pi\right) .$$

Note that the parameter λ changes with the dimension d of the spatial domain. Note further that $u = g_h$ holds. The multigrid method in [26] is not applicable to this



(a) two dimensions (b) three dimensions (c) four dimensions

FIG. 5.1. *The plots show the convergence obtained with our sparse grid multigrid method for the Helmholtz problem* (5.1). *The results indicate a convergence factor bounded independently of the mesh width, i.e., independently of the number of grid points. Note that the plots have different ranges on the y axis.*



(a) two dimensions (b) three dimensions (c) four dimensions

FIG. 5.2. *The plots show the convergence obtained with our sparse grid multigrid method for the convection-diffusion problem* (5.2). *We can observe a convergence factor that seems to be bounded independently of the mesh width. Note that the plots have different ranges on the y axis.*

problem in more than two dimensions. We discretize the PDE (5.1) on two- (levels four to six), three- (levels four to six), and four-dimensional (levels three to four) sparse grids and then solve the corresponding system of linear equations with our multigrid method based on the auxiliary coefficient representation. The convergence results are plotted in Figure 5.1 and do not show a significant dependence on the mesh width, i.e., on the number of grid points. We only require four Q-cycles to reduce the maximum norm of the residual to 10^{-8} and below.

Consider now a multi-dimensional convection-diffusion equation

$$\begin{aligned} -\Delta u + 10\partial_{x_d} u + \lambda u &= 0\,, \qquad \text{in } \Omega, \\ u_{|\partial\Omega} &= g_c\,, \end{aligned} \tag{5.2}$$

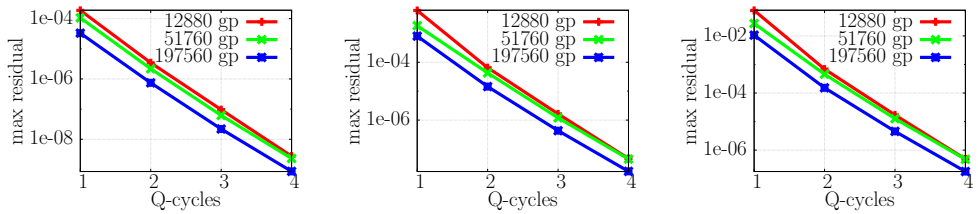
with $\lambda = 10\sqrt{d-1}\pi$ and

$$g_c(\boldsymbol{x}) = \exp\left(-\sqrt{d-1}\pi x_d\right) \prod_{j=1}^{d-1} \sin\left(x_j \pi\right)\,.$$

We note that the multigrid method introduced in [26] is not applicable to this problem because of the convection term. We discretize the problem again on two-, three-, and four-dimensional sparse grids and solve the corresponding system with our multigrid method. The convergence behavior is shown in Figure 5.2. Again, the results indicate a convergence factor that is bounded independently of the mesh width.

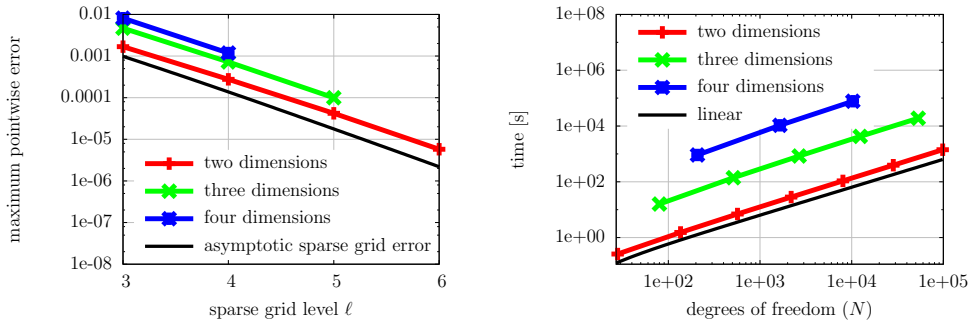
Next, we consider anisotropic Laplace problems

$$\begin{aligned} \epsilon\partial^2_{x_1} u + \textstyle\sum_{i=2}^{d} \partial^2_{x_d} u &= 0\,, \qquad \text{in } \Omega, \\ u_{|\partial\Omega} &= g_h\,, \end{aligned} \tag{5.3}$$



