COMPUTATIONAL METHODS IN APPLIED MATHEMATICS, Vol.1(2001), No.3, pp.222–241

ON THE REPRESENTATION OF FUNCTIONS AND FINITE DIFFERENCE OPERATORS ON ADAPTIVE DYADIC GRIDS

P.W. HEMKER

CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
E-mail: pieth@cwi.nl

F. SPRENGEL

Institute for Algorithms and Scientific Computing
GMD German National Research Center for Information Technology
Schloss Birlinghoven, D–53754 Sankt Augustin, Germany
E-mail: sprengel@gmd.de

Abstract — In this paper we describe methods to approximate functions and differential operators on adaptive sparse (dyadic) grids. We distinguish between several representations of a function on the sparse grid and we describe how finite difference (FD) operators can be applied to these representations.
For general variable coefficient equations on sparse grids, genuine finite element (FE) discretizations are not feasible and FD operators allow an easier operator evaluation than the adapted FE operators. However, the structure of the FD operators is complex. With the aim to construct an efficient multigrid procedure, we analyze the structure of the discrete Laplacian in its hierarchical representation and show the relation between the full and the sparse grid case. The rather complex relations, that are expressed by scaling matrices for each separate coordinate direction, make us doubt about the possibility of constructing efficient preconditioners that show spectral equivalence. Hence, we question the possibility of constructing a natural multigrid algorithm with optimal $\mathcal{O}(N)$ efficiency.
We conjecture that for the efficient solution of a general class of adaptive grid problems it is better to accept an additional condition for the dyadic grids (condition L) and to apply adaptive *hp*-discretization.

2000 Mathematics Subject Classification: 65D15; 65N06; 65N30; 65N55.

Keywords: sparse grid, dyadic grid, adaptive methods.

1. Introduction

When applied to d-dimensional problems ($d \geqslant 2$), all numerical methods using regular rectangular grids have one problem in common: *the curse of dimension*. This means that if one refines the grid — for instance, by repeatedly halving the mesh size — the number of grid points grows exponentially with the dimension, i.e., like N^d in $\mathbb{R}^d$ where $N = \mathcal{O}(2^n)$

denotes the number of points per direction. One way out seems to be the use of *sparse grids*, where the number of points only grows like $N\,(\log N)^{d-1}$. Under certain conditions on the mixed derivatives of the approximated function, for sparse grids the approximation accuracy degrades only by a logarithmic factor compared with the accuracy achieved on the regular grid with the same N, see [17, 20, 42].

If we study the efficient representation of functions on a grid in more dimensions, i.e., if we relate the accuracy of the approximation to the number of degrees of freedom (number of data points) used, then it becomes clear that the assumed regularity of the approximated function is the essential issue. For sufficiently smooth functions, more efficient methods are obtained by higher order approximation. In this light, sparse grid approximation can be considered as a method where a particular regularity assumption (viz., bounded mixed derivatives) is exploited, that is intermediate between usual assumptions of C^k-regularity. More efficient than the usual sparse grid approximation, by piecewise (multi-) linear functions, is approximation by a higher order. The same argument holds for higher order sparse grid approximations.

More efficient representations are further obtained by applying mesh refinement not over the whole domain, but *only* in those regions where a function has no sufficient regularity to apply the higher order approximation. So we arrive at the use of adaptive grids. By simultaneous use of many regular, overlapping grids, for which each time the mesh width is halved in one of the coordinate directions, and by using only those parts of the grids that are needed for an efficient representation, we arrive at the adaptive dyadic grid structure. This structure may possibly yield an optimally efficient algorithm for the solution of partial differential equations. In fact, both the usual sparse and the regular full grids are special cases for this structure.

Sparse grids and related methods already have a long tradition in numerical quadrature and approximation theory (see, e.g., [7, 22, 34, 39]). During the last decade, since sparse grid methods have been introduced for the numerical treatment of elliptic boundary problems by Zenger [42], several authors have used sparse grids for solving partial differential equations with finite elements [4–6, 12, 17, 26, 29, 31], finite volumes [18] or spectral methods [23]. Beside this, sparse grids are nowadays used for high-dimensional approximation [3, 37] and numerical integration of functions needed, e.g., in financial mathematics [10,28,41], in wavelet theory [8, 35] or in visualization [38].

Since the efficient evaluation of genuine variable coefficient 3-dimensional finite element operators is impossible, modified Galerkin-type discretizations have been proposed [1, 9, 30]. However, generally these discretizations loose the typical Galerkin nesting property and, e.g., symmetric differential operators turn into asymmetric difference operators. Thus, the proposed Galerkin methods are essentially not more attractive than a finite difference method. Hence, recently also finite difference methods for sparse grids have been developed [13, 16, 32, 33]. Note that consistent finite difference operators on sparse grids *cannot* be chosen as simple stencils involving the nearest neighbors of a point, cf. [32; Appendix]. In [13, 32], the whole machinery for finite differences on sparse grids (which needs linear combination, multiplication, and approximate differentiation) is provided. In Section 3, we show a simple way of presenting the finite difference discretization, and we show a direct construction of the second derivative from the hierarchical coefficients, by recursion.

In Section 4, we restrict ourselves to the special case of Poisson's equation and regular sparse grids in order to analyze the finite difference operator in detail. The resulting matrix is ill-conditioned. So, it takes (even in the Jacobi-preconditioned version [21, 32]) many

iterations for an iterative solver (like BiCGStab) to obtain a solution. Our purpose is to construct Galerkin relations which may motivate efficient multilevel algorithms. An attempt to obtain such an algorithm is demonstrated. However, like [32], we did not succeed in finding a method with a level-independent convergence rate. The Galerkin relations appear to be complex and we doubt if spectral equivalence can be realized. Hence, finally, we conclude that it may be practical to restrict the adaptive structure further in order to obtain a dyadic *hp*-adaptive structure satisfying 'condition L'. For such a structure, in which a unique 'finest' discretization is available, Galerkin-type methods give no special problems.

2. Representation of ASG functions

2.1. Basic notation

To be able to describe adaptive dyadic or sparse grid (ASG) function representations, we first summarize some necessary notation. For background see, e.g., [20].

- Domain $\Omega \subset \mathbb{R}^d$, with coordinates x_j, $j = 1, \ldots, d$.
- Multi-integer: $\mathbf{m} = (m_1, m_2, \ldots, m_d) \in \mathbb{N}_0^d$,
  - $\mathbf{0} = (0, 0, \ldots, 0)$,
  - $\mathbf{e} = (1, 1, \ldots, 1)$,
  - $\mathbf{e}_j = (\ldots, 0, 1, 0, \ldots)$, the j-th unit vector,
  - $|\mathbf{m}| = \sum_{j=1}^d m_j$,
  - $\|\mathbf{m}\| = \prod_{j=1}^d m_j$,
  - $\mathbf{m} < \mathbf{n} \quad \Leftrightarrow \quad m_j < n_j \ \forall j = 1, 2, \ldots, d$.
- Dyadic mesh $\Omega_{\mathbf{k}}$, $\mathbf{k} \geqslant \mathbf{0}$, mesh with dyadic mesh-width $\mathbf{h}_{\mathbf{k}}$,
  - mesh-width: $\mathbf{h}_{\mathbf{k}} = (h_{k_1}, h_{k_2}, \ldots, h_{k_d})$ with $h_{k_i} = 2^{-k_i}$,
  - dyadic grid: $\Omega_{\mathbf{k}}^+ = \{\mathbf{x}_{\mathbf{k},\mathbf{j}} \mid \mathbf{x}_{\mathbf{k},\mathbf{j}} = \mathbf{j}\mathbf{h}_{\mathbf{k}} = (j_1 h_{k_1}, j_2 h_{k_2}, \ldots, j_d h_{k_d})\} \cap \overline{\Omega}$, i.e., the grid of nodal points in the mesh $\Omega_{\mathbf{k}}$,
  - sparse grid: $\Omega_\ell^+ = \bigcup_{|\mathbf{k}|=\ell} \Omega_{\mathbf{k}}^+$,
  - a point with $\|\mathbf{m}\|$ odd is called a hierarchical point,
  - with "$\|\mathbf{j}\|$ odd" we mean: for all $i = 1, 2, \ldots, d$, either j_i is an odd integer, or $k_i = 0$.
- Derivatives: $D^{\mathbf{m}} = \prod_{j=1}^d \frac{\partial^{m_j}}{\partial x_j}$.
- Univariate hat function: $\varphi(x) = \max(0, 1 - |x|)$.
- Univariate Haar function: $\eta(x) = 1$ for $0 < x < 1$, and $\eta(x) = 0$ for $x < 0$ or $x > 1$.
- Basis hat functions: $\varphi_{\mathbf{k},\mathbf{j}}(\mathbf{x}) = \prod_{i=1}^d \varphi(x_i/h_{k_i} - j_i)$.
- Basis Haar functions: $\eta_{\mathbf{k},\mathbf{j}}(\mathbf{x}) = \prod_{i=1}^d \eta(x_i/h_{k_i} - j_i)$.
- Space of piecewise d-linear functions on $\Omega_{\mathbf{k}}$: $V_{\mathbf{k}} = \operatorname{span}\{\varphi_{\mathbf{k},\mathbf{j}} \mid \mathbf{j} \in \mathbb{Z}^d, \mathbf{x}_{\mathbf{k},\mathbf{j}} \in \Omega_{\mathbf{k}}^+\}$.
- Space of hierarchical surpluses on $\Omega_{\mathbf{k}}$:

$$W_{\mathbf{k}} = \operatorname{span}\{\varphi_{\mathbf{k},\mathbf{j}} \mid \|\mathbf{j}\| \text{ odd}, \mathbf{j} \in \mathbb{Z}^d, \mathbf{x}_{\mathbf{k},\mathbf{j}} \in \Omega_{\mathbf{k}}^+\}\,.$$

  Without loss of generality, we assume here that $\mathbf{k} = \mathbf{0}$ yields "the coarsest grid".

