IMPROVED MULTILEVEL ALGORITHMS FOR FULL AND SPARSE GRID PROBLEMS

M. GRIEBEL , C. ZENGER AND S. ZIMMER *

Summary. We present grid-oriented and newly developed point-oriented robust multilevel methods for full and sparse grid discretizations. Especially the point-oriented multilevel methods are very well suited for parallelization and behave fairly robust for anisotropic model problems. They can be generalized easily to domain-oriented multilevel methods with the same properties. Beside multiplicative variants, we consider also their additive counterparts.

We report the results of numerical experiments regarding the reduction rates of these new algorithms.

Key words. Anisotropic problems, block-Gauss-Seidel iteration, BPX-preconditioner, conjugate gradients, Gauss-Seidel iteration, hierarchical basis, multigrid methods, multilevel methods, partial differential equations, point-oriented methods, semidefinite system, sparse grids, sparse grid preconditioner.

AMS(MOS) subject classifications. 65F10, 65N20, 65N30

1. Introduction.

Recently, so called sparse grid techniques for the solution of elliptic PDE's have been developed. There, in contrast to standard methods, substantially less grid points are needed in the discretization process. The obtained accuracy, however, is nearly as good as in the full grid case.

Using certain product type hierarchical basis functions (HB) for the associated sparse grid finite element space, the Galerkin process leads for linear PDE's to a linear system that has to be solved efficiently. Applying for example conjugate gradient (CG) or Gauss-Seidel (GS) iterations results basically in the HB-preconditioned CG or the HB-multigrid method. Crucial for the convergence, however, is the condition number of the sparse grid system that was shown in [Oswald, 1992] to behave at least like $O(2^{k/2})$ if k denotes the number of levels employed. Thus, CG or GS-iterations are quite slow and the resulting convergence rates depend exponentially on k.

However, for simple model problems, special multilevel type methods for the sparse grid system had been developed (see [Griebel, 1991b], [Griebel, 1991c]) that show a convergence rate independent of the grid size. But a general, systematic approach for the construction of multilevel type algorithms for sparse grid problems was missing.

In [Griebel, 1991a], a new concept for the development of multigrid and BPX-like multilevel algorithms for standard full grid problems has been presented. There, instead of a basis approach on the finest grid and the acceleration of the basic iteration by a MG-coarse grid correction or a BPX type preconditioner, a generating system was used to allow a non-unique level-wise decomposed representation of the solution. The degrees of freedom are associated to the nodal basis functions of all levels under consideration. Furthermore, the grids of the different levels are connected by standard refinement techniques. With this non-unique multilevel decomposed representation of a function,

* Institut für Informatik, Technische Universität München, Arcisstraße 21, D-8000 München 2

the Galerkin approach leads to a semidefinite linear system with unknowns on all levels. Its solution is non-unique but equivalent to the unique solution of the standard problem on the finest grid.

Furthermore, it was shown that traditional iterative methods for the semidefinite system are equivalent to modern elaborated multilevel methods that exhibit optimal convergence properties. The conjugate gradient method (with appropriate diagonal scaling) for the semidefinite system is equivalent to the BPX-conjugate gradient method for the fine grid system. Gauss-Seidel type iterations for the semidefinite system are equivalent to certain multigrid methods. For details, see [Griebel, 1991a].

Now, we generalize this concept beyond the standard refinement case and adopt it to the sparse grid application. This allows us to construct new multilevel algorithms that work for standard grid problems and sparse grid problems.

First, we introduce an enlarged generating system for the non-unique representation of functions. This generating system contains the nodal basis functions of all possible levels of discretization. There, in contrast to [Griebel, 1992], not only the coarser grids are contained that result from standard coarsening, but additionally all coarser grids are contained that can be gained by semi-coarsening with respect to each possible coordinate direction. In this sense, our approach is related to the work of Hackbusch [Hackbusch, 1987, Hackbusch, 1989] and the ideas of Mulder [Mulder, 1988] or Naik and van Rosendale [Naik & van Rosendale, 1991].

Then, the Galerkin approach results in a semidefinite linear system with non-unique solution. Here, we consider the full grid case and the sparse grid case. The arising system can be solved efficiently by Gauss-Seidel iterations or (after appropriate diagonal scaling) by the conjugate gradient method. These methods are level-oriented and can be considered as a level-block technique. An outer iteration switches from level to level and an inner iteration operates on the specific grid. In this way, we obtain robust multigrid and BPX-like multilevel methods for the full grid problem that show convergence properties similar to the respective multilevel methods based on standard coarsening only. Additionally, multigrid and multilevel algorithms can be obtained easily for so called sparse grid problems.

Furthermore, we consider the semidefinite system from a different point of view. We group all unknowns together that are associated to the same grid point. This results in point-oriented methods and can be considered as a point-block technique. Now, an outer iteration switches from grid point to grid point. The local system that belongs to all basis functions of different levels centered in the same grid point can be solved either directly or by an inner iteration that runs over all levels that are associated to the grid point under consideration. Furthermore, grid points can be grouped together to form subdomains. In this sense, we get some sort of simple domain decomposition method that exhibits MG type convergence properties.

Throughout this paper, we consider the full and the sparse grid case. It turns out that we obtain convergence rates and efficiencies for the new grid- and point-oriented approaches that are similar to the standard coarsening case as reported in [Griebel, 1992]. However, due to the underlying semi-coarsening aspect with respect to each possible coordinate direction, we gain more robust algorithms, especially in the case of anisotropic operators.

Since the point-oriented method allows directly an interpretation in terms of domain decomposition, its parallelization is straightforward. In contrast to the parallelization of a multilevel method where communication between subdomains has to take place on all levels to maintain good convergence rates, our point-block approach needs substantially less communication, i. e. only one communication per iteration step between different subdomains. This also results in a simpler program structure for complicated domains. In this sense, our new method is superior to other parallel multigrid and multilevel methods and we believe that its future will be bright.

We report the results of numerical experiments regarding the convergence rates of these new algorithms.

2. The semidefinite system for the multiple refined grid case. Consider a partial differential equation in two dimensions with a linear, second order operator in the domain $\Omega = (0,1)^2$,

$$Lu = f \text{ in } \Omega, \tag{1}$$

with appropriate boundary conditions and solution u. For reasons of simplicity we restrict ourselves to homogeneous Dirichlet boundary conditions.

Given an appropriate function space V, the problem can be expressed equivalently in its variational form: Find a function $u \in V$ with

$$a(u,v) = (f,v) \; \forall \; v \in V. \tag{2}$$

(In the case of homogeneous Dirichlet boundary conditions, V is the Sobolev space $H_0^1(\Omega)$.) Here, $a : V \times V \rightarrow V$ is a bounded, positive definite, symmetric bilinear form, and $(.,.)$ is the usual linear form for the right hand side. Let $\|.\|_a := \sqrt{a(.,.)}$ denote the induced energy norm. We assume, that V is complete with respect to $\|.\|_a$, which is true if $a(.,.)$ is H_0^1-elliptic. The Lax-Milgram lemma guarantees the existence and uniqueness of the solution of (2). If we consider directly the functional $E(u) = 1/2$ $a(u,u) - (f,u)$, the problem can be stated alternatively as minimization of $E(u)$ in V.

2.1. Splitting of finite element spaces by multiple refined grids. Assume now that for the discretization of the problem a tableau of grids $\Omega_{m,n}$

$$\begin{array}{ccccccc}
\Omega_{1,1} & \Omega_{1,2} & \Omega_{1,3} & \Omega_{1,4} & \dots & \Omega_{1,k} & \dots \\
\Omega_{2,1} & \Omega_{2,2} & \Omega_{2,3} & \Omega_{2,4} & \dots & \Omega_{2,k} & \dots \\
\Omega_{3,1} & \Omega_{3,2} & \Omega_{3,3} & \Omega_{3,4} & \dots & \Omega_{3,k} & \dots \\
\Omega_{4,1} & \Omega_{4,2} & \Omega_{4,3} & \Omega_{4,4} & \dots & \Omega_{4,k} & \dots \\
\vdots & \vdots & \vdots & \vdots & & \vdots & \\
\Omega_{k,1} & \Omega_{k,2} & \Omega_{k,3} & \Omega_{k,4} & \dots & \Omega_{k,k} & \dots \\
\vdots & \vdots & \vdots & \vdots & & \vdots &
\end{array} \tag{3}$$

is given on Ω with mesh sizes $h_m{=}2^{-m}$, $m{=}1,..,k$ and $h_n{=}2^{-n}$, $n{=}1,..,k$ in the x- and y-direction. Associated to it is the tableau of spaces of piecewise bilinear functions

$$\begin{array}{ccccccccccc}
V_{1,1} & \subset & V_{1,2} & \subset & V_{1,3} & \subset & V_{1,4} & \subset \cdots \subset & V_{1,k} & \subset \cdots \\
\cap & & \cap & & \cap & & \cap & & \cap & \\
V_{2,1} & \subset & V_{2,2} & \subset & V_{2,3} & \subset & V_{2,4} & \subset \cdots \subset & V_{2,k} & \subset \cdots \\
\cap & & \cap & & \cap & & \cap & & \cap & \\
V_{3,1} & \subset & V_{3,2} & \subset & V_{3,3} & \subset & V_{3,4} & \subset \cdots \subset & V_{3,k} & \subset \cdots \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots & \\
V_{k,1} & \subset & V_{k,2} & \subset & V_{k,3} & \subset & V_{k,4} & \subset \cdots \subset & V_{k,k} & \subset \cdots \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots &
\end{array} \tag{4}$$

with dimensions

$$N_{m,n} := dim(V_{m,n}) = (2^m - 1) \times (2^n - 1), \quad m, n = 1, .., k. \tag{5}$$

Here, 1 denotes the coarsest and k the finest level of discretization with respect to each direction. Note that the space $V_{m,n}$ is a subspace of $V_{m+1,n}$, $V_{m,n+1}$ and $V_{m+1,n+1}$. However, for example, the spaces $V_{m+1,n-1}$ and $V_{m-1,n+1}$ are not nested. Furthermore, consider the sets of grid points $\{x_1, ..., x_{N_{m,n}}\}$ in $\Omega_{m,n}$, $m, n{=}1,..,k$, not lying on the boundary.

