Electronic Transactions on Numerical Analysis.
Volume 6, pp. 63-77, December 1997.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

A MULTIGRID ALGORITHM FOR HIGHER ORDER FINITE ELEMENTS ON SPARSE GRIDS*

HANS-JOACHIM BUNGARTZ†

Abstract. For most types of problems in numerical mathematics, efficient discretization techniques are of crucial importance. This holds for tasks like how to define sets of points to approximate, interpolate, or integrate certain classes of functions as accurate as possible as well as for the numerical solution of differential equations. Introduced by Zenger in 1990 and based on hierarchical tensor product approximation spaces, sparse grids have turned out to be a very efficient approach in order to improve the ratio of invested storage and computing time to the achieved accuracy for many problems in the areas mentioned above.

Concerning the sparse grid finite element discretization of elliptic partial differential equations, recently, the class of problems that can be tackled has been enlarged significantly. First, the tensor product approach led to the formulation of unidirectional algorithms which are essentially independent of the number d of dimensions. Second, techniques for the treatment of the general linear elliptic differential operator of second order have been developed, which, with the help of domain transformation, enable us to deal with more complicated geometries, too. Finally, the development of hierarchical polynomial bases of piecewise arbitrary degree p has opened the way to a further improvement of the order of approximation.

In this paper, we discuss the construction and the main properties of a class of hierarchical polynomial bases and present a symmetric and an asymmetric finite element method on sparse grids, using the hierarchical polynomial bases for both the approximation and the test spaces or for the approximation space only, resp., with standard piecewise multilinear hierarchical test functions. In both cases, the storage requirement at a grid point does not depend on the local polynomial degree p, and p and the resulting representations of the basis functions can be handled in an efficient and adaptive way. An advantage of the latter approach, however, is the fact that it allows the straightforward implementation of a multigrid solver for the resulting system which is discussed, too.

Key words. sparse grids, finite element method, higher order elements, multigrid methods.

AMS subject classifications. 35J05, 65N15, 65N30, 65N55.

1. Sparse Grids.

Though the hierarchical representation of functions for problems like interpolation or numerical quadrature has a long tradition that at least goes back to Archimedes, it was only a couple of years ago that a hierarchical approach was studied in detail for a PDE or, to be more precise, a finite element context [5, 34]. One of the main advantages of hierarchical bases compared with standard nodal point bases is probably the fact that its multilevel structure enables us to distinguish between high-level basis functions with a large support that usually (in the smooth case, at least) already contain a significant part of the information, and functions living on lower levels whose contribution to an interpolant or a finite element approximation is rather small. The decrease of the hierarchical coefficients from level to level can be used, of course, to control adaptive grid refinement, but, if it is combined with a tensor product approach for the higher dimensional case, it can be used for an a priori reduction of the number of grid points involved in the calculation, too.

In order to illustrate the transition from the well-known regular full grid $G_n^{(d)}$ with constant mesh width 2^{-n} for each coordinate direction to its corresponding sparse grid $\tilde{G}_n^{(d)}$, let us look at the subspace splitting that comes along with hierarchical bases on tensor product elements. Figure 1.1 shows the 1 D case of a piecewise linear hierarchical basis, Fig. 1.2 illustrates the tensor product construction of piecewise bilinear hierarchical basis functions on quadrilaterals. Note that we use recursive data structures like binary trees for the representation of our grid functions.

* Received May 31, 1997. Accepted August 29, 1997. Communicated by S. Parter.
† Institut für Informatik, Technische Universität München, D - 80290 München, Germany, (bungartz@informatik.tu-muenchen.de). This work was supported by the Bayerische Forschungsstiftung via FORTWIHR – The Bavarian Consortium for HPSC.

ETNA
Kent State University
etna@mcs.kent.edu

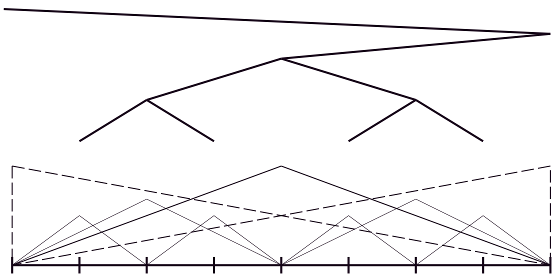


FIG. 1.1. *Piecewise linear hierarchical basis and corresponding binary tree*

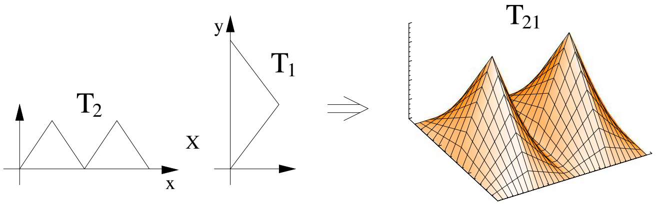


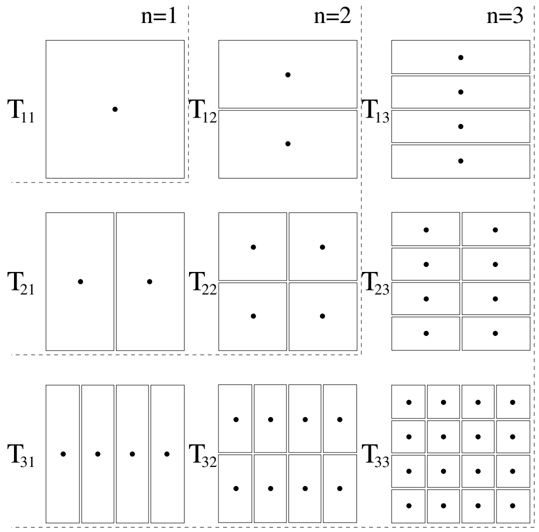
FIG. 1.2. *Tensor product approach for two piecewise bilinear hierarchical basis functions*

For $d = 2$, Fig. 1.3 shows a sector of the theoretically infinite scheme of subspaces. Here, a standard full grid $G_n^{(2)}$ with $(2^n - 1)^2$ inner grid points corresponds to a square sector of n^2 subspaces T_{i_1,i_2}, and T_{i_1,i_2} contains all basis functions with congruent supports of the same aspect ratio. Obviously, the dimension (i. e. the number of grid points) of all subspaces T_{i_1,i_2} with $i_1 + i_2 = c$ is just 2^{c-2}. Furthermore, for functions u continuous on the unit square $\bar{\mathbf{Q}}$, it has been shown that the contribution of all T_{i_1,i_2} with $i_1 + i_2 = c$ to the interpolant of u is of the same order $O(2^{-2c})$ with respect to the L_2- or L_∞-norm and $O(2^{-c})$ with regard to the H^1-norm, if $\frac{\partial^4 u}{\partial x_1^2 \partial x_2^2}$ and some lower mixed derivatives of u are continuous on $\bar{\mathbf{Q}}$ (see [7, 8, 30, 35]). For general d, analogous results have been shown for subspaces $T_{i_1,\dots,i_d}$ with $i_1 + \dots + i_d = c$, if $\frac{\partial^{2d} u}{\partial x_1^2 \dots \partial x_d^2}$ and some lower mixed derivatives of u are continuous on $\bar{\mathbf{Q}} = [0,1]^d$ (see [8, 30]). Due to these properties concerning cost (number of grid points) and benefit (order of approximation), it turns out to be more reasonable to deal with triangular schemes of subspaces as given in Fig. 1.4 instead of using square ones. The grids or patterns of grid points $\tilde{G}_n^{(d)}$ resulting from such triangular sections are called *sparse grids*. For a formal definition of sparse grids, see [8, 35].