2.2. The hierarchical (H-) and lattice (L-) condition

Given a continuous function $u \in C(\Omega)$, we can approximate it by the interpolant $u_{\mathbf{n}} \in V_{\mathbf{n}}$ by means of interpolation on the grid $\Omega_{\mathbf{n}}^+$, i.e.,

$$u_{\mathbf{n}}(\mathbf{x}_{\mathbf{n},\mathbf{j}}) = u(\mathbf{x}_{\mathbf{n},\mathbf{j}}) \quad \forall \mathbf{x}_{\mathbf{n},\mathbf{j}} \in \Omega_{\mathbf{n}}^+ .$$

Obviously, the function $u_{\mathbf{n}}$ on $\Omega_{\mathbf{n}}$ is given by

$$u_{\mathbf{n}} = \sum_{\mathbf{j}} a_{\mathbf{n},\mathbf{j}} \varphi_{\mathbf{n},\mathbf{j}} , \tag{1}$$

where $a_{\mathbf{n},\mathbf{j}} = u(x_{\mathbf{n},\mathbf{j}})$. The error of approximation is wellknown. However, in contrast to the classical approximation we are not interested in approximation for a fixed $\mathbf{n}$, but in the approximation on (the union of) a number of grids $\Omega_{\mathbf{n}}^+$.

We can make the approximation (1) for all grids $\Omega_{\mathbf{n}}^+$ with $\mathbf{n} \geqslant \mathbf{0}$. It can be made arbitrarily accurate for $\mathbf{n}$ large enough, but the number of degrees of freedom increases geometrically with $|\mathbf{n}|$. Therefore, in practice we select a 'smallest' $\mathbf{n}$ such that an accuracy criterion is satisfied. Notice that keeping the representations in *all* coarser $V_{\mathbf{k}}$ (all $V_{\mathbf{k}}$, $\mathbf{0} \leqslant \mathbf{k} \leqslant \mathbf{n}$) does not take essentially more coefficients than the representation on the finest grid in $V_{\mathbf{n}}$ alone.

In order to obtain an efficient approximation, it may be useful to distinguish among different subregions where we make the finest approximation of u in a different $V_{\mathbf{n}}$. We make full and efficient use of the system $\{V_{\mathbf{n}} \mid \mathbf{n} \in \mathbb{N}_0^d\}$, by *in principle* approximating a given function $u \in C(\Omega)$ in *all* $\{V_{\mathbf{n}} \mid \mathbf{n} \in \mathbb{N}_0^d\}$, but using *in practice* only those coefficients that contribute to a sufficiently accurate representation. This implies that, in practice, the function u can be represented in a particular $V_{\mathbf{n}}$ only on part of the domain Ω. To introduce structure in the family of approximating basis functions $\{\varphi_{\mathbf{n},\mathbf{j}}\}$, we introduce the following conditions **H** and **L**, which respectively introduce locally a *hierarchical* and a *lattice* structure.

Condition H: If a basis function $\varphi_{\mathbf{n},\mathbf{j}}$ is used in the representation (1), all corresponding coarser basis functions, i.e., functions $\varphi_{\mathbf{k},\mathbf{i}}$ for which $\mathrm{supp}(\varphi_{\mathbf{k},\mathbf{i}}) \supset \mathrm{supp}(\varphi_{\mathbf{n},\mathbf{j}})$, are also used for the approximation.

Condition L: If two basis functions $\varphi_{\mathbf{n}_1,\mathbf{j}_1}$ and $\varphi_{\mathbf{n}_2,\mathbf{j}_2}$ with $\mathrm{supp}(\varphi_{\mathbf{n}_1,\mathbf{j}_1}) \cap \mathrm{supp}(\varphi_{\mathbf{n}_2,\mathbf{j}_2}) \neq \emptyset$ are used in the representation (1), then the corresponding finer basis function, i.e., a function $\varphi_{\mathbf{n}_3,\mathbf{j}_3}$ for which $\mathrm{supp}(\varphi_{\mathbf{n}_3,\mathbf{j}_3}) = \mathrm{supp}(\varphi_{\mathbf{n}_1,\mathbf{j}_1}) \cap \mathrm{supp}(\varphi_{\mathbf{n}_2,\mathbf{j}_2})$, is also used for the approximation.

Condition **H** is a minimal structure required for sparse grids. Much more structure appears in the partially ordered family of grids $\{\Omega_{\mathbf{n}}\}$ used for the approximation if an approximation satisfies additionally the condition **L**. Then the grids in the (local) approximation form a lattice. Condition **L** holds for the grids used in multigrid methods with semi-coarsening [24, 25]. It guarantees that all coarse grid approximations can be obtained by restriction from a unique finest grid. Typically, sparse grids do not satisfy condition **L**.

2.3. E-, C-, D- and H-representation

We call the representation of the approximation of a function $u \in C(\Omega)$ by a collection of such (partial) approximations in the family of spaces $\{V_{\mathbf{n}}\}$ the *nodal representation* or the *E-representation* of the approximation. This E-representation requires the coefficients $a_{\mathbf{n},\mathbf{j}} = u(\mathbf{x}_{\mathbf{n},\mathbf{j}})$, corresponding to grid points $\mathbf{x}_{\mathbf{n},\mathbf{j}}$, to be equal on the different grids $\Omega_{\mathbf{n}}^+$ at coinciding grid points $\mathbf{x}_{\mathbf{n},\mathbf{j}}$. Thus, because points from coarser grids coincide with those from

finer ones, a certain consistency is required (and a redundancy exists) in the E-representation of an approximation.

During the computation in an approximation process, the representations of the approximations on all different grids $\Omega_{\mathbf{k}}^+$ do not necessarily always satisfy the consistency condition required for the E-representation. In that case an approximation exists of the form of (1) on each separate grid $\Omega_{\mathbf{n}}^+$, and the approximation on the whole system is not uniquely determined. Such a representation is called the *collateral* or *C-representation* of a (nonunique) approximation. There, for different $\mathbf{n}$, the approximations $u_{\mathbf{n}}(\mathbf{x})$ do not necessarily coincide at corresponding grid points $\mathbf{x}_{\mathbf{n},\mathbf{j}}$.

Another way of representing approximations on the family of grids $\{\Omega_{\mathbf{n}}^+\}$ is to partition the approximation over the different grids. Then, instead of (1) the approximation reads

$$u_h = \sum_{\mathbf{n}} \sum_{\mathbf{j}} a_{\mathbf{n},\mathbf{j}} \varphi_{\mathbf{n},\mathbf{j}} \,. \tag{2}$$

In this case, of course, the set of coefficients $\{a_{\mathbf{n},\mathbf{j}}\}$ always determines a unique function u_h. An approximation in this form is called a *distributed* or *D-representation*. However, for a given function u_h, now the coefficients $\{a_{\mathbf{n},\mathbf{j}}\}$ are not uniquely determined because the $\{\varphi_{\mathbf{n},\mathbf{j}}\}$ are linearly dependent.

One way to select a special unique D-representation is by choosing the coefficients $a_{\mathbf{n},\mathbf{j}}$ such that $a_{\mathbf{n},\mathbf{j}} \neq 0$ only for those $(\mathbf{n},\mathbf{j})$ for which $\|\mathbf{j}\|$ is odd. This implies that $a_{\mathbf{n},\mathbf{j}} = 0$ except for a pair $(\mathbf{n},\mathbf{j})$ for which $\Omega_{\mathbf{n}}^+$ is the coarsest grid which contains the nodal point $\mathbf{x}_{\mathbf{n},\mathbf{j}}$. This representation

$$u_h = \sum_{(\mathbf{n},\mathbf{j}),\|\mathbf{j}\| \text{ odd}} a_{\mathbf{n},\mathbf{j}} \varphi_{\mathbf{n},\mathbf{j}} \tag{3}$$

is called the *H-representation* because it represents the approximation in the *hierarchical basis*

$$\left\{ \varphi_{\mathbf{n},\mathbf{j}} \,|\; \mathbf{n} \in \mathbb{N}_0^d, \mathbf{j} \in \mathbb{Z}^d,\ \|\mathbf{j}\| \text{ odd}, \mathbf{x}_{\mathbf{n},\mathbf{j}} \in \Omega_{\mathbf{n}}^+ \right\} , \tag{4}$$

and the part of u_h in

$$W_{\mathbf{n}} = \operatorname{span}\{\varphi_{\mathbf{n},\mathbf{j}} \mid \mathbf{j} \in \mathbb{Z}^d,\ \|\mathbf{j}\| \text{ odd}, \mathbf{x}_{\mathbf{n},\mathbf{j}} \in \Omega_{\mathbf{n}}^+\}$$

is the *hierarchical contribution* from the grid $\Omega_{\mathbf{n}}^+$ to the approximation. We notice that

$$V_{\mathbf{n}} = W_{\mathbf{n}} + \sum_{j=1}^{d} V_{\mathbf{n}-\mathbf{e}_j} = \sum_{\mathbf{0} \leqslant \mathbf{m} \leqslant \mathbf{n}} W_{\mathbf{m}} \,,$$

and that the sparse grid space is defined by

$$V_L = \sum_{0 \leqslant |\mathbf{m}| \leqslant L} W_{\mathbf{m}} \,.$$

When u is interpolated at the nodal points $\mathbf{x}_{\mathbf{n},\mathbf{j}}$, the hierarchical coefficients $a_{\mathbf{n},\mathbf{j}}$ in

$$u(\mathbf{x}_{\mathbf{n},\mathbf{j}}) = \sum_{(\mathbf{n},\mathbf{j}),\|\mathbf{j}\| \text{ odd}} a_{\mathbf{n},\mathbf{j}}\ \varphi_{\mathbf{n},\mathbf{j}}(\mathbf{x}_{\mathbf{n},\mathbf{j}}) \tag{5}$$

are determined by (cf. [20])

$$a_{\mathbf{n},\mathbf{j}} = \prod_{i=1}^{d} \left[-\frac{1}{2}, 1, -\frac{1}{2} \right]_{h_{n_i}\mathbf{e}_i} u(\mathbf{j}\mathbf{h}_{\mathbf{n}}) \,, \tag{6}$$

where $\left[-\frac{1}{2}, 1, -\frac{1}{2}\right]_{h_{n_i}\mathbf{e}_i}$ denotes the difference stencil for the mesh-size h_{n_i} in the i-th coordinate direction. Notice that this expression is well defined for each odd $\mathbf{j}$ because Condition **H** requires that all h_i-neighbors are nodal points in the approximation. Another expression for the coefficient $a_{\mathbf{n},\mathbf{j}}$ is found in the following lemma, see [20].