The standard finite element basis that spans $V_{m,n}$ on the grid $\Omega_{m,n}$ is denoted by $B_{m,n}$. It contains the nodal basis functions $\phi_i^{(m,n)}$, $i{=}1,..,N_{m,n}$, that are defined by

$$\phi_i^{(m,n)}(x_j) = \delta_{i,j}, \quad x_j \in \Omega_{m,n}. \tag{6}$$

Note that we use rectangular grids. Therefore, the 2D basis functions can be written as the product of two 1D basis functions with, in general, different support for the different coordinate directions.

Now, the standard full grid space on $\Omega_{k,k}$ can be decomposed by

$$V_k := V_{k,k} = \sum_{m=1}^{k} \sum_{n=1}^{k} V_{m,n} = \sum_{m=1}^{k} \sum_{n=1}^{k} \sum_{i=1}^{N_{m,n}} V_{m,n,x_i}, \tag{7}$$

with $V_{m,n,x_i} := \text{span}\{\phi_i^{(m,n)}\}$, see Figure 1. Following [Zenger, 1991], the sparse grid can be defined by

$$\Omega_k^S := \bigcup_{m=1}^{m} \bigcup_{n=1}^{k+1-m} \Omega_{m,n} \tag{8}$$

(see Figure 2), with the corresponding finite element space

$$V_k^S := \sum_{m=1}^{k} \sum_{n=1}^{k+1-m} V_{m,n} = \sum_{m=1}^{k} \sum_{n=1}^{k+1-m} \sum_{i=1}^{N_{m,n}} V_{m,n,x_i}. \tag{9}$$

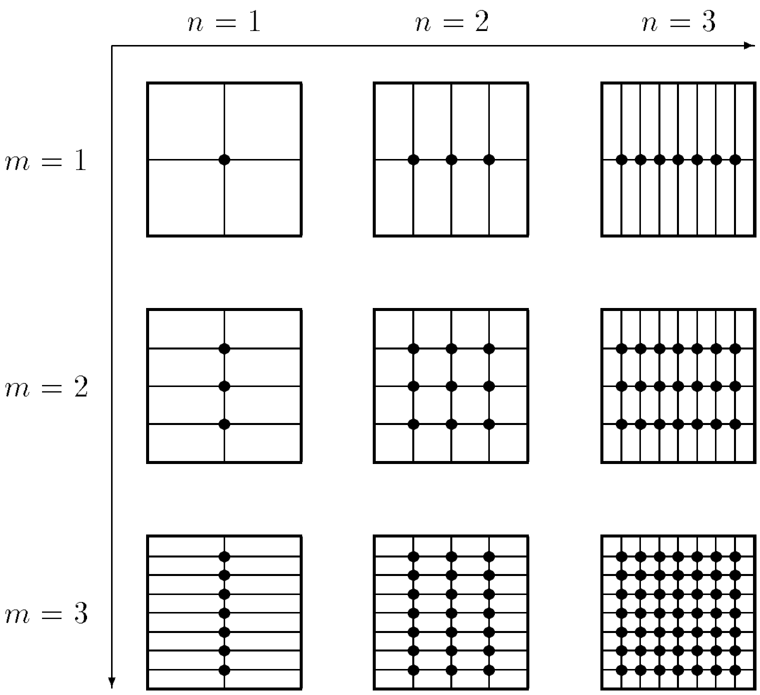

FIG. 1. *The quadratic scheme of grids* $\Omega_{m,n}$*,* $1 \leq m, n \leq k$*, contained in the full grid,* $k = 3$*.*

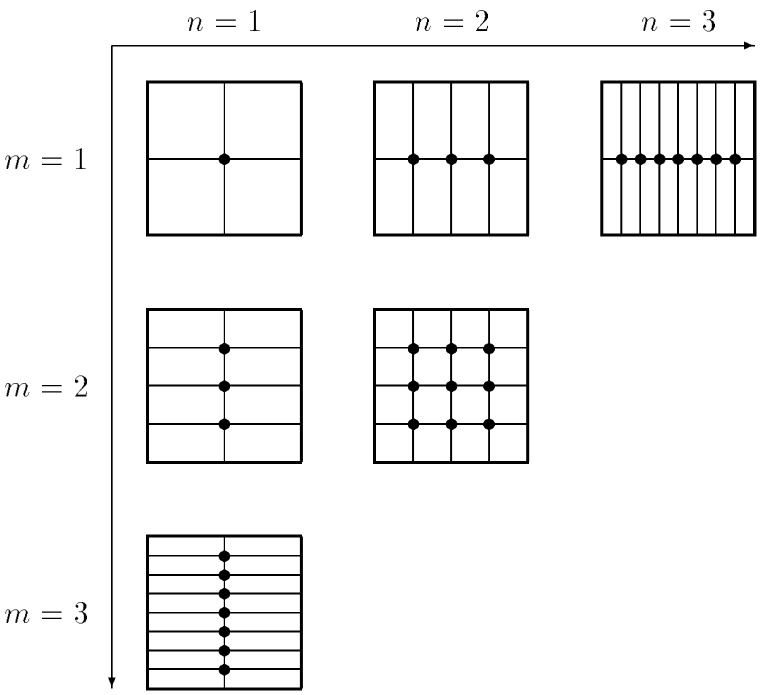

FIG. 2. *The triangular scheme of grids* $\Omega_{m,n}$*,* $m + n \leq k + 1$*, contained in the sparse grid,* $k = 3$*.*

Using the sparse grid space V_k^S instead of the full grid space V_k, the dimension is reduced substantially from $\dim V_k = O(h_k^{-2})$ to $\dim V_k^S = O(h_k \cdot \log(h_k^{-1}))$, while the order of the truncation error remains $O(h_k)$ with respect to the energy norm and is only slightly deteriorated from $O(h_k^2)$ to $O(h_k^2 \cdot \log(h_k^{-1}))$ with respect to the L_2-norm. For further discussions and details on the sparse grid approach see [Bungartz, 1992], [Griebel, 1991c], [Griebel et al., 1991] and [Zenger, 1991].

While any function $u \in V_k$ can be expressed uniquely by

$$u = \sum_{\phi \in B_{k,k}} u_\phi^B \cdot \phi \tag{10}$$

with the vector $u_k^B = (u_\phi^B)_{\phi \in B_{k,k}}$ of nodal values on the finest grid for some given ordering of the functions of $B_{k,k}$, the sparse grid space V_k^S does not possess such a nodal basis, since there exists no finest grid in the triangular scheme. Instead, following [Zenger, 1991], a product type hierarchical basis for V_k^S can be constructed by collecting for each grid $\Omega_{m,n}$ those basis functions from $B_{m,n}$ where the center point is not contained in any coarser grid. We define

$$\tilde{B}_{m,n} := \begin{cases} B_{1,1} & \text{for } m = n = 1, \\ \{\phi_i^{(m,1)} \in B_{m,1} : x_i \notin \Omega_{m-1,1}\} & \text{for } m > 1, n = 1, \\ \{\phi_i^{(1,n)} \in B_{1,n} : x_i \notin \Omega_{1,n-1}\} & \text{for } m = 1, n > 1, \\ \{\phi_i^{(m,n)} \in B_{m,n} : x_i \notin \Omega_{m-1,n} \cup \Omega_{m,n-1}\} & \text{for } m > 1, n > 1 \end{cases} \tag{11}$$

and obtain the hierarchical basis for V_k^S by

$$H_k^S := \bigcup_{m=1}^{k} \bigcup_{n=1}^{k+1-m} \tilde{B}_{m,n}. \tag{12}$$

Then, the hierarchical basis representation for a function $u \in V_k^S$ is denoted by

$$u = \sum_{\phi \in H_k^S} u_\phi^{HS} \cdot \phi \tag{13}$$

with the vector $u_k^{HS} := (u_\phi^{HS})_{\phi \in H_k^S}$ of hierarchical coefficients for some given ordering of H_k^S.

The use of this type of basis is not restricted to the sparse grid case. We introduce the hierarchical basis

$$H_k := \bigcup_{m=1}^{k} \bigcup_{n=1}^{k} \tilde{B}_{m,n} \tag{14}$$

for the full grid case. The corresponding hierarchical representation of a function $u \in V_k$ is now denoted by

$$u := \sum_{\phi \in H_k} u_\phi^H \cdot \phi \tag{15}$$

with the vector $u_k^H := (u_\phi^H)_{\phi \in H_k}$ of hierarchical coefficients for some given ordering of H_k.

The use of a basis has the advantage that any function can be expressed uniquely. However, with respect to the fast solution of the variational problem, the use of a basis can even be an obstacle. The resulting linear system is not well conditioned and algorithms for its solution have to be accelerated by quite complicated techniques like BPX-preconditioning or, in the multigrid context, by coarse grid correction. Therefore, in the following, we allow, analogously to [Griebel, 1991a], a non-unique representation of functions by using a generating system instead.

To this end, we define the set of functions

$$E_k = \bigcup_{m=1}^{k} \bigcup_{n=1}^{k} B_{m,n} \tag{16}$$

for the full grid case. This corresponds to the decomposition of V_k by (7), see also Figure 1. Clearly, as E_k contains now linearly dependent functions, it is no longer a basis for V_k but merely a generating system. A function $u \in V_k$ is represented non-uniquely in terms of the generating system E_k by

$$u = \sum_{\phi \in E_k} u_\phi^E \cdot \phi \tag{17}$$

with the vector $u_k^E := (u_\phi^E)_{\phi \in E_k}$ for some given ordering of E_k. The length of u_k^E is

$$N_k^E := \sum_{m=1}^{k} \sum_{n=1}^{k} N_{m,n}, \tag{18}$$

which is less than four times the length $N_{k,k}$ of the vector for the basis representation (10). This is due to the geometric progression of the number of grid points from level (m, n) to the levels $(m-1, n)$ and $(m, n-1)$ that decreases approximately by the factor $1/2$. So, the finest grid contains more than one quarter of all grid points in the quadratic scheme (Figure 1).