(a) $\epsilon = 10$ (b) $\epsilon = 100$ (c) $\epsilon = 1000$

FIG. 5.3. *The plots show the reduction of the residual obtained with our multigrid method for the anisotropic Laplace problems* (5.3) *in three dimensions. The results confirm that the combination of the proposed multigrid method and the Q-cycle is robust with respect to anisotropy. Note that the y axes have different ranges due to the translation of the norm of the residuals by the factor* ϵ.



(a) discretization error (b) runtime

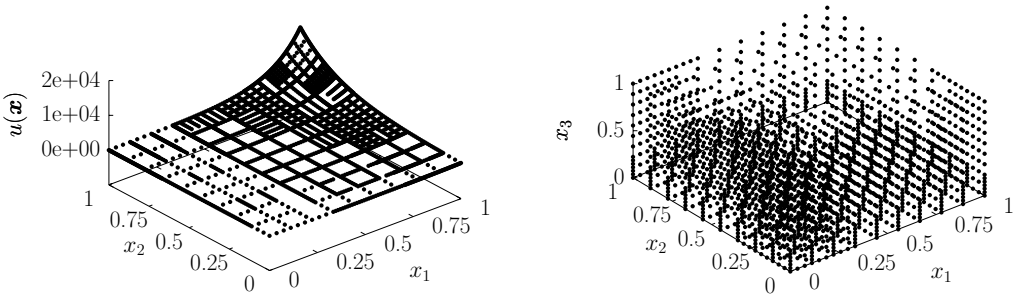
FIG. 5.4. *The plot in (a) shows the discretization or truncation error of the Helmholtz problem* (5.1) *on non-adaptive sparse grids. The obtained numerical results (slope) match the expected asymptotic convergence rate. The plot in (b) shows the linear runtime scaling of one multigrid cycle of the proposed method. The slope is approximately 1.*

with $\epsilon \in \{10, 100, 1000\} \subset \mathbb{R}$. Anisotropic problems cannot be efficiently treated with standard multigrid components [31]. One option to recover multigrid performance is semi-coarsening. Our multigrid method uses the Q-cycle which employs semi-coarsening to reach the grids of all hierarchical subspaces of the sparse grid space. Thus, our method should be robust with respect to anisotropy. The convergence results reported in Figure 5.3 show that this is indeed the case. Our method retains the same convergence behavior independent of the mesh width and the parameter ϵ.

We finally consider the runtime of our multigrid method. In Theorem 4.4, we showed that the runtime of one multigrid cycle scales linearly with the number of degrees of freedom N. Figure 5.4b confirms this result numerically for the Helmholtz problem (5.1). The curves corresponding to the two-, three-, and four-dimensional problems are translated due to the dependency of the runtime on the dimension; see the proofs of Lemma 4.3 and Theorem 4.4. The time measurements were performed on nodes with Intel Xeon E5-1620 CPUs and 32GB RAM.

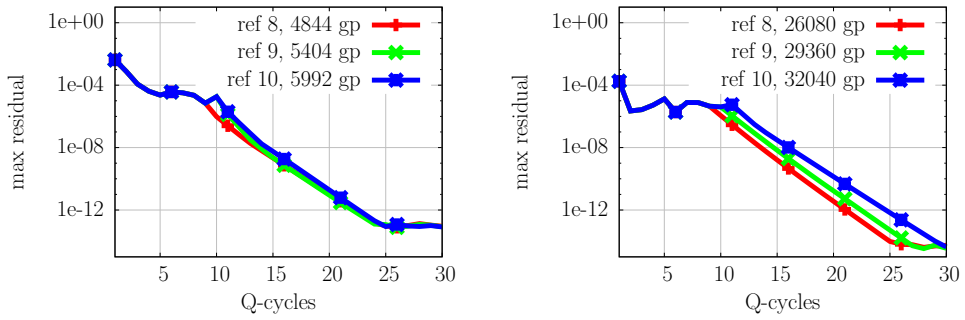
5.2. Adaptive sparse grids. We consider the Helmholtz problem (5.1) and discretize it on adaptively refined sparse grids. The grid is dynamically refined during the multigrid cycles.