Lemma 1. *Let $u \in C^{\mathbf{e}+\mathbf{m}}$, for a given $\mathbf{m}$ with $\mathbf{0} \leqslant \mathbf{m} \leqslant \mathbf{e}$. Then, for each $\varphi_{\mathbf{n},\mathbf{j}} \in W_{\mathbf{n}}$, we have*

$$\begin{aligned} a_{\mathbf{n},\mathbf{j}} &= \prod_{i=1}^{d} \left[-\frac{1}{2}, 1, -\frac{1}{2}\right]_{h_{n_i}\mathbf{e}_i} u(\mathbf{j}\mathbf{h}_{\mathbf{n}}) \\ &= (-1)^{|\mathbf{e}+\mathbf{m}|}\, 2^{-d-|\mathbf{n}|} \int_{\Omega} D^{\mathbf{e}+\mathbf{m}} u(\mathbf{x})\, D^{\mathbf{e}-\mathbf{m}} \varphi(2^{\mathbf{n}}\mathbf{x} - \mathbf{j}) d\Omega. \end{aligned} \tag{7}$$

2.4. Transformation of representations and data structure used

From the above, it is clear that each H-representation is a D-representation and each E-representation a C-representation. For piecewise d-linear functions, it is often described [11–13] how a pyramid algorithm can be used to convert an E-representation to a H-representation, and vice versa. Such a conversion can be executed in $\mathcal{O}(N)$ operations, where N is the total number of coefficients (degrees of freedom). The transformation from a D-representation to an H-representation is equally straightforward.

The E-, H-, D-, and C-representations can also be used for piecewise constant functions, and — because of the tensor product structure — discrete function representations can be combined in the different coordinate directions. For example , a discrete function can be piecewise constant in one and piecewise linear in the other coordinate directions. Also, for the piecewise constant functions, efficient pyramid conversion algorithms exist between the different (H-, D-, E-) representation styles. In this case, one should opt for either left- or right-continuity at the discontinuities in the representation.

The data structure used to implement all the above possibilities of an adaptive (sparse) grid representation can be efficient and relatively simple. For the d-dimensional case ($d = 1, 2, 3$), we use the data structure BASIS3 [19,27] that takes the 'patch' $P_{\mathbf{n},\mathbf{j}}$ as an elementary entity. This $P_{\mathbf{n},\mathbf{j}}$ takes all information related to a right-open left-closed cell:

$$\prod_{k=1}^{3} \left[j_k 2^{-n_k}, (j_k + 1)2^{-n_k}\right) .$$

This implies that there exist as many patches in the data structure as there are points used in the description of the approximation. The patches are related to each other by means of pointers in an intertwined tree structure, where each patch has at most 15 pointers to related patches (3 fathers, 6 neighbors, and 6 kids). The data structure is symmetric with respect to any of the coordinate directions.

3. Evaluation of difference operators to ASG functions

Although finite element (FE) discretization of a PDE on a sparse grid is feasible for a constant coefficient problem in two dimensions, finite elements for higher-dimensional problems and variable coefficients give rise to problems. The difficulty arises because, with the hierarchical

basis (4) for test and trial space, the computational complexity of the evaluation of the discrete operator becomes too large. This is caused by the fact that the intersection of the supports of an arbitrary trial and test function is much smaller than the supports of these functions themselves. This has as a consequence that the advantage of sparse grids is lost if the FE discrete operator is evaluated.

The alternative is to essentially modify the FE discretization as in [1, 9, 30] or, as it was already proposed in [13, 32], to use a finite difference discretization. To this end we should be able to apply (approximate) differentiation to discrete representations of approximations as described in Section 2. Then, the application of linear difference operators approximating the linear differential operator

$$\sum_{i,j} \frac{\partial}{\partial x_i} \left(A_{ij}(\mathbf{x}) \frac{\partial}{\partial x_j} \right) + \sum_i B_i(\mathbf{x}) \frac{\partial}{\partial x_i} + C(\mathbf{x}) \tag{8}$$

comes down to the construction of linear combinations of ASG functions, and pointwise multiplication and approximate differentiation of such functions. In any of the representations from Section 2.3 the construction of linear combinations is directly computed by linear combination of the coefficients. Pointwise multiplication is only possible in the E-representation, in which the function values at grid points are directly available. Below we describe differentiation, which requires some more attention, distinguishing between the evaluation of first and second order derivatives. Essentially, FD discretizations are also found in [13, 32] but we take a slightly different point of view which simplifies the description.

3.1. First order derivatives

For a piecewise d-linear ASG function, the derivative $\frac{\partial}{\partial x_i} u_h(\mathbf{x})$ is well defined almost everywhere. Written in D-representation (2), its derivative is simply described by

$$\begin{aligned}
\frac{\partial}{\partial x_i} u_h(\mathbf{x}) \equiv D_i u_h(\mathbf{x}) &= \sum_{\mathbf{n},\mathbf{j}} a_{\mathbf{n},\mathbf{j}} D_i \varphi_{\mathbf{n},\mathbf{j}}(\mathbf{x}) \\
&= \sum_{\mathbf{n},\mathbf{j}} a_{\mathbf{n},\mathbf{j}} \; D_i \prod_{k=1}^{d} \varphi(2^{n_k} x_k - j_k) \\
&= \sum_{\mathbf{n},\mathbf{j}} a_{\mathbf{n},\mathbf{j}} \; (\eta(2^{n_i} x_i - j_i + 1) - \eta(2^{n_i} x_i - j_i)) \prod_{k=1, k\neq i}^{d} \varphi(2^{n_k} x_k - j_k) \,.
\end{aligned}$$

This, again, is a function in D-representation, piecewise constant in the i-direction and piecewise linear in the other directions. It can be described by coefficients associated with nodal points if we decide to choose either a left- or a right-continuous representation.

3.2. Second order derivatives

The computation of second order derivatives of piecewise d-linear ASG functions, $D_i^2 u_h(x)$, seems to be less obvious because second derivatives of the piecewise d-linear functions vanish almost everywhere on Ω. Nevertheless the approximation of the second order derivatives is useful and can easily be derived from the representation of u_h.

In order to approximate $D_i^2 u_h$, we first construct the representation which is the H-representation in the i-th coordinate direction and the E-representation in the other coordinate

directions, i.e., we apply the pyramid algorithm only in the i-th direction. This implies, cf. (7),

$$\begin{aligned} a_{\mathbf{n,j}} &= \left[-\frac{1}{2}, 1, -\frac{1}{2}\right]_{h_{n_i}\mathbf{e}_i} u(\mathbf{j}\mathbf{h}_\mathbf{n}) \\ &= -2^{-1-n_i} \int D_i^2 u(\mathbf{x})\, \varphi(2^{n_i} x_i - j_i) dx_i, \end{aligned} \tag{9}$$

where $x_k = 2^{-n_k} j_k$ for $k \neq i$. The difference from [13, 32] is that we use the hierarchical coefficients in the i-th direction directly instead of forming second differences of an E-representation. This gives us the possibility of describing the approximation of the second derivatives in hierarchical representation explicitly in Section 4.

For the representation of the second derivative, we see that the coefficients in expression (2) are given by (7). It follows that the hierarchical coefficient $a_{\mathbf{n,j}}$ with respect to the i-th coordinate direction corresponds with a measure of $D^2_{x_i} u_h(\mathbf{x})$ in the $\mathbf{h_n}$-neighborhood of $\mathbf{x_{n,j}}$. Considering the i-th coordinate in (9), writing $h = 2^{-n}$, and omitting higher order terms in h, we see

$$\begin{aligned} a_{n,j} &= -\frac{h}{2} \int D_i^2 u(x) \varphi(x/h - j)\, dx \\ &= -\frac{h}{2} \int \left(D_i^2 u(jh) + (\xi h) D_i^3 u(jh) + \frac{(\xi h)^2}{2} D_i^4 u(z) \right) \varphi(\xi)\, d\xi h \\ &= -2^{-1-2n} D_i^2 u(j2^{-n}) + R \end{aligned} \tag{10}$$

with $|R| \leqslant \frac{4}{3} \left(\frac{h}{2}\right)^4 \|D_i^4 u(z)\|$. Here $\|\cdot\|$ denotes the maximum norm in the $\mathbf{h_n}$-neighborhood of $\mathbf{x_{n,j}}$. Such coefficients, and hence such approximate second derivatives, are directly available at the hierarchical points (points with odd j). At points with even j (not at the boundary), we can use

$$\begin{aligned} a_{n,2j} &= -\frac{h}{2} \int D_i^2 u(x)\ \varphi(\frac{x}{h} - 2j)\, dx \\ &= -\frac{h}{2} \int D_i^2 u(x) \left(\varphi\left(\frac{x}{2h} - 2j\right) - \frac{\varphi(\frac{x}{h} - (2j-1)) + \varphi(\frac{x}{h} - (2j+1))}{2} \right) dx \\ &= \frac{1}{2}(a_{n-1,j} - a_{n,j-1} - a_{n,j+1}). \end{aligned} \tag{11}$$

The values $a_{n-1,j}$ are available (by recursion) from coarser grids, where we have

$$a_{n-1,j} \approx -2^{1-2n} D_i^2 u(j2^{1-n})\,.$$

This procedure can be used in any of the coordinate directions i. As a result, we find the nodal representation of the (approximate) second derivative in the i-th direction. If we start with the H-representation (in every direction) of the function, this procedure delivers the second derivative (in the i-th direction) in E-representation in the i-th direction and in H-representation in all other directions. This can simply be converted in either an E- or an H-representation.