For the sparse grid case, we define analogously the set of functions

$$E_k^S = \bigcup_{m=1}^{k} \bigcup_{n=1}^{k+1-m} B_{m,n}. \tag{19}$$

This corresponds to the splitting (9) of the sparse grid space, see also Figure 2. As above, E_k^S contains linearly dependent functions and is no longer a basis for V_k^S but merely a generating system. A sparse grid function $u \in V_k^S$ is represented non-uniquely by

$$u = \sum_{\phi \in E_k^S} u_\phi^{ES} \cdot \phi \tag{20}$$

with the vector $u_k^{ES} := (u_\phi^{ES})_{\phi \in E_k^S}$ for some given ordering of E_k^S. The length of u_k^{ES} is

$$N_k^{ES} := \sum_{m=1}^{k} \sum_{n=1}^{k+1-m} N_{m,n}, \tag{21}$$

which is again less than four times the length of the corresponding basis representation (13). This is due to the inequality

$$|\tilde{B}_{m,n}| < \frac{1}{4}|B_{m,n}| \tag{22}$$

that holds for every grid $\Omega_{m,n}$.

Note once more that the representations (17) and (20) in terms of E_k and E_k^S are non-unique. However, for a given representation u_k^E of u in E_k or u_k^{ES} in E_k^S we can easily compute its representation u_k^B or v_k with respect to $B_{k,k}$ or H_k^S. This involves the bilinear interpolation which can be expressed and implemented by MG-prolongation operators with respect to standard- and semi-coarsening.

2.2. Galerkin approach and linear systems. Using the nodal basis $B_{k,k}$, the Galerkin-approach results in the discrete variational problem for $u \in V_k$

$$\forall \phi_i \in B_{k,k} : a(u, \phi_i) = f(\phi_i), \tag{23}$$

and the equivalent linear system of equations for the vector of nodal values

$$L_k^B u_k^B = f_k^B, \tag{24}$$

where

$$\begin{array}{rcll} (L_k^B)_{i,j} & := & a(\phi_j^{(k,k)}, \phi_i^{(k,k)}), & 1 \le i, j \le N_{k,k} \\ (f_k^B)_i & := & (f, \phi_i^{(k,k)}), & 1 \le i \le N_{k,k}. \end{array} \tag{25}$$

For the generating system E_k, the Galerkin-approach leads to the variational problem

$$\forall \phi_i \in E_k : a(u, \phi_i) = f(\phi_i), \tag{26}$$

and, with representation (17), to the linear system

$$L_k^E u_k^E = f_k^E, \tag{27}$$

where

$$\begin{array}{rcll} (L_k^E)_{i,j} & := & a(\phi_j, \phi_i), & \phi_i, \phi_j \in E_k \\ (f_k^E)_i & := & (f, \phi_i), & \phi_i \in E_k \end{array} \tag{28}$$

with an appropriate numbering of E_k (see below).

The system $L_k^E u_k^E = f_k^E$ has the following properties: The matrix L_k^E is semidefinite. It has the same rank as the matrix L_k^B that arises from the Galerkin approach by using the standard basis $B_{k,k}$ only. Thus, $N_k^E - N_{k,k}$ eigenvalues are zero. The system is solvable because the right hand side is constructed in a consistent manner. It has not one unique, but numerous, different solutions. However, the evaluation of two different solutions u_k^E and v_k^E with respect to their representation in $B_{k,k}$ by means of multigrid-prolongation operators results in the unique solution of (24). Therefore, it is sufficient to compute just one solution of the enlarged semidefinite system (27) to obtain, via interpolation and summation, the unique solution of (24).

Note that the enlarged matrix L_k^E contains the matrix L_k^B as a submatrix. The same is true for the right hand sides. The system that stems from the hierarchical basis

$$L_k^H u_k^H = f_k^H \tag{29}$$

where

$$\begin{array}{rcll} (L_k^H)_{i,j} & := & a(\phi_j, \phi_i), & \phi_i, \phi_j \in H_k \\ (f_k^H)_i & := & (f, \phi_i), & \phi_i \in H_k, \end{array} \tag{30}$$

is also contained.

Now, in the sparse grid case the systems for the generating system and the hierarchical basis are defined analogously. We obtain, using the generating system E_k^S, the linear system

$$L_k^{ES} u_k^{ES} = f_k^{ES}, \tag{31}$$

with

$$\begin{array}{rcll} (L_k^{ES})_{i,j} & := & a(\phi_j, \phi_i), & \phi_i, \phi_j \in E_k^S \\ (f_k^{ES})_i & := & (f, \phi_i), & \phi_i \in E_k^S \end{array} \tag{32}$$

and, using the hierarchical basis H_k^S, we obtain the linear system

$$L_k^{HS} u_k^{HS} = f_k^{HS} \tag{33}$$

with

$$\begin{array}{rcll} (L_k^{HS})_{i,j} & := & a(\phi_j, \phi_i), & \phi_i, \phi_j \in H_k^S \\ (f_k^{HS})_i & := & (f, \phi_i), & \phi_i \in H_k^S. \end{array} \tag{34}$$

Analogously to the full grid case, the unique solution of (33) can easily be computed from any solution of the semidefinite system (31).

3. Iterative Methods. Now, we use the semidefinite systems (27) and (31) to construct efficient solvers for the full and sparse grid problems, respectively. Analogously to [Griebel, 1991a], it is easy to see that (after appropriate scaling) the conjugate gradient method for the semidefinite systems $L_k^E u_k^E = f_k^E$ and $L_k^{ES} u_k^{ES} = f_k^{ES}$ is equivalent to certain preconditioned conjugate gradient methods for the full and sparse grid problems. The resulting preconditioners are similar to the one of Bramble, Pasciak and Xu [Bramble et al., 1990]. The Gauss-Seidel-iteration for the semidefinite systems is equivalent to the multigrid-method with a Gauss-Seidel smoother for the respective full and sparse grid problems. Here, depending on the ordering of the unknowns of the semidefinite systems, different MG-cycle strategies can be modeled easily. See the discussion in [Griebel, 1991a] for further guidance.

It is not necessary to assemble the matrices L_k^E or L_k^{ES} explicitly because it is possible to use a certain product type representation of L_k^E or L_k^{ES} analogously to [Griebel, 1991a] to implement the matrix-vector multiplication and the Gauss-Seidel iteration in $O(N_k^E)$ and $O(N_k^{ES})$ operations. The conjugate gradient or Gauss-Seidel method is still tricky to implement. A detailed description of implementation details will be given in a future paper. The basic approach follows the idea in [Bungartz, 1992] and [Zenger, 1991]. We use binary trees (of binary trees) to implement the full and sparse grid. Altogether, the required storage and the number of operations to perform one iteration is proportional to the number of grid points employed.

We will report the results of numerical experiments for the model problem

$$\begin{aligned} \epsilon^2 \cdot u_{xx} + u_{yy} &= f(x,y), \quad (x,y) \in \Omega = (0,1)^2, \\ u(x,y) &= 0, \qquad\quad\; (x,y) \in \delta\Omega. \end{aligned} \tag{35}$$

The tables contain the observed reduction factor ρ of the error in one iteration step and the number of iterations

$$it := \frac{-10}{\log \rho} \tag{36}$$

needed to reduce an arbitrary initial error by the factor 10^{-10}.

3.1. Gauss-Seidel and block Gauss-Seidel methods. First, we consider relaxation type iterations for the semidefinite system. Here, we will focus on the Gauss-Seidel (GS) and the block Gauss-Seidel (BGS) methods. We will show that different orderings of the generating systems and different partitions of the unknowns into blocks of coefficients result in a variety of relaxation schemes including multilevel methods and certain domain decomposition methods.

As usual, we decompose the semidefinite matrix L_k^E

$$L_k^E = F_k^E + G_k^E + (G_k^E)^T, \tag{37}$$

where F_k^E and G_k^E denote the diagonal and strictly lower diagonal parts of L_k^E, respectively. Then, the Gauss-Seidel iteration can be expressed by

$$W_k^E \left(u_k^{E,it+1} - u_k^{E,it} \right) = f_k^E - L_k^E u_k^{E,it} \tag{38}$$

with

$$W_k^E := F_k^E + G_k^E. \tag{39}$$

Note that, despite of the semidefiniteness of L_k^E, its lower diagonal part W_k^E is definite and thus invertible.

For the block Gauss-Seidel iteration, we use the splitting

$$L_k^E = \mathcal{F}_k^E + \mathcal{G}_k^E + (\mathcal{G}_k^E)^T, \tag{40}$$

where $\mathcal{F}_k^E$ is a block diagonal matrix and $\mathcal{G}_k^E$ is a strictly lower diagonal block matrix with respect to a given partition of the unknowns associated to E_k. We obtain the iteration

$$\mathcal{W}_k^E \left(u_k^{E,it+1} - u_k^{E,it} \right) = f_k^E - L_k^E u_k^{E,it} \tag{41}$$

with

$$\mathcal{W}_k^E := \mathcal{F}_k^E + \mathcal{G}_k^E. \tag{42}$$

Note that $\mathcal{W}_k^E$ is positive semidefinite and can be singular, depending on the respective block partitions. But, like equation (27), the equation (41) is solvable, because the right hand sides are constructed in a consistent manner. Furthermore, all solutions of (41) have the same basis representation with respect to $B_{k,k}$.