If $S_n^{(d)}$ and $\tilde{S}_n^{(d)}$ denote the corresponding piecewise d-linear approximation spaces on the full grid $G_n^{(d)}$ and on the sparse grid $\tilde{G}_n^{(d)}$, respectively, we get the following representations that reflect the recursive and tensor product based approach:

(1.1)

ETNA
Kent State University
etna@mcs.kent.edu



33	23	33	13	33	23	33
32	22	32	12	32	22	32
33	23	33	13	33	23	33
31	21	31	11	31	21	31
33	23	33	13	33	23	33
32	22	32	12	32	22	32
33	23	33	13	33	23	33

FIG. 1.3. *Hierarchical subspace decomposition: subspace scheme for full grids $G_n^{(2)}, 1 \leq n \leq 3$, (left) and corresponding pattern of grid points for $n = 3$ (right). Each square on the left-hand side shows one subspace T_{i_1,i_2} and is divided into the (equally shaped) supports of this subspace's basis functions. The numbers on the right indicate the subspace index of the respective grid points.*

$$
\begin{aligned}
S_n^{(d)} &= \sum_{1 \leq i_1,\ldots,i_d \leq n} T_{i_1,\ldots,i_d} = S_n^{(1)} \otimes S_n^{(d-1)} = \sum_{i_1=1}^{n} T_{i_1} \otimes S_n^{(d-1)}, \\
\tilde{S}_n^{(d)} &= \sum_{i_1+\ldots+i_d \leq n+d-1} T_{i_1,\ldots,i_d} = \sum_{i_1=1}^{n} T_{i_1} \otimes \tilde{S}_{n+d-1-i_1}^{(d-1)}.
\end{aligned}
$$

This correlation of the approximation spaces clearly shows the main difference between standard full grids and sparse grids: For the sparse grid, the overall resolution limited by $n+d-1$ is defined as the sum of the resolutions i_j in all coordinate directions j, whereas for the full grid, the maximum resolution is defined for each direction separately. Thus, the smaller the mesh width h_1 is in the first dimension of a sparse grid $\tilde{G}_n^{(d)}$, the coarser the resolution will be in the remaining $d-1$ dimensions.

Besides the regular sparse grids that result from skipping certain subspaces according to Fig. 1.4, there is a very straightforward access to adaptive grid refinement. The hierarchical coefficient or *hierarchical surplus* itself can be used to indicate the smoothness of u at the corresponding grid point and, consequently, the necessity to refine the grid here. Figure 1.5 shows a regular 2 D sparse grid and an adaptively refined 3 D one with singularities at the re-entrant corner and along the three edges starting from there.

Concerning the most important properties of sparse grids, we have at least to look at the number of grid points involved and at the approximation accuracy in the piecewise multilinear case. For a detailed analysis, we once again refer to [8, 35]. For general d, the approach described above and illustrated in Fig. 1.4 leads to regular sparse grids with $O(N(\log_2(N))^{d-1})$ grid points, if $\frac{1}{N}$ denotes the smallest mesh width occurring. With some modification, sparse grids with $O(N)$ grid points can be defined, too. Concerning the approximation quality, the accuracy of the sparse grid interpolant is only slightly deteriorated from $O(N^{-2})$ to $O(N^{-2}(\log_2(N))^{d-1})$ with respect to the L_2- or L_∞-norm. With regard to the H^1-norm,

ETNA
Kent State University
etna@mcs.kent.edu

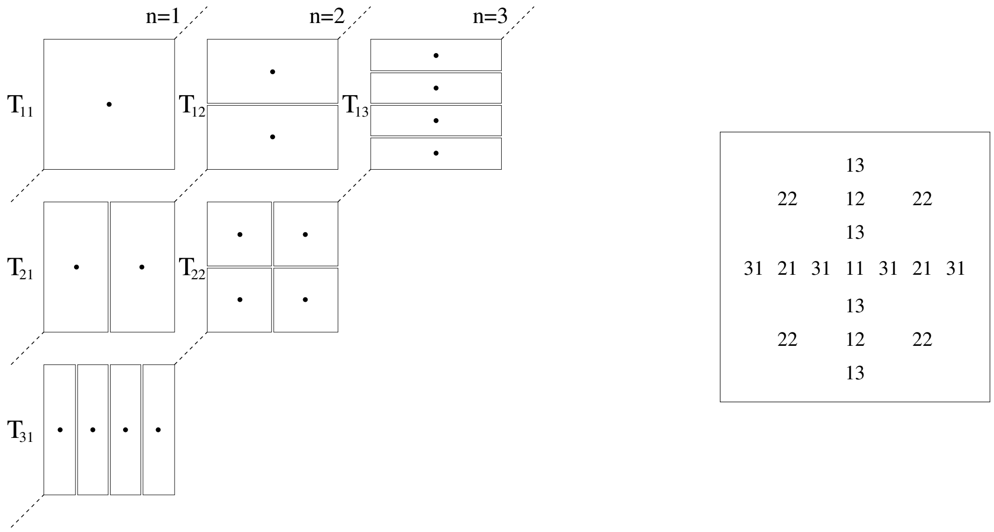


FIG. 1.4. *Hierarchical subspace decomposition: subspace scheme for sparse grids* $\tilde{G}_n^{(2)}, 1 \leq n \leq 3$, *(left) and corresponding pattern of grid points for* $n = 3$ *(right)*

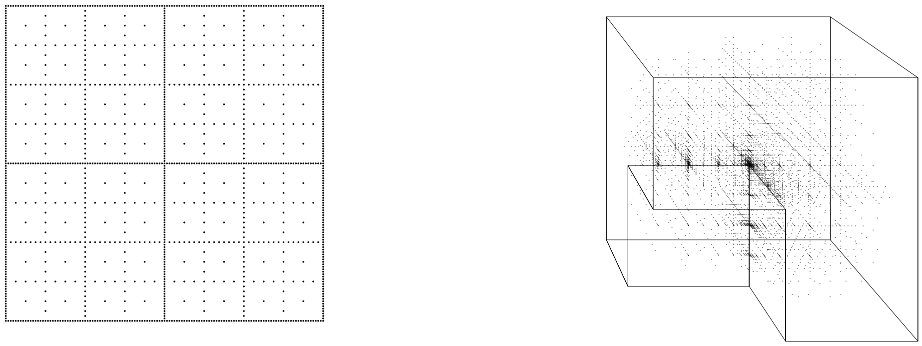


FIG. 1.5. *Sparse grids: regular example (left) and adaptive one (right)*

both the sparse grid interpolant and the finite element approximation to the solution of the given problem are of the order $O(N^{-1})$. Thus, we get a number of grid points that is nearly or even actually independent of d (a behaviour known in numerical quadrature from pseudorandom methods, e. g.), but we have to pay for it with only a logarithmic loss in accuracy. Therefore, sparse grids are a very promising approach for many tasks in numerical mathematics [4, 6, 9, 11, 12, 14, 16–21, 24–26, 32] and especially attractive for problems with a large parameter d.