4. Finite differences on sparse and full grids

As we learn from [13,32], the finite difference operators on sparse grids have a rather complex structure, including not only finite differences in each direction and linear combinations but also transformations between the E- and the H-representation. The resulting matrix is ill-conditioned and even for the Jacobi-preconditioned version rather many BiCGStab iterations are needed to solve the linear system. On the other hand, finite difference operators on full grids are well studied. They have a simple structure (7-point stencil for the 3D-Laplacian) and multigrid routines are at hand. So, the question arises whether and how we can efficiently use the finite difference operators on the full grids that are contained in the sparse grid in a multigrid routine for solving the problem on the sparse grid.

In order to answer this question, we analyze the discretized operator (8) as described in Section 3. For simplicity, we restrict ourselves to the model problem of Poisson's equation with homogeneous Dirichlet boundary conditions

$$\begin{aligned} -\Delta u &= f \quad \text{in} \quad \Omega, \\ u|_{\delta\Omega} &= 0, \end{aligned} \tag{12}$$

on the cube $\Omega = (0,1)^d$ and a regular sparse grid. We use the hierarchical representations of the functions and their second derivatives. For this special case, we give explicit formulas for the entries of the sparse grid finite difference matrix in hierarchical representation. This will help us to establish relations between the finite difference discretization on sparse and full grids. From this, we will finally propose multilevel-type algorithms to solve the linear system of equations resulting from the finite difference discretization on the sparse grid.

4.1. Characterization of sparse grid and hierarchical points

To introduce the notation for the discussion to follow, it is useful first to investigate the index sets associated with sparse grid points and hierarchical points, respectively. Consider a sparse grid Ω_L^+ of level L on $\Omega = [0,1]^d$ with a finest cell volume $h = 2^{-L}$. Then, every sparse grid point is also a point of a regular grid $\Omega_{L\mathbf{e}}^+$ (with step size h in each direction). Now, we want to characterize the points of the regular grid which belong to the sparse grid, i.e., we wish to characterize the index set $\mathcal{J}_L$ with

$$\mathcal{J}_L = \{\boldsymbol{\ell} \mid x_{L\mathbf{e},\boldsymbol{\ell}} \in \Omega_L^+\}\,.$$

For this, we use the notation of *bit reversing*. Let the integer k, satisfying $0 \leqslant k < 2^\ell$, have the binary representation

$$k = \sum_{s=0}^{\ell-1} k_s\, 2^s, \qquad k_s \in \{0,1\}\,. \tag{13}$$

Then, the binary representation *reflected at position* $\ell - 1$ (or *in bit reversed order*) is given by[1]

$$M_\ell(k) = \sum_{s=0}^{\ell-1} k_s\, 2^{\ell-1-s}\,.$$

We define the bit reversing of a multi-integer $\mathbf{k}$ with respect to a multi-integer $\boldsymbol{\ell}$ as $M_{\boldsymbol{\ell}}(\mathbf{k}) = (M_{\ell_1}(k_1), \ldots, M_{\ell_d}(k_d))$ and with a scalar m by $M_m(\mathbf{k}) = M_{m\mathbf{e}}(\mathbf{k})$.

[1] For $0 \leqslant k \leqslant 2^\ell$, we additionally define $M_\ell(2^\ell) = -1$ and $M_\ell(-1) = 2^\ell$.

For $\mathbf{r} \in \mathbb{N}_0^d$, we also introduce the set of multi-integers within a dyadic range:

$$\mathcal{K}_{\mathbf{r}} = \{\mathbf{k} \mid k_i = 2^{r_i - 1}, \ldots, 2^{r_i} - 1,\ i = 1, \ldots, d\}$$

in order to define[2] the family of multi-integers with dyadic range up to level L:

$$\mathcal{H}_L = \bigcup_{|\mathbf{r}| \leqslant L} \mathcal{K}_{\mathbf{r}}\,.$$

Then one can easily prove (generalization of [35, 36]) that the mapping of bit reversing between the two index sets

$$M_L :\ \mathcal{H}_L \ \rightarrow\ \mathcal{J}_L$$

is a bijection. Furthermore, the set $\bigcup_{|\mathbf{r}|=L} \mathcal{K}_{\mathbf{r}}$ exactly characterizes the grid points that are added to the sparse grid of level $L-1$ to obtain the sparse grid of level L.

One single set $\mathcal{K}_{\mathbf{r}}$ represents (i.e., are bit reversed indices of indices of) the hierarchical points in the subset $\Omega_{\mathbf{r}}^+ \subset \Omega_L^+$. Thus, the hierarchical points in $\Omega_{\mathbf{r}}$ are characterized by

$$\{x_{\mathbf{r},M_{\mathbf{r}}(\mathbf{k})} = x_{L\mathbf{e},M_L(\mathbf{k})} \mid \mathbf{k} \in \mathcal{K}_{\mathbf{r}}\}.$$

Remark 1. The same technique can be used starting on a coarsest grid with smaller mesh width $1/2^{t_0}$ instead of 1. Then M_ℓ has to be defined as a bit reversing w.r.t. $\ell + t_0$ instead of ℓ. If we do not start with powers of 2 the situation is more complex but in principle the same approach can be followed (see [35]).

4.2. Hierarchical representation for finite differences in the univariate case

In equation (11), we gave a recursive expression for the nodal representation of the second derivative. In this section, we find an explicit expression for the hierarchical coefficients of the second derivatives found in Section 3.2. With these explicit formulas, we pave the way for the following sections where explicit formulas for finite differences on full and sparse grids in hierarchical representation are derived from the univariate expressions by tensor products.

We first develop the hierarchical representation for second order finite differences in the univariate case. Then, our model problem reads as

$$-u''(x) = f \quad \text{for } x \in (0,1), \qquad u(0) = u(1) = 0.$$

We interpolate the function u on the equidistant grid with step size h_n by a linear spline u_n. This spline function has the hierarchical representation

$$u_n = \sum_{m=1}^{n} \sum_{k \text{ odd}} a_{m,k} \varphi_{m,k}.$$

Note that, because of the homogeneous boundary conditions, the hierarchical coefficients vanish on the coarsest level. We approximate the second derivative of a function u by the second difference

$$\Delta_{2,h} u(x) = \frac{1}{h^2}(u(x+h) - 2u(x) + u(x-h)).$$

[2] To include multi-integers $\mathbf{k}$ with $\max_i k_i = 2^\ell$, we denote $N_\ell = 2^\ell$ for $\ell \in \mathbb{N}_0$ and $N_{-1} = -1$. and $\mathcal{K}_{\mathbf{r}} = \{\mathbf{k},\ k_i = N_{r_i - 1}, \ldots, N_{r_i} - 1, i = 1, \ldots, d\}$.

From this formula for $\Delta_{2,h_n}u$, we would obtain the wellknown matrix of second differences in nodal representation directly. We now want to find a closed form for its hierarchical representation. In hierarchical points on the finest grid, the hierarchical coefficients represent the second difference $a_{n,k} = -2^{-1-n}\Delta_{2,h_n}u(x_{n,k})$ (cf. Section 3.2). In nonhierarchical points, we can use the recurrence relation (11). Using bit reversing, we write this now in closed form. Let $x_{m,k} \in \Omega_n^+$ be a hierarchical point (k odd) on the subgrid Ω_m^+ with step size h_m, then $M_m(k) \in \mathcal{K}_m$ and writing out the recursion (11), we obtain

$$-\Delta_{2,h_n}u(x_{m,k}) = 2^{1+n}\left(2^m a_{m,k} - \sum_{\ell=m+1}^{n} 2^{\ell-1}(a_{\ell,2^{\ell-m}k-1} + a_{\ell,2^{\ell-m}k+1})\right). \tag{14}$$

Now, we interpolate the nodal approximation on level n of the second derivative of the function u by a linear spline[3] $\tilde{u}_n$ and obtain

$$-\tilde{u}_n = \sum_{k=1}^{2^n-1} -\Delta_{2,h_n}u(x_{n,k})\,\varphi_{n,k} = \sum_{m=1}^{n} \sum_{k \text{ odd}} b_{m,k}\varphi_{m,k}\,. \tag{15}$$

This defines the coefficients $b_{m,k}$ which clearly satisfy

$$b_{m,k} = \frac{1}{2}(\Delta_{2,h_n}u(x_{m,k-1}) - 2\Delta_{2,h_n}u(x_{m,k}) + \Delta_{2,h_n}u(x_{m,k+1}))$$

for hierarchical points $x_{m,k}$. The nonhierarchical points $x_{m,k-1}$ and $x_{m,k+1}$ appear in this formula, too. If we want to apply formula (14), we need to characterize these points (nonhierarchical on level m) as hierarchical points on some coarser level. For this, we need some further notation.