We perform up to ten refinement steps. In each step, we refine the sparse grid points corresponding to the hierarchical coefficients with the largest absolute values.



(a) two dimensions (b) three dimensions

FIG. 5.5. *The surface plot in (a) shows the solution function of the Helmholtz problem* (5.1) *on adaptively refined sparse grids. A three-dimensional refined sparse grid is shown in (b).*



(a) two dimensions (b) three dimensions

FIG. 5.6. *The plots show the reduction of the residual achieved with our multigrid method for the Helmholtz problems* (5.1) *discretized on adaptively refined sparse grids. New grid points are added during the first ten multigrid cycles and thus the residual is not reduced there. After these refinements our multigrid method achieves a convergence factor which does not show a significant dependence on the number of refinement steps.*

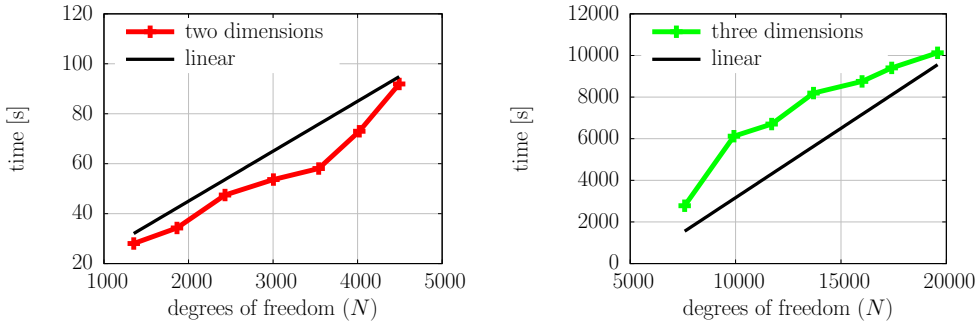
This is a standard refinement criterion in the context of sparse grids [4]. Besides the neighbors of the grid points, we also include certain auxiliary grid points which are required to obtain a valid sparse grid structure. In particular, all hierarchical ancestors have to be created; see [28] for details. With those additional points, we avoid hanging nodes [31]. In Figure 5.5a, the two-dimensional Helmholtz problem on an adaptive sparse grid after six refinement steps is shown. A three-dimensional refined sparse grid is plotted in Figure 5.5b.

Figure 5.6 shows the convergence behavior of our multigrid method on adaptive sparse grids. We refine the first 40 grid points with the largest absolute hierarchical coefficient. The plots indicate that the residual is reduced with a constant factor bounded independently of the number of refinements. A worse convergence for adaptive compared to non-adaptive sparse grids is obtained; however, this is usually compensated in the overall runtime because a distinctly smaller number of degrees of freedom is required to reach a given discretization error; see [31, Section 9.4.2]. In Table 5.1, we report the convergence factors corresponding to the homogeneous Helmholtz problem following [31, Section 2.5.2]. The adaptive sparse grid is gener-

TABLE 5.1

The table reports the convergence factors of our multigrid method for the Helmholtz problem (5.1) *but with boundary* $u_{|\partial\Omega} = 0$ *(homogeneous problem). The adaptive sparse grid is refined up to ten times. The reported results are the geometric mean of the factors measured between multigrid cycle 15 and 20; see [31, Section 2.5.2]. The convergence factors stay almost constant, independent of the number of refinements.*

	ref 4	ref 5	ref 6	ref 7	ref 8	ref 9	ref 10
two dimensions	0.321	0.318	0.406	0.400	0.404	0.392	0.400
three dimensions	0.476	0.330	0.326	0.327	0.458	0.307	0.312