It is important to note that the sparse grid scheme presented first in [35] for PDE applications has already been known for several years in interpolation, approximation, and recovery theory as well as in numerical quadrature. There, the idea of reduced tensor product spaces appears in the context of Smolyak quadrature rules [15, 31, 33] and under several other different names (hyperbolic crosses [1], Boolean methods [13]). In numerical quadrature, e. g., the intention is to choose sets or sequences of grid points of an optimal cost-profit-ratio, i. e. so-called low-discrepancy sequences or quadrature formulas, respectively [27]. As an example, Fig. 1.6 shows three patterns originating from 2 D Smolyak rules.

ETNA
Kent State University
etna@mcs.kent.edu

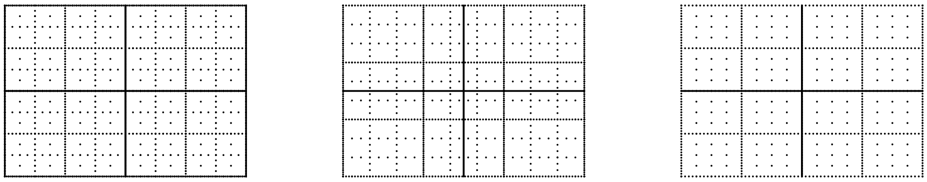


FIG. 1.6. *Smolyak quadrature rules: different grid patterns based on the trapezoidal rule as the 1 D algorithm*

2. The Unidirectional Scheme. In a finite element context, hierarchical bases usually lead to a certain fill-in of the stiffness matrix. This results from the fact that, due to the hierarchical relations, more basis functions than just neighbouring ones are connected with respect to the underlying bilinear form. Thus, often, more algorithmic skill has to be invested in order to ensure that the computational cost of an iterative scheme does not exceed a constant number of operations per step of the iteration and per grid point. Furthermore, we want to profit from the tensor product approach by keeping the step from 1 D to the general d-dimensional case as straightforward as possible. Therefore, the basic structure of the sparse grid finite element algorithms discussed here seems to be worth while studying.

The main underlying algorithmic principle of our method is the so-called *unidirectional* approach (cf. [4]), i. e. the fact that a d-dimensional problem is reduced to the simpler 1 D case via recursion. Thus, the parameter d can be handled as an input parameter of the code, and all algorithmic work can be done in just one dimension. To solve the arising linear systems, we use iterative schemes like the damped Jacobi iteration, a (preconditioned) conjugate gradient technique, or a multigrid method that will be discussed in detail in Sect. 4. The kernel of all those schemes is a routine to compute $S \cdot \vec{u}$ for the stiffness matrix S and arbitrary input vectors $\vec{u}$, and this product actually is the only part of the iteration where the hierarchical sparse grid approach comes in. Therefore, we need to have a closer look at S. In a tensor product approach, d-dimensional hierarchical basis functions $\varphi_i(x_1, \ldots, x_d)$ are defined as products of 1 D hierarchical basis functions $\varphi_{i,l}(x_l), 1 \le l \le d$:

$$\varphi_i(x_1, \ldots, x_d) \;:=\; \prod_{l=1}^{d} \varphi_{i,l}(x_l) \quad . \tag{2.1}$$

Thus, an entry $s_{i,j}$ of S for the Laplacian, e. g., is of the form

$$s_{i,j} \;=\; \sum_{k=1}^{d} \left(I_{i,j;k}^{\text{stiff}} \cdot \prod_{l \neq k} I_{i,j;l}^{\text{mass}} \right) \quad , \tag{2.2}$$

where

$$I_{i,j;k}^{\text{stiff}} \quad := \quad \int_{\mathbf{\Omega_{i,k}} \cap \mathbf{\Omega_{j,k}}} \frac{\partial \varphi_{i,k}(x_k)}{\partial x_k} \cdot \frac{\partial \varphi_{j,k}(x_k)}{\partial x_k} \, dx_k \quad , \tag{2.3}$$

$$I_{i,j;k}^{\text{mass}} \quad := \quad \int_{\mathbf{\Omega_{i,k}} \cap \mathbf{\Omega_{j,k}}} \varphi_{i,k}(x_k) \cdot \varphi_{j,k}(x_k) \, dx_k \quad , \tag{2.4}$$

and $\mathbf{\Omega_{i,k}} = \text{supp}(\varphi_{i,k}(x_k))$. Obviously, all $s_{i,j}$ are just sums of products of d 1 D integrals $I_{i,j;k}^{\text{stiff}}$ or $I_{i,j;k}^{\text{mass}}$, respectively, and all that has to be done from an algorithmic point of view is just to provide those integrals for all 1 D basis functions, i. e. for all i and j, and for all coordinate directions k.

ETNA
Kent State University
etna@mcs.kent.edu

Of course, for an efficient calculation of $S \cdot \vec{u}$, we must not compute the $s_{i,j}$ themselves, since we have lost some sparsity of S due to the use of hierarchical bases, but just the residuals or the sums $\sum_{j=1}^{N} s_{i,j}u_j$ for $1 \le i \le N$. This is done in a recursive way, such that we get all of those sums during a few passes through the data structure. In the 1 D case, we start with a vector $\vec{u}$ containing the actual solution u_i in all grid points i and make a copy $\vec{uu}$ of it. Then, with $\vec{u}$, a top-down-pass (called *down* in the following) through the data structure is done in hierarchical order, and with $\vec{uu}$, we make a bottom-up pass (called *up*). Note that, for the recursive extension, the separation of the two collection steps in *down* and *up* is important due to the hierarchical connections of the respective basis functions. After that, u_i contains the sum of all products $s_{i,j}u_j$ originating from grid points j hierarchically higher than i and from i itself, and uu_i contains all $s_{i,j}u_j$ from grid points j hierarchically lower than i. Finally, $\vec{u} := \vec{u} + \vec{uu}$ provides $\sum_{j=1}^{N} s_{i,j}u_j$ for each grid point i, and $\vec{u}$ now contains the product $S \cdot \vec{u}$. Thus, apart from the copy process, $S \cdot \vec{u}$ is calculated in place, and, therefore, we need only two variables per grid point or unknown, resp. The underlying 1 D algorithmic scheme of this process is shown in the upper part of Fig. 2.1. The recursive extension of the 1 D algorithmic principle to the general case is given by the lower part of Fig. 2.1. There, for $d > 1$, the grey boxes entitled *unidir(d-1)* have to be replaced by the $d-1$-dimensional scheme. Note that it is important to do the recursive calls *before* the down, but *after* the up step. Concerning the storage requirement, the influence of the parameter d is very small. Since we can handle the whole process on the stack, there exist only local copies of parts of the data structure which are dominated by the copy of the d-dimensional overall structure.