Alternatively, we can interpret the GS relaxation on the semidefinite system as a subspace correction method [Xu, 1990], where we relax a function $u^{(\mu)} \in V_k$ with respect to a $\phi \in E_k$ by

$$u^{(\mu+1)} := u^{(\mu)} + \lambda \cdot \phi \quad \text{with} \quad a(u^{(\mu+1)}, \phi) = f(\phi), \tag{43}$$

compare also the PMG method in [McCormick, 1992].

The BGS relaxation corresponds to the simultaneous relaxation of a set $\Phi \subset E_k$ by

$$u^{(\mu+1)} := u^{(\mu)} + \sum_{\phi \in \Phi} \lambda_\phi \cdot \phi \quad \text{with} \quad \forall \phi \in \Phi : a(u^{(\mu+1)}, \phi) = f(\phi), \tag{44}$$

which is equivalent to the solution of the variational problem for the error $e^{(\mu)} := u - u^{(\mu)}$

$$\forall \phi \in V_k : a(e^{(\mu)}, \phi) = f(\phi) - a(u^{(\mu)}, \phi), \tag{45}$$

restricted to the subspace $V_\Phi :=\ \mathrm{span}(\Phi)$.

Now, the GS method with respect to the generating system is the cyclic application of (43) for all $\phi \in E_k$ in a given order, while the BGS method applies (44) to all sets of a disjoint decomposition $E_k = \cup \Phi_k$ in some fixed order.

More generally, we allow to restrict the relaxation to some subset $E_k^{GS} \subset E_k$ and the multiple relaxation of some $\phi \in E_k^{GS}$ in one iteration step or, for BGS, a decomposition $E_k^{GS} = \cup \Phi_k$ with intersecting subsets. These iterations are convergent if and only if E_k^{GS} is a generating system for V_k.

Regarding the sparse grid case (31), just the generating system E_k for the full grid has to be exchanged by its sparse grid counterpart E_k^S and the above definitions and constructions can be repeated analogously.

3.1.1. Hierarchical bases method. If we iterate only the unknowns of the semidefinite system that are associated to the functions of the hierarchical basis, i. e. if we only use the subset $E_k^{GS} := H_k$ in the relaxation (43), we obtain the hierarchical basis method. Note that this is a GS iteration for the definite system $L_k^H u_k^H = f_k^H$, that differs from the usual GS iteration for $L_k^B u_k^B = f_k^B$ only by the use of the hierarchical

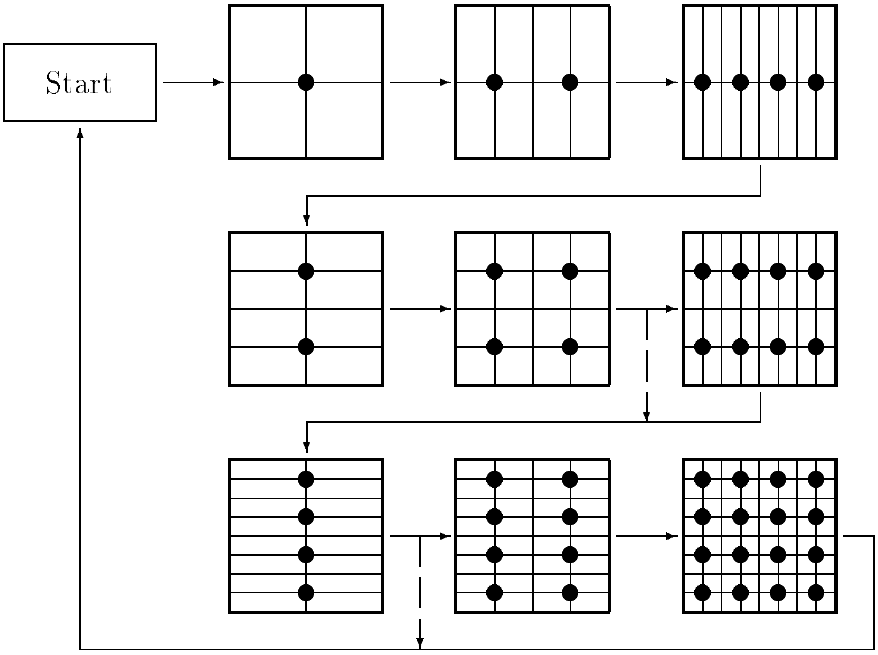

FIG. 3. *Sequence of relaxations for the hierarchical basis method. The dashed lines mark the short cuts for the sparse grid case.*

basis instead of the nodal basis on the finest grid. It is also suitable for the sparse grid case (with $E_k^{GS} := H_k^S$), see [Bungartz, 1992] for details.

Ordering the hierarchical basis level-wise

$$\begin{array}{cccccc}
\tilde{B}_{1,1}, & \tilde{B}_{1,2}, & \tilde{B}_{1,3}, & \tilde{B}_{1,4}, & \dots & \tilde{B}_{1,k}, \\
\tilde{B}_{2,1}, & \tilde{B}_{2,2}, & \tilde{B}_{2,3}, & \tilde{B}_{2,4}, & \dots & \tilde{B}_{2,k}, \\
\tilde{B}_{3,1}, & \tilde{B}_{3,2}, & \tilde{B}_{3,3}, & \tilde{B}_{3,4}, & \dots & \tilde{B}_{3,k}, \\
\tilde{B}_{4,1}, & \tilde{B}_{4,2}, & \tilde{B}_{4,3}, & \tilde{B}_{4,4}, & \dots & \tilde{B}_{4,k}, \\
 & & \vdots & & & \\
\tilde{B}_{k,1}, & \tilde{B}_{k,2}, & \tilde{B}_{k,3}, & \tilde{B}_{k,4}, & \dots & \tilde{B}_{k,k},
\end{array} \tag{46}$$

we traverse the unknowns associated to H_k according to Figure 3. Note that each $\tilde{B}_{i,j}$ contains only basis functions with disjoint supports, hence the sequence of relaxations inside each block $\tilde{B}_{i,j}$ is arbitrary and can be done parallel.

We obtain the algorithm

Hierarchical basis method (full grid)
    for $m = 1 \dots k$
        for $n = 1 \dots k$
            for $\phi \in \tilde{B}_{m,n}$
                relax ϕ according to (43).

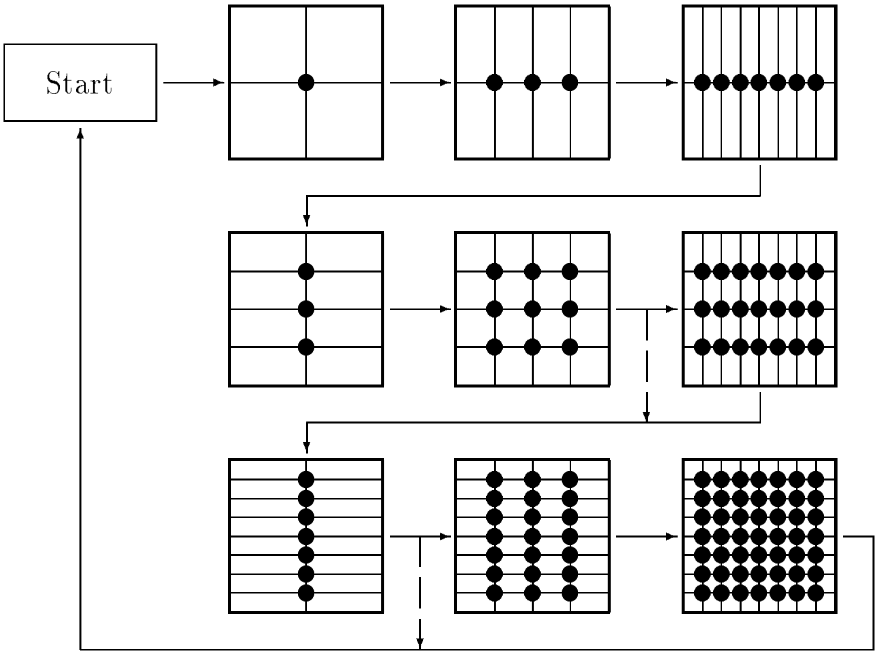

FIG. 4. *Sequence of relaxations for the multilevel method (outer loop). Again the dashed lines mark the short cuts for the sparse grid case.*

Analogously, we obtain for the sparse grid case the algorithm

Hierarchical basis method (sparse grid)
  for $m = 1 \ldots k$
    for $n = 1 \ldots k + 1 - m$
      for $\phi \in \tilde{B}_{m,n}$
        relax ϕ according to (43).

Table 1 shows the reduction rates of the hierarchical bases method for the full and sparse grid case. The number *it* of iterations seems to grow like $O(2^k)$ for the full grid case and like $O(2^{k/2})$ for the sparse grid case. Compare also the corresponding condition numbers in Tables 10 and 11.

TABLE 1
Reduction rates and numbers of iterations for the HB-Gauss-Seidel, $\epsilon = 1$

	k	3	4	5	6	7	8	9	10
Full grid	ρ	0.77	0.88	0.905	0.964	0.971	0.984	0.992	0.996
	it	88	180	230	630	780	1400	2900	5700
Sparse grid	ρ	0.56	0.68	0.77	0.83	0.88	0.913	0.938	0.955
	it	40	60	88	120	180	250	360	500

3.1.2. Grid-oriented methods. Instead of the functions $\tilde{B}_{m,n}$ as for the hierarchical basis method, we now take into account the complete nodal bases $B_{m,n}$ on the grids $\Omega_{m,n}$. This approach results in a multilevel-style iteration with an outer loop running over the different grids and inner loops relaxing the unknowns associated to the basis functions on the respective grid.

The traversal ordering for the grids is described in Figure 4. In contrast to the hierarchical basis method, the ordering of the unknowns of each grid is no longer arbitrary, since the supports of the associated functions overlap. Here, a lexicographical or a four-color ordering is sufficient. Altogether, we obtain in the full grid case the algorithm

Multilevel method (full grid)
  for $m = 1 \ldots k$
    for $n = 1 \ldots k$
      perform ν times a Gauss-Seidel iteration on $\Omega_{m,n}$.