Let k have the binary representation (13). Now we need to characterize the indices m_- and m_+ for which $M_m(k-1) \in \mathcal{K}_{m_-}$ and $M_m(k+1) \in \mathcal{K}_{m_+}$, where $m_\pm$ denotes the level on which $x_{m,k\pm1}$ is hierarchical. We see that

$$m_- = \begin{cases} 0 & \text{for } k = 1, \\ \max\limits_{s=1,\ldots,m-1}\{m-1-s, k_s = 1\} & \text{otherwise,} \end{cases}$$

and

$$m_+ = \begin{cases} 0 & \text{for } k = 2^m - 1, \\ \max\limits_{s=1,\ldots,m-1}\{m-1-s, k_s = 0\} & \text{otherwise.} \end{cases}$$

Thus, if the points $x_{m,k}$ are not next to the boundary, i.e., for[4] $m_\pm \neq 0$,

$$-\Delta_{2,h_n}u(x_{m,k\pm1}) = 2^{1+n}\Bigg(2^{m_\pm}a_{m_\pm,2^{m_\pm-m}(k\pm1)} - \sum_{\ell=m_\pm+1}^{n} 2^{\ell-1}\big(a_{\ell,2^{\ell-m}(k\pm1)-1} + a_{\ell,2^{\ell-m}(k\pm1)+1}\big)\Bigg)\,.$$

[3] For notational convenience, we set in the end points $\Delta_{2,h_n}u(0) = \Delta_{2,h_n}u(1) = 0$.

[4] For points next to the boundary, i.e., for $m_\pm = 0$, we use the notation from the last three footnotes and set $\Delta_{2,h_n}u(x_{m,k\pm1}) = 0$.

Collecting all terms, we get the hierarchical coefficient

$$\begin{aligned} b_{m,k} = 2^{1+n}\Bigg(& 2^m a_{m,k} - \sum_{\ell=m+1}^{n} 2^{\ell-1}(a_{\ell,2^{\ell-m}k-1} + a_{\ell,2^{\ell-m}k+1}) \\ & - 2^{m_--1} a_{m_-,2^{m_--m}(k-1)} + \sum_{\ell=m_-+1}^{n} 2^{\ell-2}(a_{\ell,2^{\ell-m}(k-1)-1} + a_{\ell,2^{\ell-m}(k-1)+1}) \\ & - 2^{m_+-1} a_{m_+,2^{m_+-m}(k+1)} + \sum_{\ell=m_++1}^{n} 2^{\ell-2}(a_{\ell,2^{\ell-m}(k+1)-1} + a_{\ell,2^{\ell-m}(k+1)+1})\Bigg) \end{aligned} \tag{16}$$

with modifications for $m_\pm = 0$. Note that this expression depends on m, k, and n, i.e., it depends not only on the point $x_{m,k}$ but also on the finest grid chosen.

We can write the coefficients $a_{m,k}$ and $b_{m,k}$ in vector notation and describe the transformation process by a matrix. For this, we combine hierarchical coefficients in a vector as

$$\boldsymbol{v}_n = \big(v_{m,k}\big)_{1\leqslant m\leqslant n, M_m(k)\in\mathcal{K}_m} \tag{17}$$

in order to define the vectors $\boldsymbol{a}_n$ and $\boldsymbol{b}_n$, and nodal coefficients simply as

$$\boldsymbol{u}_n = \big(u(x_{n,1}), \ldots, u(x_{n,2^n-1})\big)^T .$$

In this way, we define the matrices $\boldsymbol{A}_n$ and $\boldsymbol{H}_n$ by

$$\boldsymbol{A}_n\, \boldsymbol{a}_n = \boldsymbol{b}_n \qquad \text{and} \qquad \boldsymbol{H}_n\, \boldsymbol{u}_n = \boldsymbol{a}_n.$$

By construction, $\boldsymbol{H}_n^{-1}\boldsymbol{A}_n\boldsymbol{H}_n$ is the usual matrix of second differences in nodal representation and hence symmetric positive definite. Thus, the matrix $\boldsymbol{A}_n$ of finite differences in terms of hierarchical coefficients has the same, i.e., only real positive eigenvalues. However, it is not symmetric and cannot be symmetric positive definite.

Solving the discretized system in its hierarchical form now comes down to solving

$$\boldsymbol{A}_n\, \boldsymbol{a}_n = \boldsymbol{H}_n \big(f(x_{n,1}), \ldots, f(x_{n,2^n-1})\big)^T .$$

4.3. Hierarchical representation for finite differences on the sparse grid

To gain some more insight into the structure of the sparse grid FD operator, we want to obtain an explicit expression for the hierarchical coefficients of the discrete Laplacian. That is, we look for an expression for the elements of the FD matrix $\boldsymbol{A}_L$. To this end, we investigate the discrete second derivative in the x_1-direction first.

Let L be the (highest) level of the sparse grid. Let u_L be given in H-representation:

$$u_L = \sum_{|\mathbf{m}|\leqslant L} \sum_{\|\boldsymbol{\ell}\| \text{ odd}} a_{\mathbf{m},\boldsymbol{\ell}}\, \varphi_{\mathbf{m},\boldsymbol{\ell}}.$$

Because of the boundary conditions, here and in the sequel the notation $|\mathbf{m}| \leqslant L$ means: $\mathbf{m} > \mathbf{0}$, $|\mathbf{m}| \leqslant L$. (The coefficients responsible for the boundary are zero.) Let $\tilde{u}_L$ be the approximation of the resulting function Δu on the sparse grid Ω_L^+ with

$$-\tilde{u}_L = \sum_{|\mathbf{m}|\leqslant L} \sum_{\|\boldsymbol{\ell}\| \text{ odd}} b^{\Delta}_{\mathbf{m},\boldsymbol{\ell}}\, \varphi_{\mathbf{m},\boldsymbol{\ell}}\,.$$

We denote the approximation of the second derivative in the x_ν-direction on the sparse grid Ω_L^+ by

$$-\tilde{u}_L^{(\nu)} = \sum_{|\mathbf{m}|\leqslant L} \sum_{\|\boldsymbol{\ell}\| \text{ odd}} b_{\mathbf{m},\boldsymbol{\ell}}^{(\nu)}\, \varphi_{\mathbf{m},\boldsymbol{\ell}}.$$

Then, the hierarchical coefficients obviously fulfill

$$b_{\mathbf{m},\boldsymbol{\ell}}^{\Delta} = b_{\mathbf{m},\boldsymbol{\ell}}^{(1)} + b_{\mathbf{m},\boldsymbol{\ell}}^{(2)} + \cdots + b_{\mathbf{m},\boldsymbol{\ell}}^{(d)}\,. \tag{18}$$

Let $\mathbf{x}_{\mathbf{m},\mathbf{k}} \in \Omega_L^+$ be a hierarchical point on the grid $\Omega_{\mathbf{m}}^+$. The full grid $\Omega_{\mathbf{j}}^+ \subset \Omega_L$, which is the finest in the x_1-direction such that $\mathbf{x}_{\mathbf{m},\mathbf{k}} \in \Omega_{\mathbf{j}}^+$, is characterized by the multi-index $\mathbf{j} = (m_1 + L - |\mathbf{m}|, m_2, \ldots, m_d)$. Then the hierarchical coefficient of the second difference in the x_1-direction at point $\mathbf{x}_{\mathbf{m},\mathbf{k}}$ is

$$b_{\mathbf{m},\mathbf{k}}^{(1)} = -\alpha_{\mathbf{m},\mathbf{k}} + \frac{1}{2}\big(\alpha_{\mathbf{m},\mathbf{k}-\mathbf{e}_1} + \alpha_{\mathbf{m},\mathbf{k}+\mathbf{e}_1}\big)\,, \tag{19}$$

where $\alpha_{\mathbf{m},\boldsymbol{\ell}}$ denotes the coefficient of the approximation of the second derivative in the x_1-direction (E-representation in the x_1-direction, H-representation in all other directions) at the point $\mathbf{x}_{\mathbf{m},\boldsymbol{\ell}}$.

By construction $M_{\mathbf{m}}(\mathbf{k}) \in \mathcal{K}_{\mathbf{m}}$. Now using the results of the previous section, we obtain from the finite difference operator in the x_1-direction, see (14),

$$\alpha_{\mathbf{m},\mathbf{k}} = -2^{1+2m_1+L-|\mathbf{m}|}\Bigg(a_{\mathbf{m},\mathbf{k}} - \sum_{\ell=m_1+1}^{m_1+L-|\mathbf{m}|} 2^{\ell-m_1-1}\big(a_{(\ell,m_2,\ldots,m_d),(2^{\ell-m_1}k_1-1,k_2,\ldots,k_d)} + a_{(\ell,m_2,\ldots,m_d),(2^{\ell-m_1}k_1+1,k_2,\ldots,k_d)}\big)\Bigg).$$