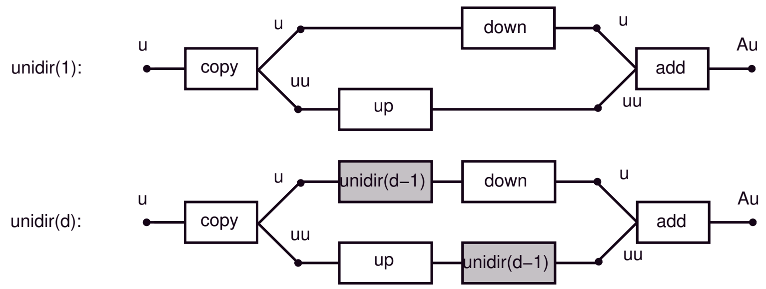


FIG. 2.1. *Scheme of the unidirectional algorithm: one-dimensional (top) and general d-dimensional case (bottom)*

In conclusion, we want to emphasize that the presented unidirectional algorithmic structure is independent of whether you work with standard full grids or with sparse grids, and that it does not depend on the type of hierarchical basis actually chosen. It is just based upon a hierarchical tensor product approach.

3. Hierarchical Tensor Product Bases of Higher Order. Up to now, piecewise multilinear and piecewise constant hierarchical bases have been the focus of interest in the sparse grid context. However, a first step towards higher order techniques on sparse grids was made in [32], where a piecewise bicubic hierarchical Hermite basis is used to solve the 2 D biharmonic equation $\Delta^2 u = 0$. Since the values of both the function and of its first derivatives have to be fixed, this approach leads to 2^d degrees of freedom per grid point in the general d-dimensional case. Furthermore, recently, concepts for hierarchical polynomial bases of piecewise arbitrary degree in each coordinate direction have been introduced [9, 10]. Such an approach allows us to combine the efficiency of sparse grids (which, in some sense, can be seen as a method of higher order themselves) and their intrinsic h-adaptivity (cf. Sect. 1) with the improved approximation qualities of higher order basis functions. Thus, in spite of a quite

ETNA
Kent State University
etna@mcs.kent.edu

different approach, there are close relations to the p- and h-p-versions of the finite element method [2, 3, 22, 23].

In [9], the hierarchical Lagrangian interpolation, a choice of hierarchical polynomials based on $\mathcal{C}^0$-elements with still one degree of freedom per grid point or element, resp., has been presented and studied in detail for $d = 2$ and $p = 2$. In [10], the principle of the hierarchical Lagrangian interpolation has been extended to arbitrary values of d and p. In the following, the characteristics of this approach shall be summarized. Since our tensor product approach provides us with d-dimensional functions if 1 D functions are defined, only this simpler case will be regarded in the explanations below.

Obviously, there is exactly one quadratic polynomial φ_i that fulfils $\varphi_i(x_i) = 1$ and $\varphi_i(x_i - h_i) = \varphi_i(x_i + h_i) = 0$ and that can be used, consequently, as a quadratic hierarchical basis function in grid point x_i with the support $[x_i - h_i, x_i + h_i]$. But, for some φ_i of a degree $p > 2$, additional degrees of freedom must be fixed. For the construction of a spline interpolant, e. g., those degrees of freedom are invested in more smoothness. In p-version-type algorithms, interpolation conditions at certain points within a typically large element (Gauß-Lobatto points, e. g.) are used to compensate the degrees of freedom. However, since we don't want to work with anything else but classical h-version-type $\mathcal{C}^0$-elements, we use *that* φ_i with support $[x_i - h_i, x_i + h_i]$ and $\varphi_i(x_i) = 1$ that is part of the polynomial of degree p with zeroes at the support's two boundary points and at the $p - 2$ next direct hierarchical ancestors x_k of x_i with respect to the hierarchical ordering of the grid points or associated subspaces (cf. Figs. 1.1, 1.3, and 1.4). Note that those ancestors x_k are situated *outside* the support; they are only used to construct φ_i. Consequently, we get just one degree of freedom in each grid point for arbitrary p, which makes it very comfortable to work with different polynomial degrees in different grid points or elements, respectively. On the other hand, of course, due to this reduced number of degrees of freedom, there is only a subspace of the space of all piecewise polynomials of degree p that can be represented.

Figure 3.1 illustrates the quartic case. Here, the actual grid point is $x_i = -0.75$, and, due to the hierarchical relations of the points given by the right-hand side of Fig. 3.1, the four ancestors used for the hierarchical interpolation are $-1, -0.5, 0$, and 1. The left-hand part of Fig. 3.1, finally, shows the whole resulting interpolant (dotted line) and the part of it that is afterwards used as the hierarchical basis function in x_i (solid line).

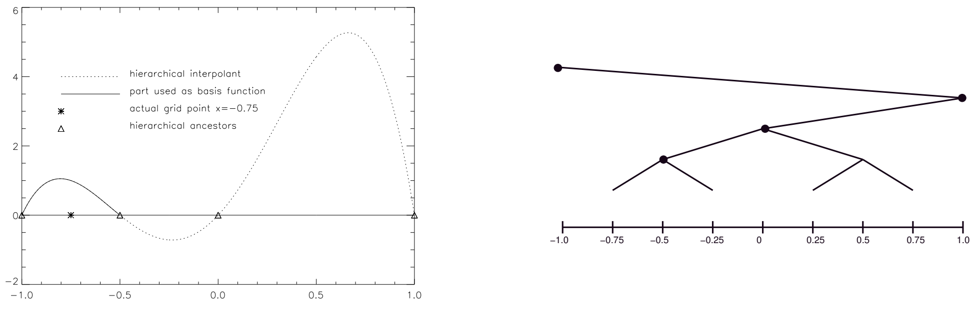


FIG. 3.1. *Construction of a hierarchical basis function of degree 4 by hierarchical Lagrangian interpolation (left) and corresponding hierarchical structure of the grid points (right)*

There are several interesting consequences of this construction. First, since the relative position of the different zeroes of φ_i depends on the hierarchical position of its corresponding grid point x_i, we get more than one different type of basis functions for $p > 2$. Actually, there are two different (but symmetric) basis functions of degree 3, four of degree 4, and, in general,

ETNA
Kent State University
etna@mcs.kent.edu

2^{p-2} of degree p. Figure 3.2 shows the basis functions for $p = 2$ and $p = 3$, Fig. 3.3 illustrates the situation for $p = 4$, where we get two pairs of symmetric polynomials.

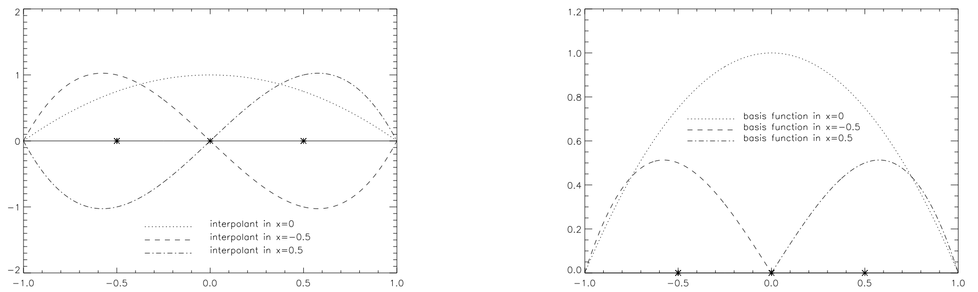


FIG. 3.2. *Hierarchical basis functions for $p = 2$ and $p = 3$ (different scaling for reasons of clarity): construction via hierarchical interpolation (left) and used restriction to the respective hierarchical support (right)*