(a) two dimensions

(b) three dimensions

FIG. 5.7. *The plots show the runtime of one multigrid cycle of the proposed method for the Helmholtz problem* (5.1) *discretized on adaptively refined sparse grids. The runtime scales linearly with the number of degrees of freedom.*

ated with Helmholtz problem (5.1) and stored. The homogeneous Helmholtz problem, with boundary $u_{|\partial\Omega} = 0$, is then discretized on the adapted sparse grid and initialized with a random start solution. The convergence factors shown in Table 5.1 are the geometric means of the factors obtained between multigrid cycle 15 and 20, as suggested in [31, Section 2.5.2]. The solution is rescaled after each multigrid cycle to avoid problems with numerical precision. The convergence factors stabilize after the first few refinements to a constant value. Figure 5.7 shows the runtime of one multigrid cycle in case of adaptively refined sparse grids. The results confirm that the runtime of our method scales linearly with the number of degrees of freedom, also in the case of adaptive sparse grids.

6. Conclusions and outlook. We presented a multigrid method that exploits the unconventional structure of systems of linear equations stemming from sparse grid discretizations of multi-dimensional PDEs. The key ingredient was a reformulation of the right-hand sides of the coarse grid equations. We developed a hierarchical ANOVA-like decomposition of the sparse grid solution function and derived so-called auxiliary coefficients. These coefficients hide the computationally cumbersome sparse grid structure and so allowed us to represent the right-hand sides in a nodal point basis on low-dimensional full grids. We then showed that the auxiliary coefficients can be computed during grid transfer with linear costs. This then led to our sparse grid multigrid method with runtime costs bounded linearly in the number of sparse grid points. We demonstrated our method with multi-dimensional Helmholtz and convection-diffusion problems. Our method achieved convergence factors bounded independently of the mesh width, also for anisotropic problems. In case of spatially

adaptive sparse grids, we could confirm that the factors are bounded independently of the number of refinement steps.

Future work could follow two directions. First, the presented multigrid method employs the auxiliary coefficient representation only from an algorithmic point of view; however, the coefficients have a rich structure that reflects the properties of the PDE [10, 13, 14]. This structure could be used to select only a few auxiliary coefficients and thus to adapt the discretization to the PDE. This would introduce another approximation, but it could also significantly speed up the runtime. Second, we introduced the auxiliary coefficients in the context of our multigrid method, but the concept of the auxiliary coefficients is more general. It could be applied to other sparse grid algorithms used in, e.g., data-driven problems. There the optimization problems can often be related to elliptic PDEs and thus could be discretized with the auxiliary coefficients too [24, 28].

REFERENCES

[1] S. Achatz, *Adaptive finite Dünngitter-Elemente höherer Ordnung für elliptische partielle Differentialgleichungen mit variablen Koeffizienten*, PhD thesis, TU München, 2003.

[2] H. Bin Zubair, C. C. W. Leentvaar, and C. W. Oosterlee, *Efficient d-multigrid preconditioners for sparse-grid solution of high-dimensional partial differential equations*, International Journal of Computer Mathematics, 84 (2007), pp. 1131–1149.

[3] H.-J. Bungartz, *A multigrid algorithm for higher order finite elements on sparse grids*, Electronic Transactions on Numerical Analysis, 6 (1997), pp. 63–77.

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

[5] B. Efron and C. Stein, *The jackknife estimate of variance*, Annals of Statistics, 9 (1981), pp. 586–596.

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

[7] M. Griebel, *Sparse grids and related approximation schemes for higher dimensional problems*, in Foundations of Computational Mathematics (FoCM05), Santander, L. Pardo, A. Pinkus, E. Suli, and M.J. Todd, eds., Cambridge University Press, 2006, pp. 106–161.

[8] M. Griebel and H. Harbrecht, *On the convergence of the combination technique*, in Sparse grids and Applications, vol. 97 of Lecture Notes in Computational Science and Engineering, Springer, 2014, pp. 55–74.

[9] M. Griebel and M. Holtz, *Dimension-wise integration of high-dimensional functions with applications to finance*, Journal of Complexity, 26 (2010), pp. 455–489.

[10] M. Griebel and M. Holtz, *Dimension-wise integration of high-dimensional functions with applications to finance*, Journal of Complexity, 26 (2010), pp. 455–489.