In the same way as in the previous section, we denote by m_{1-} and m_{1+} the indices for which $M_{m_1}(k_1-1) \in \mathcal{K}_{m_{1-}}$ and $M_{m_1}(k_1+1) \in \mathcal{K}_{m_{1+}}$ (i.e., m_{1-} and m_{1-} are the x_1-levels on which $\mathbf{x}_{\mathbf{m},\mathbf{k}}$ are hierarchical points in the x_1-direction). Now we can characterize the remaining terms in (19) and obtain

$$\begin{aligned} b_{\mathbf{m},\mathbf{k}}^{(1)} = 2^{1+2m_1+L-|\mathbf{m}|} \;\cdot\; \Bigg(& a_{\mathbf{m},\mathbf{k}} - \sum_{\ell=m_1+1}^{m_1+L-|\mathbf{m}|} 2^{\ell-m_1-1}\big(a_{(\ell,m_2,\ldots,m_d),(2^{\ell-m_1}k_1-1,k_2,\ldots,k_d)} \\ &+ a_{(\ell,m_2,\ldots,m_d),(2^{\ell-m_1}k_1+1,k_2,\ldots,k_d)}\big) \\ &- 2^{m_{1+}-m_1-1} a_{(m_{1+},m_2,\ldots,m_d),(2^{m_{1+}-m_1}(k_1+1),k_2,\ldots,k_d)} \\ &+ \sum_{\ell=m_{1+}+1}^{m_1+L-|\mathbf{m}|} 2^{\ell-m_1-2}\big(a_{(\ell,m_2,\ldots,m_d),(2^{\ell-m_1}(k_1+1)-1,k_2,\ldots,k_d)} \\ &+ a_{(\ell,m_2,\ldots,m_d),(2^{\ell-m_1}(k_1+1)+1,k_2,\ldots,k_d)}\big) \\ &- 2^{m_{1-}-m_1-1} a_{(m_{1-},m_2,\ldots,m_d),(2^{m_{1-}-m_1}(k_1-1),k_2,\ldots,k_d)} \\ &+ \sum_{\ell=m_{1-}+1}^{m_1+L-|\mathbf{m}|} 2^{\ell-m_1-2}\big(a_{(\ell,m_2,\ldots,m_d),(2^{\ell-m_1}(k_1-1)-1,k_2,\ldots,k_d)} \\ &+ a_{(\ell,m_2,\ldots,m_d),(2^{\ell-m_1}(k_1-1)+1,k_2,\ldots,k_d)}\big)\Bigg) \end{aligned} \tag{20}$$

(with obvious modifications for points next to the boundary, where $m_{1-} = 0$ or $m_{1+} = 0$). Here, the finest grid in the x_1-direction depends on the other coordinates of the evaluation point. Therefore, $b^{(1)}_{\mathbf{m},\mathbf{k}}$ depends on $\mathbf{m}, \mathbf{k}$ and the highest level L.

Now, we write the hierarchical coefficients $b^{(\nu)}_{\mathbf{m},\mathbf{k}}$, $b^{\Delta}_{\mathbf{m},\mathbf{k}}$, and $a_{\mathbf{m},\mathbf{k}}$ in vector form as $\boldsymbol{b}^{(\nu)}_L$, $\boldsymbol{b}^{\Delta}_L$, and $\boldsymbol{a}_L$, respectively, using the d-dimensional version of (17):

$$\boldsymbol{v}_L = \big(v_{\mathbf{m},\mathbf{k}}\big)_{|\mathbf{m}|\leqslant L, M_{\mathbf{m}}(\mathbf{k})\in\mathcal{K}_{\mathbf{m}}}\,.$$

We define the matrices $\boldsymbol{A}_L$ and $\boldsymbol{A}^{(\nu)}_L$ by

$$\boldsymbol{A}_L\boldsymbol{a}_L = \boldsymbol{b}^{\Delta}_L \quad \text{and} \quad \boldsymbol{A}^{(\nu)}_L\boldsymbol{a}_L = \boldsymbol{b}^{(\nu)}_L\,, \tag{21}$$

respectively. Then, obviously $\boldsymbol{A}_L = \boldsymbol{A}^{(1)}_L + \boldsymbol{A}^{(2)}_L + \cdots + \boldsymbol{A}^{(d)}_L$.

Remark 2. Note that the formulas (16), (20) and the corresponding formula on full grids can also be used in the case of *adaptive* grids and deliver another representation of the same finite differences as in [32]. The sums run over the hierarchical points from the coarser to the finer grids. Leaving out points on finer grids (if the hierarchical coefficients are set to zero) only shortens the sums. The restriction to regular sparse grids is only for simplicity of presentation.

4.4. Hierarchical representation for finite differences on full grids

The finite difference discretization of (12) on a full grid is wellknown (the usual 7-point stencil in 3D). Here we rewrite its matrix in the hierarchical representation in order to unfold the relation between the finite difference operators on full and sparse grids, with the aim to find proper multigrid algorithms. Let a discrete function on $\Omega^+_{\mathbf{n}}$, say $u_{\mathbf{n}}$, be given in its hierarchical representation

$$u_{\mathbf{n}} = \sum_{\mathbf{0}<\mathbf{m}\leqslant\mathbf{n}} \sum_{\|\mathbf{k}\| \text{ odd}} a_{\mathbf{m},\mathbf{k}}\,\varphi_{\mathbf{m},\mathbf{k}}\,.$$

Let $\tilde{u}^{\Delta}_{\mathbf{n}}$ denote the approximation of Δu, similar to (15) in the one-dimensional case in Section 4.2,

$$-\tilde{u}^{\Delta}_{\mathbf{n}} = \sum_{\mathbf{0}<\mathbf{m}\leqslant\mathbf{n}} \sum_{\|\mathbf{k}\| \text{ odd}} \tilde{b}^{\Delta}_{\mathbf{m},\mathbf{k}}\,\varphi_{\mathbf{m},\mathbf{k}}.$$

Assume that $\tilde{b}^{(\nu)}_{\mathbf{m},\mathbf{k}}$ are hierarchical coefficients in the approximation of the second derivative in the x_ν-direction $\tilde{u}^{(\nu)}_{\mathbf{n}}$. The coefficients corresponding to the subgrid $\Omega^+_{\mathbf{n}} \subset \Omega^+_L$ are written in vector-form as

$$\boldsymbol{v}_{\mathbf{n}} = \big(v_{\mathbf{m},\mathbf{k}}\big)_{\mathbf{m}\leqslant\mathbf{n}, M_{\mathbf{m}}(\mathbf{k})\in\mathcal{K}_{\mathbf{m}}}.$$

Similar to (18) in the previous section, we have $\tilde{\boldsymbol{b}}^{\Delta}_{\mathbf{n}} = \tilde{\boldsymbol{b}}^{(1)}_{\mathbf{n}} + \tilde{\boldsymbol{b}}^{(2)}_{\mathbf{n}} + \cdots + \tilde{\boldsymbol{b}}^{(d)}_{\mathbf{n}}$, and we write

$$\boldsymbol{A}^{(\nu)}_{\mathbf{n}}\boldsymbol{a}_{\mathbf{n}} = \tilde{\boldsymbol{b}}^{(\nu)}_{\mathbf{n}} \qquad \text{and} \qquad \boldsymbol{A}_{\mathbf{n}}\boldsymbol{a}_{\mathbf{n}} = \tilde{\boldsymbol{b}}^{\Delta}_{\mathbf{n}}.$$

Obviously,

$$\boldsymbol{A}_{\mathbf{n}} = \boldsymbol{A}^{(1)}_{\mathbf{n}} + \boldsymbol{A}^{(2)}_{\mathbf{n}} + \ldots + \boldsymbol{A}^{(d)}_{\mathbf{n}}\,. \tag{22}$$

We introduce a one-dimensional index set by $\mathcal{I}_n = \bigcup_{m=1}^{n} \{(m,k) \mid \ M_m(k) \in \mathcal{K}_m\}$ and the corresponding identity matrix by

$$\boldsymbol{I}_n = \big(\delta_{\mu,\nu}\big)_{\mu,\nu=1}^{\#(\mathcal{I}_n)}.$$

Then, for the d-dimensional case $\{(\mathbf{m},\mathbf{k}) \mid \ M_{\mathbf{m}}(\mathbf{k}) \in \mathcal{K}_{\mathbf{m}}\} = \mathcal{I}_{n_1} \times \mathcal{I}_{n_2} \times \cdots \times \mathcal{I}_{n_d}$ and we find the matrix of second differences in the ν-th direction as the Kronecker product

$$\boldsymbol{A}_{\mathbf{n}}^{(\nu)} = \boldsymbol{I}_{n_1} \otimes \cdots \otimes \boldsymbol{I}_{n_{\nu-1}} \otimes \boldsymbol{A}_{n_\nu} \otimes \boldsymbol{I}_{n_{\nu+1}} \otimes \cdots \otimes \boldsymbol{I}_{n_d}.$$

where $\boldsymbol{A}_{n_\nu}$ are the same as in Section 4.2. As a Kronecker sum of the matrices $\boldsymbol{A}_{n_\nu}$, the matrix $\boldsymbol{A}_{\mathbf{n}}$ also has only positive eigenvalues. This is not surprising because $\boldsymbol{A}_{\mathbf{n}}$ has the same eigenvalues as the finite difference matrix for the nodal representation on the full grid which is symmetric positive definite.

4.5. Relations between finite differences on full and sparse grids

In this section, we establish relations between the discrete Laplacian on full and on sparse grids. If one considers the Galerkin or FEM approach for discretization on sparse grids, using the hierarchical basis or the generating system of nodal bases functions as test and trial functions, one obtains simple relations between the stiffness matrices for the sparse and the full grids. The basis functions for a full grid $\Omega_{\mathbf{n}}^{+} \subset \Omega_L^{+}$ simply form a subset of the hierarchical basis/generating system for the sparse grid. So, by construction it can be written as a Galerkin product using the matrix for the sparse grid. As a result, one can immediately write down multiplicative subspace correction algorithms like in [15] and interpret them as block iteration methods.

In the finite difference case things are different. In contrast to the sparse grid, on a full grid, for a constant coefficient differential equation the evaluation of the finite difference at a certain point does not depend on the location of the point. For this reason, the finite difference matrix of a full grid $\Omega_{\mathbf{n}}^{+}$ cannot be written as a Galerkin product including the finite difference matrix of the sparse grid Ω_L^{+} of level $L \geqslant |\mathbf{n}|$. This will become immediately clear from equation (24), below.

In what follows, we discuss the kind of relation between the finite difference operators for the Laplacian discretized on a full and on a sparse grid, and we propose a solution algorithm resulting from this.