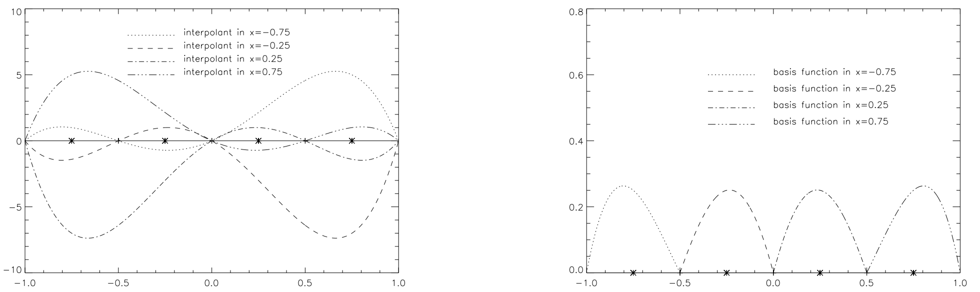


FIG. 3.3. *Hierarchical basis functions for $p = 4$ (different scaling for reasons of clarity): construction via hierarchical interpolation (left) and used restriction to the respective hierarchical support (right)*

Second, though our approach is based on a simple Lagrangian interpolation without any influence on the position of the respective nodal points, we get no numerical problems due to oscillations, since only a small and uncritical part of the resulting interpolant is taken into account. As it can be seen in Fig. 3.4, the shape of the different basis functions does not change that much even for larger p. I. e., there are only slight changes in the basis functions when we switch from p to $p + 1$, e. g. Therefore, we can expect an only moderate influence of p on the condition of the stiffness matrix (cf. the discussion and Fig. 4.6 in Sect. 4.3).

A third remark concerns the dependency of p in x_i on this grid point's hierarchical level. Since we need points outside φ_i's support if $p > 2$, it is clear that for the point on level 1, e. g. (i. e. $x_i = 0.5$ for $\mathbf{Q} =]0, 1[$), no basis function with $p > 2$ can be constructed. Thus, degree p can only occur starting from level $p - 1$ (i. e. $i_1 \geq p - 1$ for the subspace index i_1; cf. Sect. 1). Therefore, the exact representation of a polynomial u of degree p, e. g., at least needs the existence of level $p - 1$ and a total of $2^{p-1} + 1$ grid points in 1 D.

To study the efficiency of our approach, let us turn to some of its implementational properties. First of all, the same data structure can be used for arbitrary $p \geq 0$, because, in any case, just the hierarchical coefficient is stored in each grid point. Therefore, if M denotes the number of grid points (i. e. $M = O(N(\log_2(N))^{d-1})$), the representation of a function needs only M variables, and the unidirectional algorithm for the matrix-vector product discussed in Sect. 2 leads to a storage requirement bounded by $c \cdot M$ with a small constant c

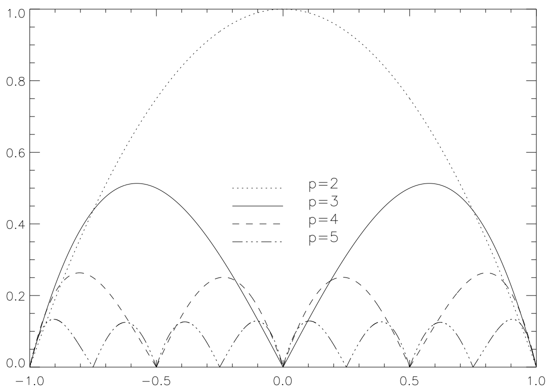


FIG. 3.4. *Resulting hierarchical basis for $G_4^{(1)}$ and $p_{\max} = 5$ (different scaling)*

neither depending on p nor on d. The algorithmic handling of the different polynomials and the residuals is done in 1 D procedures only and is organized on the stack. I. e., there are local vectors of the length $p+1$ which are passed on during the recursive *up* and *down* procedures described in Sect. 2. Thus, for a sparse grid $\tilde{G}_n^{(d)}$ of depth n and a fixed maximum degree p_{max}, there is only an additional storage of $O(n \cdot (p_{\text{max}}+1)) = O(\log_2(N) \cdot (p_{\text{max}}+1))$ for the stack. Since we use a classical Taylor representation $\vec{a} \in \mathbb{R}^{p_{\text{max}}+1}$ for all occurring basis functions and local interpolants $f(x)$,

$$f(x) = \sum_{k=0}^{p_{\text{max}}} a_k \cdot \frac{x^k}{k!} \quad , \tag{3.1}$$

an adaptive treatment of p can be easily achieved by omitting the leading coefficient a_p or by taking into account a new a_{p+1}, resp., if suitable criteria for such a process are developed. However, up to now, no experiments with an adaptive handling of p have been made. Finally, in order to avoid repeated calculations with our basis polynomials, we precompute the Taylor coefficients of each basis function φ_i and several integrals during a setup phase before the iteration. Due to the 2^{p-2} different basis polynomials of degree p, this leads to an additional storage requirement of $O(2^{p_{\text{max}}} \cdot (p_{\text{max}}+1))$. Since $p_{\text{max}} \leq n+1$ for $\tilde{G}_n^{(d)}$ due to the hierarchical Lagrangian approach and since $p_{\text{max}} \ll n$ for reasonable applications with bigger n, both terms $n \cdot (p_{\text{max}}+1)$ and $2^{p_{\text{max}}} \cdot (p_{\text{max}}+1)$ depending on p_{max} are significantly smaller than the $c \cdot M$ variables of the data structure itself. Consequently, there is only a slight influence of the polynomial degree on the overall storage requirement.

Concerning the number of arithmetic operations, the only part of the algorithm where p is important is the updating and passing of the vectors $\vec{a}$ of length $p+1$ to grid points on the next lower or higher level, resp. For the *down* process of Sect. 2, this is equivalent to a multiplication

$$\vec{a}^{\text{(son)}} := T \cdot D \cdot \vec{a}^{\text{(father)}} \quad , \tag{3.2}$$

for the *up* process, the updating and passing corresponds to

$$\vec{a}^{\text{(father)}} := D \cdot T^T \cdot \vec{a}^{\text{(son)}} \quad , \tag{3.3}$$

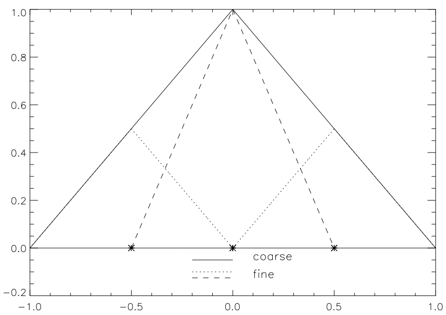
where $D \in \mathbb{R}^{(p+1)\times(p+1)}$ represents a diagonal scaling, and $T \in \mathbb{R}^{(p+1)\times(p+1)}$ is an upper triangular Toeplitz matrix. Note that both D and T are constant matrices that do neither depend on the actual grid point nor on the hierarchical level. From (3.2) and (3.3), it follows that the number of arithmetic operations is of the order $O(p^2)$ in each grid point. Thus, if we work with a fixed maximum degree p_{max}, we get a total of $O(M \cdot p_{\text{max}}^2) = O(N(\log_2(N))^{d-1} \cdot p_{\text{max}}^2)$ operations for the product of the stiffness matrix S with a given solution $\vec{u}$. However, if on each level l degree $p := l + 1$ is applied with no limit for p, the result is a total of $O(M \cdot (\log_2(N))^2) = O(N(\log_2(N))^{d+1})$ operations.