[11] M. Griebel and A. Hullmann, *On a multilevel preconditioner and its condition numbers for the discretized laplacian on full and sparse grids in higher dimensions*, in Singular Phenomena and Scaling in Mathematical Models, Michael Griebel, ed., Springer International Publishing, 2014, pp. 263–296.

[12] M. Griebel, A. Hullmann, and P. Oswald, *Optimal scaling parameters for sparse grid discretizations*, Numerical Linear Algebra with Applications, (2014), pp. n/a–n/a.

[13] M. Griebel, F. Y. Kuo, and I. H. Sloan, *The smoothing effect of the ANOVA decomposition*, Journal of Complexity, 26 (2010), pp. 523–551.

[14] ———, *The smoothing effect of integration in $\mathbb{R}^d$ and the ANOVA decomposition*, Math. Comp., 82 (2013), pp. 383–400. Also available as INS preprint No. 1007, 2010.

[15] M. Griebel and P. Oswald, *On additive schwarz preconditioners for sparse grid discretizations*, Numerische Mathematik, 66 (1993), pp. 449–463.

[16] ———, *Tensor product type subspace splittings and multilevel iterative methods for anisotropic problems*, Advances in Computational Mathematics, 4 (1995), pp. 171–206.

[17] M. Griebel, M. Schneider, and C. Zenger, *A combination technique for the solution of sparse grid problems*, in Iterative Methods in Linear Algebra, P. de Groen and R. Beauwens, eds., IMACS, Elsevier, North Holland, 1992, pp. 263–281.

[18] H. Harbrecht, R. Schneider, and C. Schwab, *Multilevel frames for sparse tensor product spaces*, Numerische Mathematik, 110 (2008), pp. 199–220.

[19] M. Hegland, J. Garcke, and V. Challis, *The combination technique and some generalisa-*

tions, Linear Algebra and its Applications, 420 (2007), pp. 249 – 275.
[20] X. Ma and N. Zabaras, *An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations*, Journal of Computational Physics, 229 (2010), pp. 3884–3915.
[21] B. Peherstorfer, *Model Order Reduction of Parametrized Systems with Sparse Grid Learning Techniques*, PhD thesis, TU München, 2013.
[22] B. Peherstorfer and H.-J. Bungartz, *Semi-coarsening in space and time for the hierarchical transformation multigrid method*, Procedia Computer Science, 9 (2012), pp. 2000–2003.
[23] B. Peherstorfer, C. Kowitz, D. Pflüger, and H.-J. Bungartz, *Recent applications of sparse grids*, Numerical Mathematics: Theory, Methods and Applications, (2014). to appear.
[24] B. Peherstorfer, D. Pflüger, and H.-J. Bungartz, *Density Estimation with Adaptive Sparse Grids for Large Data Sets*, SIAM, 2014, ch. 50, pp. 443–451.
[25] C. Pflaum, *Convergence of the combination technique for second-order elliptic differential equations*, SIAM Journal on Numerical Analysis, 34 (1997), pp. 2431–2455.
[26] ———, *A multilevel algorithm for the solution of second order elliptic differential equations on sparse grids*, Numerische Mathematik, 79 (1998), pp. 141–155.
[27] D. Pflüger, *Spatially Adaptive Sparse Grids for High-Dimensional Problems*, Verlag Dr. Hut, 2010.
[28] D. Pflüger, B. Peherstorfer, and H.-J. Bungartz, *Spatially adaptive sparse grids for high-dimensional data-driven problems*, Journal of Complexity, 26 (2010), pp. 508–522.
[29] H. Rabitz and Ö. Alis, *General foundations of high-dimensional model representations*, Journal of Mathematical Chemistry, 25 (1999), pp. 197–233.
[30] C. Schwab and R.A. Todor, *Sparse finite elements for stochastic elliptic problems higher order moments*, Computing, 71 (2003), pp. 43–63.
[31] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, 2001.
[32] G. Wahba, *Spline models for observational data*, SIAM, 1990.
[33] A. Zeiser, *Fast Matrix-Vector multiplication in the Sparse-Grid galerkin method*, Journal of Scientific Computing, (2010), pp. 1–19.