In the sparse grid case, we obtain the analogous algorithm

Multilevel method (sparse grid)
  for $m = 1 \ldots k$
    for $n = 1 \ldots k + 1 - m$
      perform ν times a Gauss-Seidel iteration on $\Omega_{m,n}$.

Note the close relation of our algorithm to the methods of Mulder [Mulder, 1988], Naik and van Rosendale [Naik & van Rosendale, 1991] or Hackbusch [Hackbusch, 1987, Hackbusch, 1989].

With respect to the semidefinite system this corresponds to a level-wise partition of the coefficient vector u_k^E by

$$u_k^{E,L} := (u^{m,n})_{1 \le m,n \le k}, \text{ where } u^{m,n} := (u_\phi^E)_{\phi \in B_{m,n}} \tag{47}$$

denotes the coefficients belonging to $B_{m,n}$. In the sparse grid case, we obtain the level-wise partitioned coefficient vector

$$u_k^{ES,L} := (u^{m,n})_{1 \le m \le k, 1 \le n \le k+1-m} \,. \tag{48}$$

If we apply only one GS sweep on every grid, we obtain the standard GS method (38)-(39) for the semidefinite system with associated level-wise ordering of the unknowns.

Note that various traversal orderings for the different levels are possible beside the one described in figure 4. For example, a symmetric version of this method is obtained by a sweep as described in Figure 4 followed by s sweep in reverse ordering of the unknows, i. e. by the application of the symmetric Gauss-Seidel iteration with respect to the vector partition (47).

However, in numerical experiments with different traversal orderings, we obtained no particular difference in the observed reduction rates. Therefore, we restrict ourselves to the traversal ordering of Figure 4.

Table 2 shows the results for the full grid case. Here, a four color Gauss-Seidel relaxation is applied ν times on each grid. Table 3 shows the analogous results for the sparse grid case. Additionally, we present the results for the case of an exact solution

Table 2
Reduction rates and numbers of iterations for the multilevel Gauss-Seidel on the full grid, $\epsilon = 1$.

	k	3	4	5	6	7	8	9	10
$\nu = 1$	ρ	0.087	0.092	0.093	0.093	0.092	0.093	0.093	0.093
	it	9.4	9.7	9.7	9.7	9.7	9.7	9.7	9.7
$\nu = 2$	ρ	0.021	0.024	0.025	0.026	0.026	0.026	0.026	0.026
	it	6.0	6.2	6.2	6.3	6.3	6.3	6.3	6.3

Table 3
Reduction rates and numbers of iterations for the multilevel Gauss-Seidel on the sparse grid, $\epsilon = 1$.

	k	3	4	5	6	7	8	9	10
$\nu = 1$	ρ	0.066	0.10	0.18	0.22	0.27	0.32	0.33	0.34
	it	8.5	10	13	15	18	20	21	21
$\nu = 2$	ρ	0.036	0.098	0.054	0.11	0.054	0.11	0.080	0.11
	it	6.9	9.9	7.9	10.4	7.9	10.4	9.1	10.4
BGS	ρ	0.0026	0.0036	0.0058	0.0046	0.0058	0.0052	0.0058	–
	it	3.9	4.1	4.5	4.3	4.5	4.4	4.5	–

of the subproblems arising on each level of the sparse grid discretization for E_k^S. This is the block Gauss-Seidel iteration (41)-(42) with the level-wise decomposition (48) of the coefficient vector and the simultaneous relaxation of all unknowns belonging to the same grid. Hence, the diagonal blocks of $\mathcal{F}$ contain the matrices arising from the discretization of the problem on the corresponding grids.

For the full grid case, such a BGS method makes no sense. It would converge in one step, since the space $V_{k,k}$ associated to the finest grid contains the spaces $V_{m,n}$, $1 \le m, n \le k$ associated to all other grids. Therefore, the relaxation with respect to $V_{k,k}$ would cause the residual to vanish on all other grids.

Both, in the sparse and in the full grid case, it can be seen clearly that the reduction rates and the number of iterations are independent of k.

In Table 4, the case of the anisotropic operator is shown for fixed $k = 8$ and varying values of ϵ. We see that although the reduction factors get worse for $\epsilon \to 0$ and $\epsilon \to \infty$, they are still bounded. This demonstrates the robustness of our level-oriented method. Compare also the discussion in [Hackbusch, 1989] and [Naik & van Rosendale, 1991].

Table 4
Multilevel Gauss-Seidel in the anisotropic case, $k = 8$, $\nu = 1$.

	ϵ	0.0001	0.01	0.1	0.5	0.7	1	1.4	2	10	100	10000
Full grid	ρ	0.37	0.37	0.36	0.12	0.12	0.093	0.12	0.18	0.36	0.37	0.37
	it	23.2	23.2	22.5	10.9	10.9	9.7	10.9	13.4	22.5	23.2	23.2
Sparse grid	ρ	0.37	0.37	0.36	0.33	0.33	0.32	0.33	0.33	0.36	0.37	0.37
	it	23.2	23.2	22.5	20.8	20.8	20.2	20.8	20.8	22.5	23.2	23.2

3.1.3. Point-oriented methods. Now, we partition the semidefinite system into groups of unknowns where the associated generating functions are centered in the same grid point. Then, we perform a block Gauss-Seidel iteration on the associated block

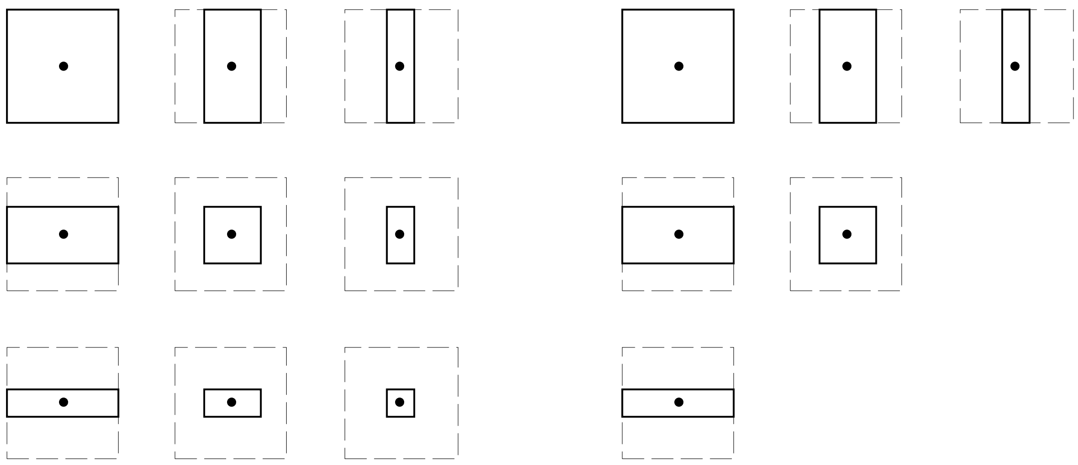

FIG. 5. *Supports of the generating functions* $B_{(\frac{1}{2},\frac{1}{2})}$ *(left) and* $B^S_{(\frac{1}{2},\frac{1}{2})}$ *(right) for* $k=3$.

partitioned system. Thus, as blocks in the iteration (41)-(42), we consider the unknowns that belong to the functions

$$B_x := \{\phi \in E_k : \phi(x) = 1\} \tag{49}$$

that are centered in the same grid point $x \in \Omega_{k,k}$. This corresponds to a point-oriented decomposition

$$V_k = \sum_{x\in\Omega_{k,k}} \sum_{m,n:x\in\Omega_{m,n}} V_{m,n,x} \tag{50}$$

of the space V_k. Note that in comparison to the level-wise decomposition (7) just the summations are exchanged. According to the decomposition of V_k, the coefficient vector u_k^E is partitioned as

$$u_k^{E,P} := (u^x)_{x\in\Omega_{k,k}}, \text{ where } u^x := (u_\phi^E)_{\phi\in B_x}. \tag{51}$$

For the sparse grid case, we define analogously

$$B_x^S := \{\phi \in E_k^S : \phi(x) = 1\}. \tag{52}$$

This corresponds to the decomposition

$$V_k = \sum_{x\in\Omega_k^S} \sum_{m+n\le k+1:x\in\Omega_{m,n}} V_{m,n,x} \tag{53}$$

and the partition

$$u_k^{ES,P} := (u^{S,x})_{x\in\Omega_k^S}, \text{ where } u^{S,x} := (u_\phi^{ES})_{\phi\in B_x^S}. \tag{54}$$

For an illustrating example, see Figure 5.

Now we can step through the sets of grid points $\Omega_{k,k}$ or Ω_k^S, respectively, and relax simultaneously the unknowns that belong to the same point. The relaxations (44) result in systems of linear equations for every grid point $x \in \Omega_{m,n} \setminus (\Omega_{m-1,n} \cup \Omega_{m,n-1})$ with $|B_x| = (k+1-m)(k+1-n)$ or $|B_x^S| = (k+2-m-n)(k+3-m-n)/2$ unknowns, respectively. The coupling between the point blocks is described by the off-diagonal blockmatrices of $\mathcal{G}_k^E$, or, for the formulation as subspace corrections, by the evaluation of $a(u^\mu, \phi)$ in (44). It is easy to see that information is exchanged by this coupling on all respective levels of discretization simultaneously.

In practice, however, not all traversal orderings through the set of grid points are advisable. We restrict ourselves to the traversal ordering already introduced for the hierarchical basis method (see also Figure 3), and obtain the following algorithm for the full grid case

Point-block method (full grid)
  for $m = 1 \ldots k$
    for $n = 1 \ldots k$
      for $x \in \Omega_{m,n} \setminus (\Omega_{m-1,n} \cup \Omega_{m,n-1})$
        relax B_x according to (44).