Finally, after studying the storage requirement and the computational cost of our algorithm, we must have a look at the quality of the underlying sparse grid finite element approximation. Analogously to the quadratic case, where for $\tilde{G}_n^{(d)}$ an interpolation accuracy of $O(N^{-3}(\log_2(N))^{d-1}) = O(h^3|\log_2(h)|^{d-1})$ with respect to the L_2- and the L_∞-norm and an approximation accuracy of $O(N^{-2}) = O(h^2)$ with respect to the H^1-norm have been proved, if $\frac{\partial^{3d} u}{\partial x_1^3 \ldots \partial x_d^3}$ and lower mixed derivatives of u are continuous on $\bar{\mathbf{Q}}$ (cf. [9]), we can show orders of $O(h^{p+1}|\log_2(h)|^{d-1})$ or $O(h^p)$, resp., for the general (non-p-adaptive) situation of a degree p. The proof follows [9], but gets a bit more technical due to the increasing number of types of basis functions for larger p. The smoothness requirements have to be increased in a corresponding way, and we need now continuous mixed derivatives up to $\frac{\partial^{(p+1)d} u}{\partial x_1^{p+1} \ldots \partial x_d^{p+1}}$. Note that, of course, the intrinsic h-adaptivity of sparse grids is not influenced or reduced by our higher order approach.

4. Finite Element Algorithms and Multigrid Solution. In this section, first, two types of sparse grid finite element methods of higher order are presented: a symmetric (Ritz-Galerkin) one with higher order functions in both the approximation and the test spaces and an asymmetric (Petrov-Galerkin) one with higher order functions only in the approximation spaces. Based on the second algorithm, a simple multigrid scheme for the solution of the resulting system is introduced. Finally, some first numerical results for a simple test problem are given.

4.1. Different Choices of the Test Space. Up to now, without explicitly mentioning it, we have started out from a standard Ritz-Galerkin approach of a symmetric choice of the approximation and test spaces. Indeed, in our first experiments with a preconditioned conjugate gradient method or a damped Jacobi iteration as a solver, the polynomial bases from Sect. 3 were used for both spaces. However, for efficient multigrid algorithms, straightforward grid transfer operators between nodal point bases of different levels are essential. Here, things become more complicated with our polynomial bases. In the left part of Fig. 4.1, the simple case of the grid transfer with piecewise linear basis functions is illustrated, where a linear combination of three neighbouring fine grid functions based on the $(\frac{1}{2}, 1, \frac{1}{2})$ stencil leads to the corresponding coarse grid one. Problems arise already in the quadratic case shown on the right-hand side of Fig. 4.1, where no linear combination of three equally shaped fine grid functions can result in the corresponding coarse grid one. Therefore, a totally new type of fine grid function (dashed line) with a doubled support has to be defined in the coarse grid points. For $p > 2$, this situation becomes even worse due to the increasing number of different basis polynomials.

Switching to standard piecewise linear hierarchical *test* functions turns out to be a remedy for these problems. Now, the residuals can be passed from level to level according to the left part of Fig. 4.1. At this point, it is important to remember that the different polynomial degrees for the test and approximation spaces cause no problems at all, since we always have just one degree of freedom per grid point – for arbitrary values of p. However, since this Petrov-Galerkin approach leads to asymmetric stiffness matrices, iterative solution strategies that need symmetry like the conjugate gradient method can not be used any longer.



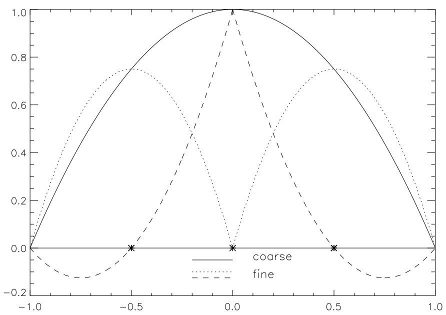


FIG. 4.1. *Change of level: linear (left) and quadratic case (right)*

Concerning the accuracy of the solutions, both variants turn out to be of the same quality. For the situation of the 2 D Laplace model problem from Sect. 4.3, Fig. 4.2 compares the maximum errors of the higher order test functions (solid lines) and the piecewise linear ones (dashed lines) for different values of n and $p \in \{2, 4, 6\}$.

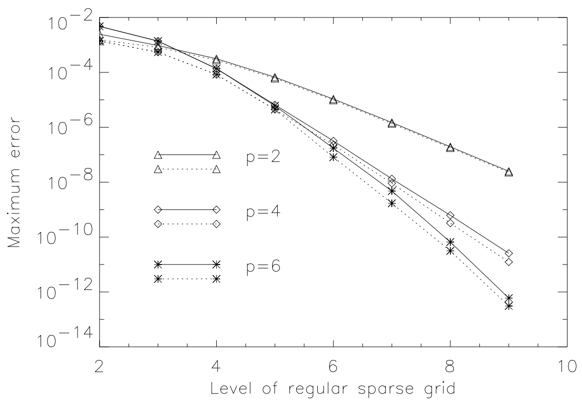


FIG. 4.2. *Comparison of the achieved accuracy: piecewise linear (solid lines) and higher order test functions (dashed lines)*

4.2. A Multigrid Scheme. For a sparse grid $\tilde{G}_n^{(d)}$, there are no natural finer or coarser grids in a standard multigrid sense like $G_{n-1}^{(d)}$ and $G_{n+1}^{(d)}$ are for the full grid $G_n^{(d)}$. However, due to the tensor product approach, we can refine or coarsen in each coordinate direction separately. Therefore, each full grid $G_{i_1,\ldots,i_d}$ with mesh widths $h_l = 2^{-i_l}, 1 \leq l \leq d$, consisting of the grid points from all $T_{j_1,\ldots,j_d}$ with $j_l \leq i_l, 1 \leq l \leq d$ (cf. Figs. 1.3 and 1.4 for the 2 D case), and being contained in $\tilde{G}_n^{(d)}$ can be seen as a coarse grid with respect to $\tilde{G}_n^{(d)}$. This suggests a choice of coarse grids closely related to the well-known semi-coarsening schemes [28, 29]. Thus, preceded by one damped Jacobi smoothing step on $\tilde{G}_n^{(d)}$, a sequence of coarse grid corrections of again one damped Jacobi step on the respective coarse grid is executed for various full (semi-) coarsened grids $G_{i_1,\ldots,i_d}$. Figure 4.3 illustrates two possible variants of this coarse grid correction scheme for 2 D problems. Note that, in contrast to Figs. 1.3 and 1.4, each small square in Fig. 4.3 stands for a standard full grid G_{i_1,i_2}. Now, a coarse

ETNA
Kent State University
etna@mcs.kent.edu

grid correction step is made on all G_{i_1,i_2} represented in grey colour. Thus, on the left-hand side, each full grid contained in $\tilde{G}_n^{(2)}$ is visited, whereas on the right-hand side, the coarse grid correction is restricted to those G_{i_1,i_2} with $i_1 + i_2 = n + 1$ or $i_1 = i_2 \leq n/2$. Since it is known from [16, 17, 21] that the reduced coarse grid selection of the right-hand side of Fig. 4.3 is sufficient in order to achieve multigrid efficiency, this variant has been chosen for the numerical experiments.