4.5.1. Full grids as subgrids of the sparse grid. Assume $|\mathbf{n}| \leqslant L$. We define a prolongation $\boldsymbol{P}_{L,\mathbf{n}} : V_{\mathbf{n}} \to V_L$ by $\boldsymbol{w}_L = \boldsymbol{P}_{L,\mathbf{n}} \boldsymbol{v}_{\mathbf{n}}$ with

$$w_{\mathbf{m},\mathbf{k}} = \begin{cases} v_{\mathbf{m},\mathbf{k}} & \text{for } \mathbf{m} \leqslant \mathbf{n}, \\ 0 & \text{otherwise,} \end{cases}$$

where $v_{\mathbf{m},\mathbf{k}}$ and $w_{\mathbf{m},\mathbf{k}}$ are hierarchical coefficients. This represents linear interpolation on the sparse grid. Further, we need a direction-dependent restriction $\boldsymbol{R}_{\mathbf{n},L}^{(\nu)} : V_L \to V_{\mathbf{n}}$ by $\boldsymbol{w}_{\mathbf{n}} = \boldsymbol{R}_{\mathbf{n},L}^{(\nu)} \boldsymbol{v}_L$ with

$$w_{\mathbf{m},\mathbf{k}} = 2^{n_\nu - m_\nu - L + |\mathbf{m}|} v_{\mathbf{m},\mathbf{k}} \quad \text{for } \mathbf{m} \leqslant \mathbf{n}\,. \tag{23}$$

The scaling factor in the above definition arises from the difference in scaling between (16) and (20). Comparing these two formulas and keeping (21) and (22) in mind, we see that,

with these $\boldsymbol{P}_{L,\mathbf{n}}$ and $\boldsymbol{R}^{(\nu)}_{\mathbf{n},L}$, we can write the matrix of finite differences for the Laplacian on the full grid as a sum of Galerkin products using the direction (and point) dependent weighted finite differences for the second derivatives in each direction:

$$\boldsymbol{A}_{\mathbf{n}} = \big(\boldsymbol{R}^{(1)}_{\mathbf{n},L}\boldsymbol{A}^{(1)}_{L} + \boldsymbol{R}^{(2)}_{\mathbf{n},L}\boldsymbol{A}^{(2)}_{L} + \cdots + \boldsymbol{R}^{(d)}_{\mathbf{n},L}\boldsymbol{A}^{(d)}_{L}\big)\boldsymbol{P}_{L,\mathbf{n}}. \tag{24}$$

Unfortunately, this representation is not in a Galerkin form suited for multigrid-type algorithms. We can see from (24) that the finite difference operators on full grids themselves cannot serve as good approximations for the finite difference operator on the sparse grid.

4.5.2. Full grids as subgrids of the sparse grid — An alternative approach. In the previous paragraph, we scaled the different parts, $A^{(\nu)}_{L}$, of the discrete Laplacian for the sparse grid for the different directions. On the other hand, we can scale $A^{(\nu)}_{\mathbf{n}}$, i.e., the directional parts of the matrix responsible for the full grid. With the scaling matrices

$$\begin{aligned}\boldsymbol{M}^{(\nu)}_{\mathbf{n}} &= \operatorname{diag}\big(2^{m_\nu - n_\nu}\big)_{\mathbf{m}\leqslant\mathbf{n}, M_{\mathbf{m}}(\mathbf{k})\in\mathcal{K}_{\mathbf{m}}} \\ &= \boldsymbol{I}_{n_1}\otimes\cdots\otimes\boldsymbol{I}_{n_{\nu-1}}\otimes\big(\operatorname{diag}\big(2^{m_\nu-n_\nu}\big)_{(m_\nu,k_\nu)\in\mathcal{I}_{n_\nu}}\big)\otimes\boldsymbol{I}_{n_{\nu+1}}\otimes\cdots\otimes\boldsymbol{I}_{n_d}\end{aligned}$$

and (22), we define

$$\tilde{\boldsymbol{A}}_{\mathbf{n}} = \boldsymbol{M}^{(1)}_{\mathbf{n}}\boldsymbol{A}^{(1)}_{\mathbf{n}} + \boldsymbol{M}^{(2)}_{\mathbf{n}}\boldsymbol{A}^{(2)}_{\mathbf{n}} + \cdots + \boldsymbol{M}^{(d)}_{\mathbf{n}}\boldsymbol{A}^{(d)}_{\mathbf{n}}. \tag{25}$$

This matrix is again a Kronecker sum of matrices with positive eigenvalues and so it has only positive eigenvalues. Introducing another, differently scaled, restriction that is now independent of the direction ν, $\tilde{\boldsymbol{R}}_{\mathbf{n},L} : V_L \to V_{\mathbf{n}}$, defined by $\boldsymbol{w}_{\mathbf{n}} = \tilde{\boldsymbol{R}}_{\mathbf{n},L}\boldsymbol{v}_L$ with

$$w_{\mathbf{m},\mathbf{k}} = 2^{-L+|\mathbf{m}|}v_{\mathbf{m},\mathbf{k}} \quad \text{for } \mathbf{m}\leqslant\mathbf{n}\,,$$

the new matrix can be written as the Galerkin product

$$\tilde{\boldsymbol{A}}_{\mathbf{n}} = \tilde{\boldsymbol{R}}_{\mathbf{n},L}\boldsymbol{A}_L\boldsymbol{P}_{L,\mathbf{n}}.$$

This Galerkin relation may lead to the following solution algorithm which uses the matrices $\tilde{\boldsymbol{A}}_{\mathbf{n}}$. Denote the vector of hierarchical coefficients of the right-hand side by $\boldsymbol{f}_L$, and let the approximation $\boldsymbol{a}_L$ of the hierarchical coefficients of u_L be given. Then the next iteration step is

for all $|\mathbf{n}| = k$
do $\boldsymbol{a}_L := \boldsymbol{a}_L + \boldsymbol{P}_{L,\mathbf{n}}\tilde{\boldsymbol{A}}^{-1}_{\mathbf{n}}\tilde{\boldsymbol{R}}_{\mathbf{n},L}(\boldsymbol{f}_L - \boldsymbol{A}_L\boldsymbol{a}_L)$ (26)
enddo

This defect correction algorithm of multiplicative type corresponds to a block iterative solver for the discrete Laplacian on the sparse grid with overlapping blocks. Seen as such a block iterative solver this is to some extent similar to the FEM case (see [15]), but in the case of finite differences the blocks (containing point dependent scaling) cannot really be seen as a finite difference discretization on a certain full grid. Nevertheless, in the special case of the Laplacian it is still possible to give a relation (25) between the blocks and the finite difference discretizations.

To improve the convergence rate of the above algorithm we might think of using more than one level and/or more than one iteration per level which leads to

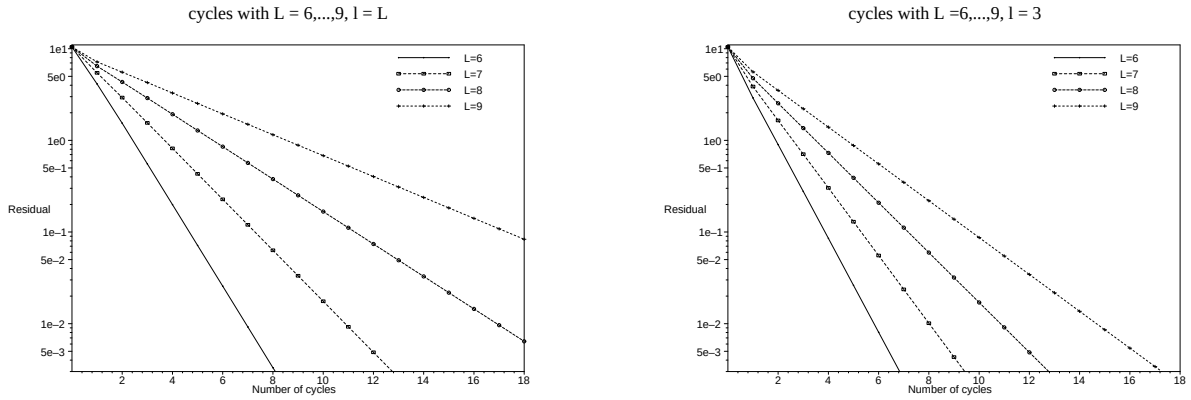

Figure 1. Left: Convergence of Algorithm (26) for the levels $L = 6, \ldots, 9$. Right: Convergence of Algorithm (27) for the levels $L = 6, \ldots, 9,\ \ell = 3$ with $\nu = 1$.

for $k = \ell$ **to** L
do for $i = 1$ **to** ν
do for all $|\mathbf{n}| = k$
$\boldsymbol{a}_L := \boldsymbol{a}_L + \boldsymbol{P}_{L,\mathbf{n}} \tilde{\boldsymbol{A}}_{\mathbf{n}}^{-1} \tilde{\boldsymbol{R}}_{\mathbf{n},L}(\boldsymbol{f}_L - \boldsymbol{A}_L \boldsymbol{a}_L)$ (27)
enddo
enddo
enddo

with a fixed lower level $\ell \geqslant d$ and a number ν of iterations per level.

Example: We apply the Algorithms (26) and (27) to the following 3D-problem. Solve (12) with the right-hand side

$$f(\mathbf{x}) = -3\pi^2 \Big(\prod_{i=1}^{3} \sin \pi x_i + 64 \prod_{i=1}^{3} \sin 8\pi x_i \Big)$$

and starting from the zero function $u_L^{(0)} \equiv 0$. We obtain the convergence behavior shown in Fig. 1. We see that we get better convergence if we include also lower levels (right). In both cases, the speed of convergence slows down with growing level. Approximately, the reduction factor gets worse with L^2, the square of the highest level.

If we apply Algorithms (26) and (27) to a similar 2D-examples, we obtain reduction factors that grow linearly with L. This is comparable to [32] where the iteration number for the Jacobi-preconditioned BiCGStab for similar 2D-problems grow at least linear in L. In a recent paper [14], a further reduction of the necessary BiCGStab iterations could be achieved by wavelet preconditioning, but also in this case the growth of the iteration numbers seems to be linear in L.