In the sparse grid case, we obtain the algorithm

Point-block method (sparse grid)
  for $m = 1 \ldots k$
    for $n = 1 \ldots k+1-m$
      for $x \in \Omega_{m,n} \setminus (\Omega_{m-1,n} \cup \Omega_{m,n-1})$
        relax B_x according to (44).

Note that it is often not necessary to compute the exact solution for each point-block problem. Then, a few GS relaxations on the respective point-block systems are sufficient. In the extreme case of one step only, we obtain the GS iteration for the semidefinite system with just a special point-oriented traversal ordering (51) or (54).

Table 5 shows the results for the full grid case with $\nu = 1$ and $\nu = 2$ GS iterations per point-block and with exact solution of the point-block systems (BGS).

TABLE 5
Reduction rates and numbers of iterations for the point-wise GS (full grid), $\epsilon = 1$.

	k	3	4	5	6	7	8	9	10
$\nu = 1$	ρ	0.17	0.18	0.22	0.24	0.27	0.28	0.29	0.32
	it	13.0	13.4	15.2	16.1	17.6	18.1	18.6	20.2
$\nu = 2$	ρ	0.069	0.089	0.092	0.093	0.10	0.13	0.17	0.21
	it	8.6	9.5	9.7	9.7	10.0	11.3	13.0	13.8
BGS	ρ	0.0022	0.0077	0.015	0.018	0.022	0.024	0.025	0.026
	it	3.8	4.7	5.5	5.7	6.0	6.2	6.2	6.3

Table 6 shows the analogous results for the sparse grid case. In both cases, we clearly see that the reduction rates are independent of k.

In Table 7, the case of the anisotropic operator is shown for fixed $k = 8$ and varying values of ϵ. We see that the reduction factors do not deteriorate for $\epsilon \to 0$ and $\epsilon \to \infty$,

TABLE 6
Reduction rates and numbers of iterations for the point-wise GS (sparse grid), $\epsilon = 1$.

	k	3	4	5	6	7	8	9	10	11	12	13
$\nu = 1$	ρ	0.087	0.22	0.028	0.32	0.35	0.40	0.42	0.42	0.42	0.43	0.43
	it	9.4	15.2	18.1	20.2	21.9	25.1	26.5	26.5	26.5	27.3	27.3
$\nu = 2$	ρ	0.032	0.095	0.15	0.20	0.21	0.26	0.26	0.27	0.29	0.29	0.29
	it	6.7	9.8	12.1	14.3	14.8	17.1	17.1	17.6	18.6	18.6	18.6
BGS	ρ	0.0022	0.0042	0.0086	0.015	0.018	0.019	0.021	0.023	0.024	0.024	0.024
	it	3.8	4.2	4.8	5.5	5.7	5.8	6.0	6.1	6.2	6.2	6.2

i. e. for strong anisotropic problems. In contrast to the results for the analogous grid-oriented methods (cf. Table 4), we even see a slight improvement of ρ for extreme values of ϵ. Thus, the worst reduction factor is obtained for the Poisson problem. This demonstrates the robustness of our point-oriented method.

TABLE 7
Point-wise BGS in the anisotropic case, $k = 8$.

	ϵ	0.0001	0.01	0.1	0.5	0.7	1	1.4	2	10	100	10000
Full grid	ρ	0.012	0.022	0.025	0.025	0.024	0.024	0.023	0.022	0.022	0.020	0.012
	it	5.2	6.0	6.2	6.2	6.2	6.2	6.1	6.0	6.0	5.9	5.2
Sparse grid	ρ	0.012	0.012	0.018	0.020	0.019	0.019	0.020	0.020	0.020	0.013	0.012
	it	5.2	5.2	5.7	5.9	5.8	5.8	5.8	5.8	5.8	5.3	5.2

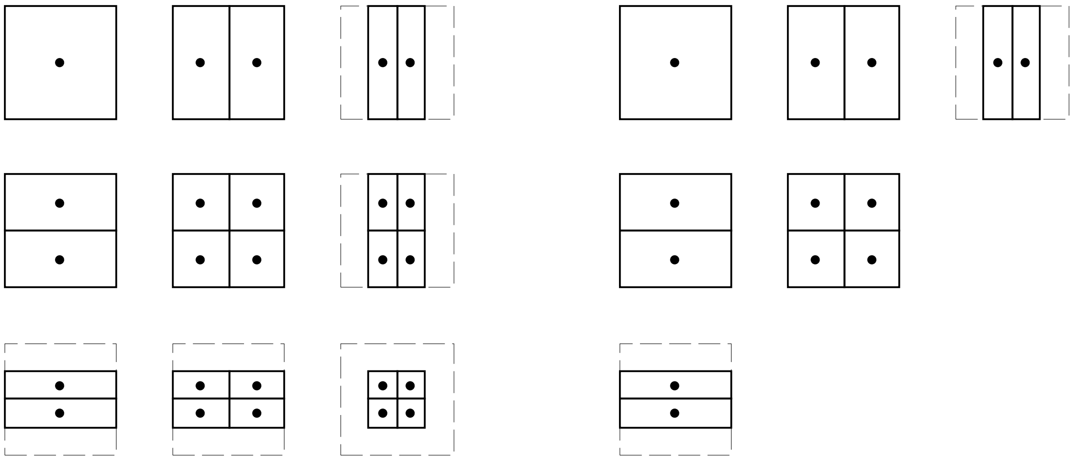

FIG. 6. *Supports of the generating functions $\hat{B}_{(\frac{1}{2},\frac{1}{2})}$ (left) and $\hat{B}^S_{(\frac{1}{2},\frac{1}{2})}$ (right) for $k = 3$.*

3.1.4. Neighborhood-oriented methods. The point-block methods can be modified using a system $\hat{B}_x$ for a grid point $x \in \Omega_{m,n} \setminus (\Omega_{m-1,n} \cup \Omega_{m,n-1})$ that contains all hierarchical basis functions centered in a point of the *hierarchical neighborhood* of x. The hierarchical neighborhood of x contains x itself and all those neighbor points of x on all finer grids $\Omega_{m',n'}$, $m \leq m' \leq k$, $n \leq n' \leq k$ that are contained in the interior of the support of $\phi_x^{(m,n)}$, the hierarchical basis function belonging to x. In the sparse grid case, $\hat{B}^S_x$ is constructed analogously. For an example, see Figure 6.

Note that $V_{B_x} := \text{span}(B_x) \subset V_{\hat{B}_x} := \text{span}(\hat{B}_x)$, since every $\phi \in B_x$ can be written in terms of $\hat{B}_x$. So, if we replace in the point-block algorithm B_x by $\hat{B}_x$, we solve the error equations (45) with respect to larger subspaces. However, Table 8 shows that this extra work gives only minimally improved reduction rates. Thus, from the efficiency point of view, the point-oriented method is superior.

TABLE 8
Reduction rates and numbers of iterations for the modified point-block Gauss-Seidel, $\epsilon = 1$.

	k	3	4	5	6	7	8	9	10
Full grid	ρ	0.0017	0.0076	0.014	0.018	0.021	0.023	0.024	–
	it	3.6	4.7	5.4	5.7	6.0	6.1	6.2	–
Sparse grid	ρ	0.0022	0.0042	0.0085	0.015	0.017	0.018	0.020	0.021
	it	3.8	4.2	4.8	5.5	5.7	5.7	5.7	5.9

3.1.5. Domain-oriented methods. The point-oriented approach can be easily generalized. We allow an arbitrary domain decomposition of Ω into K non-overlapping subdomains with associated decomposition

$$\Omega_{k,k} = \bigcup_{i=1}^{K} \Omega^i, \quad \Omega^i \cap \Omega^j = \emptyset \text{ for } i \neq j \tag{55}$$

of the gridpoints $\Omega_{k,k}$. For the sparse grid Ω_k^S, we consider an analogous decomposition. Now, we group the unknowns of the semidefinite system together that are associated to functions of E_k or E_k^S, respectively, where the center points are situated in the same $\Omega_{k,k}^i$,

$$B_{\Omega^i} := \bigcup_{x \in \Omega^i} B_x. \tag{56}$$

The resulting BGS algorithm now switches from subdomain to subdomain in some prescribed order. For practical purposes, a nested dissection-like decomposition [George, 1973] of the grid points into subdomains is advisable. Compare also Figure 7. Then, by using multigrid prolongation and restriction operators, the submatrices for the subdomains have not to be assembled explicitly and one overall BGS iteration can be performed in $O(|\Omega_{k,k}|)$ or $O(|\Omega_k^S|)$ operations, respectively.

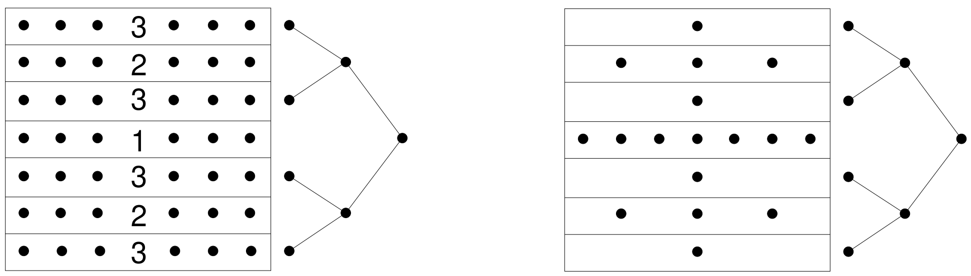

FIG. 7. *Nested dissection decomposition, full and sparse grid case, $k = 3$.*

In contrast to the usual domain decomposition approach, where just the standard basis is used in the discretization, the matrices for the subdomains are now in general no longer invertible, since they can be semidefinite. However, the application of a (level-oriented) GS-iteration automatically results in a multigrid-like solver for the subdomain problem, that, due to the generating system approach, now produces a non-unique representation of the solution for the subdomain. This non-unique solution can be cast easily via interpolation operators and additions into its unique representation with respect to $B_{k,k}$ or H_k^S, respectively.