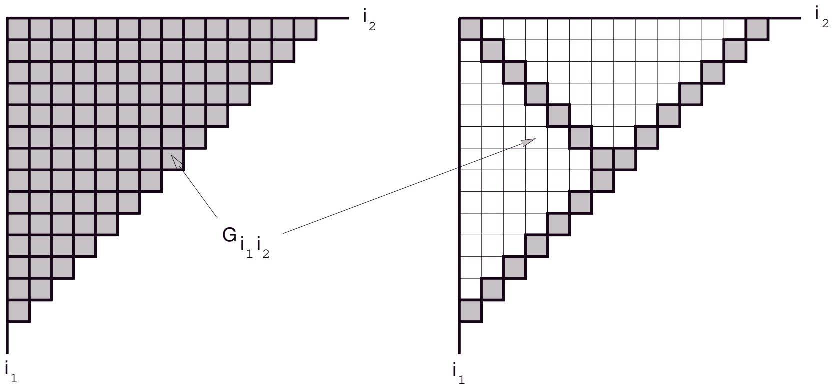


FIG. 4.3. *Sparse grid multigrid scheme: coarse grid correction on all contained full grids G_{i_1,i_2} (left) and reduced choice of coarse subgrids (right)*

Figure 4.4 gives some results for the multigrid convergence of the algorithm described above, applied to 1 D (left) and 2 D (right) Laplace model problems. Here, p denotes the maximum polynomial degree of the hierarchical basis functions used, and ρ is the average factor by which the maximum error is reduced after each kind of V-cycle described above for sparse grids $\tilde{G}_n^{(d)}, 4 \leq n \leq 9$. Note that no kind of optimization has been made concerning the number of smoothing steps on the coarse and the fine grids or concerning the damping factors. That is why the convergence rates are far from being optimal for such simple problems. However, here, it is just the convergence behaviour's independence of p that is in the centre of interest.

p	2	3	4	5	6
ρ	0.18	0.19	0.18	0.18	0.18

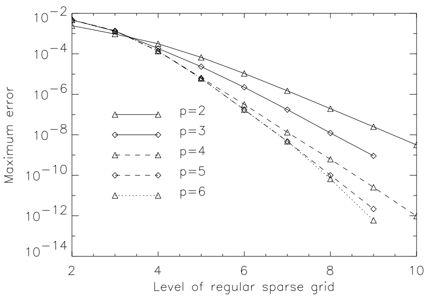
p	2	3	4	5	6
ρ	0.50	0.52	0.51	0.52	0.53

FIG. 4.4. *Average multigrid convergence rates for 1 D (left) and 2 D (right) Laplace problems*

4.3. A Numerical Example. As a simple example and model problem, we study the Laplace equation in two dimensions on the unit square **Q** with Dirichlet boundary conditions and the smooth solution

$$u(x_1, x_2) \; := \; \sin(\pi x_1) \sinh(\pi x_2) / \sinh(\pi). \tag{4.1}$$

To solve the resulting linear system, the multigrid algorithm based on a damped Jacobi smoothing on $\tilde{G}_n^{(2)}$ and successive coarse grid correction steps on a reduced choice of full subgrids (cf. the right-hand side of Fig. 4.3) presented in the previous section has been used. Figure 4.5 shows the behaviour of the L_∞-norm $\|\tilde{e}_n^{(2)}\|_\infty$ of the error on the regular sparse grids $\tilde{G}_n^{(2)}$ and the factors ρ_n of error reduction, $\rho_n := \|\tilde{e}_{n+1}^{(2)}\|_\infty / \|\tilde{e}_n^{(2)}\|_\infty$, for increasing n and $2 \leq p \leq 6$. For degree p, the approximation quality turns out to be a little bit worse than the order $O(h^{p+1}) = O(2^{-n(p+1)})$ of the full grid case, which is due to the additional logarithmic factor typical for sparse grids. Nevertheless, the higher order approximation properties can be seen clearly.



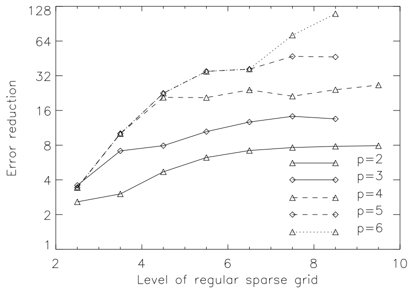


FIG. 4.5. *L_∞-error on $\tilde{G}_n^{(2)}$ (left) and factors ρ_n of error reduction (right) for various n and p*

Figure 4.6 shows the spectral condition number of the diagonally preconditioned stiffness matrix for different regular sparse grids $\tilde{G}_n^{(2)}$ and $2 \leq p \leq 6$. In comparison with the behaviour known from hierarchical polynomials in a p- or h-p-version context (cf. [36], e. g.), the influence of the polynomial degree on the condition number turns out to be quite moderate. This was to be expected, because, due to the fact that we only use a small part of the respective hierarchical interpolants as our actual basis function, the shape of the basis functions does not differ that much (cf. Fig. 3.4, e. g.).

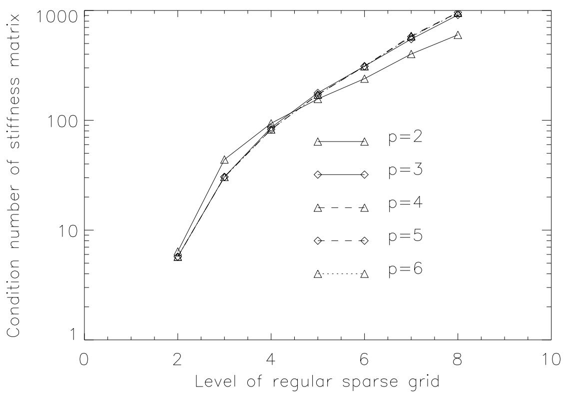


FIG. 4.6. *Condition of the (diagonally preconditioned) stiffness matrix for sparse grids $G_n^{(2)}$*

5. Concluding Remarks. In this paper, two unidirectional algorithms for higher order finite element discretizations on sparse grids have been discussed. Both are based upon a hierarchical Lagrangian construction of d-dimensional hierarchical tensor product bases of piecewise arbitrary polynomial degree p, and both open the way to some kind of p-adaptivity, if suitable criteria for the adaptive handling of p are developed. Due to the simple and cheap construction, handling, and storage of the resulting polynomial bases, this approach provides a very promising access to finite element methods of higher order on sparse grids. Furthermore, starting from the asymmetric method with different test and approximation spaces, a simple multigrid scheme has been developed that allows the fast solution of the arising linear systems. Thus, we have now an efficient sparse grid implementation of our higher order method. The next step will be to extend this approach to more general differential operators and domains as presented in [10] for the piecewise linear case.

Acknowledgements. I am indebted to Christoph Zenger and to Stefan Zimmer for many fruitful discussions and valuable suggestions.