Discussion

Since the efficient $\mathcal{O}(N)$ evaluation of finite element stiffness matrices for variable coefficient equations on sparse grids ($d > 2$) is impossible, and FE residual computation requires unattractive modifications of the FE discretization [9] (cf. [40]), one might be tempted to use

finite differences (FD) instead. With the aim to construct an efficient multigrid procedure, we analyze the structure of the FD discrete Laplacian, and Galerkin relations are established. In particular, we study the relation between the FD operators on full and sparse grids. The rather complex relations (24) and (25) for this simple special case show that it is unlikely that – for the general case – a natural and efficient multilevel preconditioner can be found that shows spectral equivalence.

Thus, so far all attempts to construct $\mathcal{O}(N)$ solution methods for finite differences on ASG failed, and the structure of the discretization as analyzed in this paper gives little hope. It is our conclusion that, for the solution of the general equation (8) in three dimensions, it is probably better to abandon the generality of sparse grids and to adopt condition L. Then, representations on coarse grids are simply obtained by restriction from a unique finest grid, the application of FE gives no particular problems and the usual Galerkin relations hold. Optimal efficiency can be obtained by exploitation of all C^k-regularity and by the application of *hp*-adaptive methods [2] (with dyadic refinement) and multigrid semi-coarsening techniques.

References

[1] R. Balder, *Adaptive Verfahren für elliptische und parabolische Differentialgleichungen auf dünnen Gittern*, Ph.D. thesis, Technische Universität München, 1994.

[2] T. J. Barth and H. Deconinck, *High-Order Methods for Computational Physics*, vol. 9 of *Lecture Notes in Computational Science and Engineering*, Springer Verlag, Harlow, 1999.

[3] V. Barthelmann, E. Novak, and K. Ritter, *High dimensional polynomial interpolation on sparse grids*, Adv. Comput. Math., **12** (1999), pp. 273–288.

[4] H.-J. Bungartz and T. Dornseifer, *Sparse grids: Recent developments for elliptic partial differential equations*, in: *Multigrid Methods V* (W. Hackbusch and G. Wittum, eds.), Lecture Notes in Computational Science and Engineering 3, Springer, Berlin, 1998.

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

[6] H.-J. Bungartz, M. Griebel, D. Röschke, and C. Zenger, *Pointwise convergence of the combination technique for the Laplace equation*, East-West J. Numer. Math., **2** (1994), pp. 21–45.

[7] F.-J. Delvos and W. Schempp, *Boolean Methods in Interpolation and Approximation*, Pitman Research Notes in Mathematics Series 230, Longman Scientific & Technical, Harlow, 1989.

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

[9] T. Dornseifer, *Diskretisierung allgemeiner elliptischer Differentialgleichungen in krummlinigen Koordinatensystemen auf dünnen Gittern*, Ph.D. thesis, Technische Universität München, 1997.

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

[11] M. Griebel, *A parallelizable and vectorizable multilevel algorithm on sparse grids*, in: *Parallel algorithms for partial differential equations*, Notes Numer. Fluid Mech. 31, Vieweg, Wiesbaden, 1991, pp. 94–100.

[12] M. Griebel, *Multilevelmethoden als Iterationsverfahren über Erzeugendensystemen*, Teubner Skripten zur Numerik, Teubner, Stuttgart, 1994.

[13] M. Griebel, *Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences*, Computing, **61** (1998), pp. 151–180.

[14] M. Griebel and F. Koster, *Adaptive wavelet solvers for the unsteady incompressible Navier–Stokes equations*, in: *Advances in Mathematical Fluid Mechanics*, (J. Malek, J. Necas, and M. Rokyta, eds.), Lecture Notes of the Sixth International School Mathematical Theory in Fluid Mechanics, Paseky, Czech Republic, September 1999. Springer Verlag, 2000.

[15] M. Griebel and P. Oswald, *On the abstract theory of additive and multiplicative Schwarz algorithms*, Numer. Math., **70** (1995), pp. 163–180.

[16] M. Griebel and T. Schiekofer, *An adaptive sparse grid Navier–Stokes solver in 3D based on finite differences*, in: *Proceedings ENUMATH97* (H.G. Bock, G. Kanschat, R. Rannacher, F. Brezzi, R. Glowinski, Y.A. Kuznetsov, and J. Periaux, eds.), Heidelberg, 1999. World Scientific Publishing.

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

[18] P. W. Hemker, *Sparse–grid finite–volume multigrid for* $3D$*–problems*, Adv. Comput. Math., **4** (1995), pp. 83–110.

[19] P. W. Hemker and P. de Zeeuw, *BASIS3, a data structure for 3-dimensional sparse grids*, in: *Euler and Navier-Stokes Solvers Using Multidimensional Upwinds Schemes and Multigrid Acceleration* (H. Deconinck, ed.), Notes Numer. Fluid Mech. 57, Vieweg, Wiesbaden, 1997, pp. 443–484.

[20] P. W. Hemker and C. Pflaum, *Approximation on partially ordered sets of regular grids*, Appl. Numer. Math., **25** (1997), pp. 55–87.

[21] P. W. Hemker and F. Sprengel, *Experience with the solution of a finite difference discretization on sparse grids*, in: *Numerical Analysis and Its Applications* (L. Vulkov, J. Wasniewski, and P. Yalamov, eds.), Springer Verlag, Berlin, 2001, pp. 402–413, Proc. of the Conference "Numerical Analysis and its Applications", 2000, Rousse.

[22] N. M. Korobov, *Approximate calculation of multiple integrals with the aid of methods in the theory of numbers*, Dokl. Akad. Nauk SSSR, **115** (1957), pp. 1062–1065, in Russian.

[23] F. Kupka, *Sparse Grid Spectral Methods for the Numerical Solution of Partial Differential Equations with Periodic Boundary Conditions*, Ph.D. thesis, Universität Wien, 1997.

[24] W. Mulder, *A high-resolution Euler solver based on multigrid semi-coarsening, and defect correction*, Journal of Computational Physics, **100** (1992), pp. 91–104.

[25] W. A. Mulder, *A new multigrid approach to convection problems*, Journal of Computational Physics, (1989), pp. 303–323.

[26] J. Noordmans and P. W. Hemker, *Convergence results for 3D sparse grid approaches*, Numer. Linear Algebra Appl., **5** (1999), pp. 363–376.

[27] J. Noordmans and P. W. Hemker, *Application of an adaptive sparse-grid technique to a model singular perturbation problem*, Computing, **65** (2000) pp. 357–378.

[28] E. Novak and K. Ritter, *High dimensional integration of smooth functions over cubes*, Numer. Math., **75** (1996), pp. 79–98.

[29] P. Oswald, *Multilevel Finite Element Approximation. Theory and Applications*, Teubner Skripten zur Numerik, Teubner, Stuttgart, 1994.

[30] C. Pflaum, *Diskretisierung elliptischer Differentialgleichungen mit dünnen Gittern*, Ph.D. thesis, Technische Universität München, 1996.

[31] C. Pflaum, *A multilevel algorithm for the solution of second order elliptic differential equations on sparse grids*, Numer. Math., **79** (1998), pp. 141–155.

[32] T. Schiekofer, *Die Methode der Finiten Differenzen auf dünnen Gittern zur Lösung elliptischer und parabolischer partieller Differentialgleichungen*, Ph.D. thesis, Universität Bonn, 1998.

[33] T. Schiekofer and G. Zumbusch, *Software concepts of a sparse grid finite difference code*, in: *Concepts of Numerical Software* (W. Hackbusch and G. Wittum, eds.), Notes in Numerical Fluid Mechanics, Vieweg, Braunschweig, 1998, .

[34] S. A. Smolyak, *Quadrature and interpolation formulas of the classes W_s^a and E_s^a*, Dokl. Akad. Nauk SSSR, **131** (1960), pp. 1028–1031, in Russian, Engl. Transl.: Soviet Math. Dokl. 1:384–387, 1963.

[35] F. Sprengel, *Interpolation and Wavelet Decomposition of Multivariate Periodic Functions*, Ph.D. thesis, University of Rostock, 1997.

[36] F. Sprengel, *Interpolation and wavelets on sparse Gauß–Chebyshev grids*, in: *Multivariate Approximation, Recent Trends and Results* (W. Haußmann, K. Jetter, and M. Reimer, eds.), Mathematical Research, Vo. 101, Akademie–Verlag, Berlin, 1997, pp. 269–286.

[37] F. Sprengel, *Interpolation of functions from Besov–type spaces on Gauß–Chebyshev grids*, J. Complexity, **16** (2000), pp. 507-523.

[38] C. Teitzel, M. Hopf, R. Grosso, and T. Ertl, *Volume visualization on sparse grids*, Computing and Visualization in Science, **2** (1999), pp. 47–59.

[39] V. N. Temlyakov, *Approximation of Periodic Functions*, Nova Science, New York, 1993.

[40] E. E. P. Veldman and R. W. C. P. Verstappen, *Higher–order discretization methods for CFD*, Nieuw Archief voor Wiskunde, **Ser. IV Vol.17** (1999), pp. 195–204.

[41] G. W. Wasilkowski and H. Woźniakowski, *Explicit cost bounds of algorithms for multivariate tensor product problems*, J. Complexity, **11** (1995), pp. 1–56.

[42] C. Zenger, *Sparse grids*, in: *Parallel Algorithms for Partial Differential Equations* (W. Hackbusch, ed.), Notes on Numerical Fluid Mechanics 31, Vieweg, Braunschweig, 1991, pp. 297–301.

Received 20 Feb. 2001
Revised 20 Jul. 2001