By using the generating system we have no longer to bother for the computation of the Schur complement. Instead, a semidefinite subsystem appears for each subdomain. Furthermore, there is no more need to speed up convergence by means of global coarse grid transport steps or by the use of overlapping domains like in standard domain decomposition methods.

Now, consider for example the stripe-wise approach in Figure 8. All functions of E_k associated to the grid points of different stripes with the same number are mutually orthogonal with respect to $a(.,.)$, since they possess disjoint supports. Therefore, the stripe subproblems with the same numbers can be computed fully in parallel and we see directly a binary tree ordering for the parallel execution of our algorithm. Communication has to take place only between father and son stripes, i. e. between next-neighbored stripes with successive numbers, and not on each level of discretization like in many parallel multigrid algorithms. Compare also Figure 8.

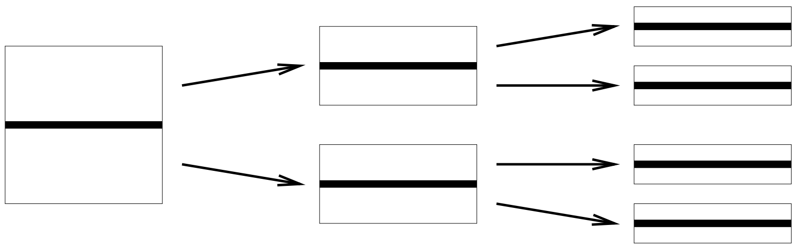

FIG. 8. *Domain decomposition of the unit square: in each subdomain all basis functions belonging to the centerline are relaxed simultaneously.*

Table 9 shows the resulting reduction rates for the line-wise BGS method. Once

TABLE 9
Reduction rates and numbers of iterations for the line-wise BGS, $\epsilon = 1$.

	k	3	4	5	6	7	8	9	10
Full grid	ρ	0.00094	0.0050	0.012	0.017	0.020	0.021	0.022	0.023
	it	3.3	4.3	5.2	5.7	5.9	6.0	6.0	6.1
Sparse grid	ρ	0.0010	0.0023	0.0053	0.013	0.016	0.018	0.020	0.021
	it	3.3	3.8	4.4	5.3	5.5	5.7	5.9	6.0

more, we see that in both cases the reduction rates are independent of k. However, in comparison to the point-oriented method (see Tables 5 and 6), we observe no significant improvement of the reduction rates. Thus, the line-wise BGS method is surely less efficient than the associated point-oriented approach, since it involves the exact solution of the stripe subproblems. If we use an inexact solver instead, like for example a point-wise BGS sweep over the subproblems with an appropriate traversal ordering of the points in each stripe, then we obtain the point-block method again, compare Tables 5 and 6 (row BGS). If we substitute in a second step the accurate BGS steps for each point by e. g. one GS step for the point-block subsystem, we obtain just the plain GS method for the semidefinite system but with a special domain- and point-oriented traversal ordering, compare Tables 5 and 6 (row $\nu = 1$). Now, the number of operations involved is comparable to the corresponding level-oriented methods. However, in contrast to the level-oriented approach, these methods have now the same parallelization possibilities as the domain-oriented methods.

3.2. Preconditioned conjugated gradient methods and scaling. Now we study conjugate gradient type methods (CG) for the semidefinite system. Since the performance of a CG step depends with

$$\rho \leq \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \tag{57}$$

on the condition number κ of the respective matrix, we will focus on different preconditioners $\mathcal{C}_k^E$ for the semidefinite matrices L_k^E and L_k^{ES}, respectively. We use in the following the so called generalized condition number

$$\hat{\kappa}(L) := \frac{\lambda_{max}(L)}{\hat{\lambda}_{min}(L)}, \tag{58}$$

the ratio of the maximal and the minimal nonzero eigenvalues of L.

Beside simple scalings, where $\mathcal{C}_k^E$ is a diagonal matrix with full or deficient rank, we consider the case of point-block preconditioning analogously to the point-wise BGS described above.

3.2.1. Hierarchical bases CG. Consider now the CG method using the hierarchical bases B_k and B_k^S, respectively. While the Gauss-Seidel relaxation is not sensitive to the scaling of the basis functions, the condition number κ of the coefficient matrices L_k^H, L_k^{HS} and thus the convergence rate of the CG method depend on the proper scaling of the system. Oswald [Oswald, 1992] shows that, without any scaling of the basis functions, the estimate

$$C_1 2^k \leq \kappa(L_k^{HS}) \leq C_2 2^k k^5 \tag{59}$$

holds with constants C_1 and C_2 independent of k. The condition number can be improved by a level-wise scaling of the basis functions according to

$$\check{B}_{m,n} := \{a(\phi,\phi)^{-1/2} \cdot \phi : \phi \in \tilde{B}_{m,n}\}. \tag{60}$$

This results in the matrix

$$\check{L}_k^{HS} = (\operatorname{diag} L_k^{HS})^{-1/2} L_k^{HS} (\operatorname{diag} L_k^{HS})^{-1/2} \tag{61}$$

for which the estimate

$$C_1 2^{k/2} \leq \kappa(\check{L}_k^{HS}) \leq C_2 2^{k/2} k^3 \tag{62}$$

can be obtained (see [Linner, 1992] or [Oswald, 1992]). In practice, even a $O(k \cdot 2^{k/2})$ behavior of $\kappa(\check{L}_k^{HS})$ can be observed (see Table 11). Furthermore, it can be shown that for *any* scaling of the hierarchical basis functions the condition number of the corresponding matrix $\bar{L}_k^{HS}$ satisfies

$$\kappa(\bar{L}_k^{HS}) \geq C \cdot 2^{k/2}. \tag{63}$$

Thus, using the hierarchical basis functions only, at least a condition number that grows exponentially with the number of levels k can be gained.

In the full grid case, we also observe an exponential grow of the corresponding condition numbers $\kappa(L_k^H)$ and $\kappa(\check{L}_k^H)$. This has to be compared to the $O(k^2)$ result obtained by Yserentant's hierarchical basis method [Yserentant, 1986], where only standard coarsening is applied. Thus, the use of the basis functions with extremely distorted rectangular support causes the convergence rate of the hierarchical basis method to deteriorate.

Table 10 shows the numerically computed eigenvalues and condition numbers without and with diagonal scaling for the full grid case, Table 11 shows the corresponding numbers for the sparse grid case. We see the improvement achieved by diagonal scaling.

TABLE 10
Eigenvalues and condition numbers of L_k^H and $\check{L}_k^H$ (full grid, $\epsilon = 1$).

	k	3	4	5	6	7	8
L_k^H	λ_{min}	0.516	0.326	0.226	0.126	0.119	0.0908
	λ_{max}	9.77	18.7	36.7	73.0	145.9	291.7
	κ	18.9	57.4	161.5	450.6	1226.1	3212.6
$\check{L}_k^H$	λ_{min}	0.139	0.0678	0.0339	0.0170	0.00847	0.00424
	λ_{max}	2.53	3.32	4.10	4.88	5.66	6.43
	κ	18.2	49.0	105.1	287.1	668.2	1517

TABLE 11
Eigenvalues and condition numbers of L_k^{HS} and $\check{L}_k^{HS}$ (sparse grid, $\epsilon = 1$).

	k	3	4	5	6	7	8	9	10
L_k^{HS}	λ_{min}	0.736	0.595	0.490	0.400	0.314	0.258	0.215	0.181
	λ_{max}	6.1	10.9	21.4	42.7	85.4	170.6	341.3	682.7
	κ	8.3	18.3	43.7	106.9	271.5	661.2	1587	3771
$\check{L}_k^{HS}$	λ_{min}	0.249	0.186	0.135	0.098	0.0665	0.0484	0.0331	0.0241
	λ_{max}	1.77	2.12	2.55	2.90	3.34	3.69	4.12	4.47
	κ	7.1	11.4	18.9	29.5	50.2	76.2	124.5	185.5

Of course, since the sparse grid matrix L_k^{HS} is a minor of the full grid matrix L_k^H, the corresponding condition number has to be less than or equal to the condition number of the full grid matrix. However, in both cases we see the dependency of the eigenvalues and condition numbers from k.

3.2.2. Level-wise scaling of the semidefinite system. Now, we consider the semidefinite linear systems L_k^E and L_k^{ES} that arise from the generating systems E_k and E_k^S, respectively. Here, we focus on the level-oriented approach. Analogously to [Griebel, 1991a], for the properly scaled generating system we obtain methods similar to the preconditioners of Bramble, Pasciak and Xu [Bramble et al., 1990] or Zhang's MDS [Zhang, 1992]. Thus, in principle, a Jacobi type diagonal scaling has to be performed for the semidefinite system. However, this is only true for the standard-refinement case. For our approach, where also the basis functions $B_{m,n}$, $m \neq n$, are contained in the generating systems E_k and E_k^S (which is a must for the sparse grid case, since these functions are needed to span V_k^S), straightforward Jacobi-scaling is no longer sufficient to gain a $O(1)$-conditioned matrix.

Therefore, we consider a weighted Jacobi scaling, where the weighting is done level-wise by

$$\breve{B}_{m,n} := \{\gamma_{m,n}^{1/2} a(\phi,\phi)^{-1/2} \cdot \phi : \phi \in B_{m,n}\}. \tag{64}$$

with constants $\gamma_{m,n} \geq 0$.