REFERENCES

[1] K. I. BABENKO, *Approximation by trigonometric polynomials in a certain class of periodic functions of several variables*, Dokl. Akad. Nauk SSSR, 132 (1960), pp. 982–985 (Russian); 672–675 (English translation).

[2] I. BABUŠKA AND M. SURI, *The p- and h-p-versions of the finite element method: An overview*, Comput. Methods Appl. Mech. Engrg., 80 (1990), pp. 5–26.

[3] I. BABUŠKA, B. A. SZABÓ, AND I. N. KATZ, *The p-version of the finite element method*, SIAM J. Numer. Anal., 18 (1981), pp. 515–545.

[4] R. BALDER AND C. ZENGER, *The solution of multidimensional real Helmholtz equations on sparse grids*, SIAM J. Sci. Comp. 17 (1996).

[5] R. E. BANK, T. DUPONT, AND H. YSERENTANT, *The hierarchical basis multigrid method*, Numer. Math., 52 (1988), pp. 427–458.

[6] T. BONK, *A new algorithm for multi-dimensional adaptive numerical quadrature*, in Proceedings of the 9^{th} GAMM-Seminar, Kiel, January 1993, W. Hackbusch, ed., Vieweg, Braunschweig, 1994.

[7] H.-J. BUNGARTZ, *An adaptive Poisson solver using hierarchical bases and sparse grids*, in Iterative Methods in Linear Algebra: Proceedings of the IMACS International Symposium, Brussels, April 1991, P. de Groen and R. Beauwens, eds., Elsevier, Amsterdam, 1992, pp. 293–310.

[8] ———, *Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung*, Dissertation, Institut für Informatik, TU München, 1992.

[9] ———, *Concepts for higher order finite elements on sparse grids*, in Houston Journal of Mathematics: Proceedings of the 3^{rd} Int. Conf. on Spectral and High Order Methods, June 1995, Houston, A. V. Ilin and L. R. Scott, eds., 1996, pp. 159–170.

[10] H.-J. BUNGARTZ AND T. DORNSEIFER, *Sparse grids: Recent developments for elliptic partial differential equations*, SFB Report 342/02/97 A, Institut für Informatik, TU München, 1997. Submitted to Proc. EMG '96.

[11] H.-J. BUNGARTZ, M. GRIEBEL, D. RÖSCHKE, AND C. ZENGER, *Pointwise convergence of the combination technique for Laplace's equation*, East-West J. Numer. Math., 2 (1994), pp. 21–45.

[12] H.-J. BUNGARTZ, M. GRIEBEL, AND U. RÜDE, *Extrapolation, combination, and sparse grid techniques for elliptic boundary value problems*, Comput. Methods Appl. Mech. Engrg., 116 (1994), pp. 243–252.

[13] F.-J. DELVOS, *d-Variate Boolean interpolation*, J. Approx. Theory, 34 (1982), pp. 99–114.

[14] T. DORNSEIFER AND C. PFLAUM, *Discretization of elliptic partial differential equations on curvilinear bounded domains with sparse grids*, Computing, 56 (1996), pp. 197–214.

[15] K. FRANK AND S. HEINRICH, *Computing discrepancies of Smolyak quadrature rules*, J. Complexity, 12 (1996), pp. 287–314.

[16] M. GRIEBEL, *Parallel multigrid methods on sparse grids*, in Multigrid Methods III: Proceedings of the 3^{rd} European Conference on Multigrid Methods, Bonn, October 1990, W. Hackbusch and U. Trottenberg, eds., Int. Ser. Num. Math. 98, Birkhäuser, Basel, 1991, pp. 211–221.

[17] ———, *A parallelizable and vectorizable multi-level algorithm on sparse grids*, in Parallel Algorithms for Partial Differential Equations: Proceedings of the 6^{th} GAMM-Seminar, Kiel, January 1990, Notes on Numerical Fluid Mechanics 31, W. Hackbusch, ed., Vieweg, Braunschweig, 1991, pp. 94–100.

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

[19] M. GRIEBEL AND P. OSWALD, *On additive Schwarz preconditioners for sparse grid discretizations*, Numer. Math., 66 (1994), pp. 449–463.

[20] ———, *On the abstract theory of additive and multiplicative Schwarz algorithms*, Numer. Math., 70 (1995), pp. 163–180.

[21] ———, *Tensor product type subspace splittings and multilevel iterative methods for anisotropic problems*, Adv. in Comput. Math., 4 (1995), pp. 171–206. Also: Institut für Informatik der TU München, SFB-Report 342/15/94.

[22] B. GUO AND I. BABUŠKA, *The h-p-version of the finite element method (Part 1: The basic approximation results)*, Comput. Mech., 1 (1986), pp. 21–41.

[23] ———, *The h-p-version of the finite element method (Part 2: General results and applications)*, Comput.

ETNA
Kent State University
etna@mcs.kent.edu

Mech., 1 (1986), pp. 203–220.

[24] K. HALLATSCHEK, *Fouriertransformation auf dünnen Gittern mit hierarchischen Basen*, Numer. Math., 63 (1992), pp. 83–97.

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

[26] S. HILGENFELDT, *Numerische Lösung der stationären Schrödingergleichung mit Finite-Element-Methoden auf dünnen Gittern*, Diplomarbeit, Institut für Informatik, TU München, 1994.

[27] A. R. KROMMER AND C. W. UEBERHUBER, *Numerical Integration on Advanced Computer Systems*, vol. 848 of LNCS, Springer-Verlag, Berlin Heidelberg, 1994.

[28] W. A. MULDER, *A new multigrid approach to convection problems*, J. Comput. Phys., 83 (1989), pp. 303–323.

[29] N. H. NAIK AND J. R. VAN ROSENDALE, *The improved robustness of multigrid elliptic solvers based on multiple semicoarsened grids*, SIAM J. Numer. Anal., 30 (1993), pp. 215–229.

[30] C. PFLAUM, *Diskretisierung elliptischer Differentialgleichungen mit dünnen Gittern*, Dissertation, Institut für Informatik, TU München, 1996.

[31] S. A. SMOLYAK, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Dokl. Akad. Nauk SSSR, 148 (1963), pp. 1042–1045 (Russian); 240–243 (English translation).

[32] T. STÖRTKUHL, *Ein numerisches, adaptives Verfahren zur Lösung der biharmonischen Gleichung auf dünnen Gittern*, Dissertation, Institut für Informatik, TU München, 1994.

[33] V. N. TEMLYAKOV, *On approximate recovery of functions with bounded mixed derivative*, J. Complexity, 9 (1993), pp. 41–59.

[34] H. YSERENTANT, *On the multilevel splitting of finite element spaces*, Numer. Math., 49 (1986), pp. 379–412.

[35] C. ZENGER, *Sparse grids*, in Parallel Algorithms for Partial Differential Equations: Proceedings of the 6^{th} GAMM-Seminar, Kiel, January 1990, Notes on Numerical Fluid Mechanics 31, W. Hackbusch, ed., Vieweg, Braunschweig, 1991.

[36] G. W. ZUMBUSCH, *Simultaneous h-p Adaption in Multilevel Finite Elements*, Shaker Verlag, Aachen, 1996.