We assume a level-wise ordering (47) of the unknowns and, with the $N_k^E \times N_k^E$ diagonal matrix Γ_k^E with diagonal blocks $\gamma_{m,n} I_{N_{m,n}}$, we obtain the preconditioned semidefinite matrix

$$\breve{L}_k^E := (\Gamma_k^E)^{1/2} (\mathrm{diag}(L_k^E))^{-1/2} L_k^E (\mathrm{diag}(L_k^E))^{-1/2} (\Gamma_k^E)^{1/2} \tag{65}$$

for the full grid case. In the sparse grid case, we obtain analogously the matrix

$$\breve{L}_k^{ES} := (\Gamma_k^{ES})^{1/2} (\mathrm{diag}(L_k^{ES}))^{-1/2} L_k^{ES} (\mathrm{diag}(L_k^{ES}))^{-1/2} (\Gamma_k^{ES})^{1/2}. \tag{66}$$

Note that $\breve{L}_k^E$ has the same generalized condition number as $C_k L_k$, where C_k is a preconditioner to the fine grid matrix L_k (24) that can be written as

$$C_k = \sum_{m=1}^{k} \sum_{n=1}^{k} \gamma_{m,n} P_{m,n} (\mathrm{diag}(L_{m,n}))^{-1} R_{m,n}. \tag{67}$$

with the canonical prolongation (interpolation) $P_{m,n} : V_{m,n} \to V_k$ and its adjoint restriction operator $R_{m,n} : V_k \to V_{m,n}$.

For the sparse grid case, we obtain

$$C_k^S = \sum_{m=1}^{k} \sum_{n=1}^{k+1-m} \gamma_{m,n} P_{m,n}^S (\mathrm{diag}(L_{m,n}))^{-1} R_{m,n}^S, \tag{68}$$

where the interpolation operator is denoted by $P_{m,n}^S : V_{m,n} \to V_k^S$ and $R_{m,n}^S : V_k^S \to V_{m,n}$ is its adjoint operator. Now, the crucial task is to determine weights $\gamma_{i,j}$ so that the generalized condition number of the arising matrix is independent of k. In the full grid case, the choice

$$\gamma_{i,j} = \delta_{i,j} \tag{69}$$

results in the BPX-preconditioner (see [Griebel, 1991a]) and thus in a $O(1)$ method [Oswald, 1991]. The search for other weights with perhaps better constants for the condition numbers as well as the theory to cover these methods are subject of actual research.

In the sparse grid case, the above choice is no longer advisable, since the rank of the preconditioned matrix $\check{L}_k^{ES}$ would be less than the rank of $\hat{L}_k^{HS}$ that stems from the hierarchical basis. Simple diagonal scaling

$$\gamma_{i,j} = 1 \tag{70}$$

results in a matrix with a condition number growing practically like $O(k^2)$ (compare Table 12) and is no suitable choice, either. However, with the weights

$$\gamma_{i,j} = \begin{cases} 1 & i = j \vee i + j = k + 1, \\ 0 & \text{else}, \end{cases} \tag{71}$$

we obtain a method that can be proven to be $O(1)$, see [Griebel & Oswald, 1992].

Table 12 shows the numerical results for the choice $\gamma_{i,j} = 1$ for the sparse grid case. We can see, at least in practice, a $O(k^2)$ behavior of the condition number.

TABLE 12
Generalized condition numbers for the multilevel preconditioned $\check{L}_k^{ES}$ with $\gamma_{i,j} = 1$, $\epsilon = 1$.

k	3	4	5	6	7	8	9	10
$\hat{\lambda}_{min}$	1.05	0.862	0.714	0.671	0.621	0.562	0.543	0.534
λ_{max}	4.14	6.24	8.75	11.3	14.3	17.5	21.0	24.6
$\hat{\kappa}$	3.94	7.24	12.3	16.8	23.0	31.2	38.7	46.1
$\hat{\kappa}/k^2$	0.44	0.45	0.49	0.47	0.47	0.49	0.48	0.46

Table 13 shows the numerical results for the choice of $\gamma_{i,j}$ from (71) for the sparse grid case. From the theoretical results in [Griebel & Oswald, 1992], we know that the condition numbers are bounded by a constant. For practical grid sizes, however, we see moderately growing condition numbers. In Table 14, the case of the anisotropic

TABLE 13
Generalized condition numbers for the multilevel preconditioned $\check{L}_k^{ES}$ with $\gamma_{i,j}$ from (71), $\epsilon = 1$.

k	3	4	5	6	7	8	9	10	11	12	13
$\hat{\lambda}_{min}$	0.962	0.862	0.709	0.667	0.621	0.560	0.543	0.535	0.518	0.513	0.510
λ_{max}	2.58	3.71	3.89	4.71	4.92	5.60	5.75	6.28	6.39	6.84	6.93
$\hat{\kappa}$	2.68	4.30	5.48	7.07	7.92	10.0	10.6	11.7	12.3	13.3	13.6

operator is shown for fixed $k = 8$ and varying values of ϵ. We see that the condition number deteriorates somewhat for $\epsilon \to \infty$, but is still bounded. The same behavior was obtained for $\epsilon \to 0$. The condition number seems to have an upper bound independent of k and ϵ, at least for a wide range of values of ϵ. In numerical experiments, we found the condition number to deteriorate only near $\epsilon \approx k$. In this sense, the choice $\gamma_{i,j}$ from (71) results in a quite robust preconditioner. In contrast to that, compare the results for the standard full grid BPX-preconditioner in Table 16.

TABLE 14
Generalized condition numbers for the multilevel preconditioned $\breve{L}_k^{ES}$ with $\gamma_{i,j}$ from (71), $k = 8$.

ϵ	1	1.4	2	10	100	1000	10000
$\hat{\lambda}_{min}$	0.560	0.549	0.532	0.264	0.503	0.501	0.500
λ_{max}	5.60	6.31	7.14	9.42	11.2	11.5	11.5
$\hat{\kappa}$	10.0	11.5	13.4	35.7	22.3	23.0	23.0

In the full grid case we recommend the scaling $\gamma_{i,j} := \delta_{i,j}$, at least for the case of the Laplacian. That results in the well-known BPX type preconditioning. Table 15 shows, for comparison only, the resulting eigenvalues and condition numbers. Compare also the results in [Zhang, 1992]. In Table 16, the case of the anisotropic operator is shown for fixed $k = 5$ and varying values of ϵ. Here, with growing ϵ, we see a strong increase of the condition number for already quite moderate values of ϵ. Although it reaches a (really bad) upper bound for extreme values of ϵ, the resulting CG-algorithm is worthless for strongly anisotropic problems. The appropriate choice of $\gamma_{i,j}$ for the anisotropic operator in the full grid case is still an open question.

TABLE 15
Generalized condition numbers for the multilevel preconditioned $\breve{L}_k^{E}$ with $\gamma_{i,j} = \delta_{i,j}$, $\epsilon = 1$.

k	3	4	5	6	7	8	9	10	11
$\hat{\lambda}_{min}$	0.77	0.76	0.76	0.76	0.75	0.75	0.75	0.75	0.75
λ_{max}	2.3	2.7	3.1	3.3	3.6	3.8	3.9	4.1	4.2
$\hat{\kappa}$	3.0	3.6	4.1	4.4	4.7	5.0	5.3	5.5	5.6

TABLE 16
Generalized condition numbers for the multilevel preconditioned $\breve{L}_k^{E}$ with $\gamma_{i,j} = \delta_{i,j}$, $k = 5$.

ϵ	1	1.4	2	10	100	1000	10000	100000
$\hat{\lambda}_{min}$	0.76	0.51	0.30	0.196	0.00272	0.00244	0.00242	0.00242
λ_{max}	3.1	3.8	4.6	5.6	5.7	5.7	5.7	5.7
$\hat{\kappa}$	4.1	7.5	15.3	28.6	2095	2336	2355	2355

3.2.3. Point-block inversion and scaling. For an analogon of the point-block relaxation methods in the CG context, we consider the systems B_x, $x \in \Omega_{k,k}$, see (49), but now with point-wise chosen weighting factors γ_x,

$$\breve{B}_x := \{\gamma_x^{1/2} \cdot \phi : \phi \in B_x\}. \tag{72}$$

With a point-wise ordering (51) of the unknowns, the scaling is done by the diagonal matrix Γ_k^E, that contains the diagonal blocks $\gamma_x I_{|B_x|}$.

Now, we define for each $x \in \Omega_{k,k}$ the space

$$V_x := \langle B_x \rangle \subset V_{k,k}, \tag{73}$$

the canonical prolongation

$$P_x : V_x \to V_{k,k}, \tag{74}$$

its adjoint restriction

$$R_x := P_x^T \tag{75}$$

and the point-block matrix

$$L_x := R_x L_k P_x, \tag{76}$$

which is the same matrix that appeared as an entry in the block matrices $\mathcal{F}_k^E$ and $\mathcal{G}_k^E$ of the point-block relaxation. Note that, since B_x is a basis for V_x, the matrix L_x is positive definite.

Using the same ordering as for Γ_k^E, we define the block diagonal matrix Π_k^E with L_x as diagonal blocks and get the point-wise preconditioned matrix for the semidefinite system

$$\breve{L}_k^E := (\Gamma_k^E)^{1/2}(\Pi_k^E)^{-1/2} L_k^E (\Pi_k^E)^{-1/2}(\Gamma_k^E)^{1/2}. \tag{77}$$

Again this is equivalent to a preconditioner C_k for the fine grid matrix L_k with

$$C_k := \sum_{x\in\Omega_{k,k}} \gamma_x P_x L_x^{-1} R_x. \tag{78}$$

For the sparse grid, we define analogously

$$\breve{B}_x^S := \{\gamma_x^{1/2}\cdot\phi : \phi \in B_x^S\}, \tag{79}$$

$V_x^S = \langle B_x^S\rangle$, the prolongation $P_x^S : V_x^S \to V_k^S$, its adjoint restriction $R_x^S := (P_x^S)^T$, and the matrices $L_x^S := R_x^S \hat{L}_k^S P_x^S$, Γ_k^{ES}, Π_k^{ES} and get

$$\breve{L}_k^{ES} := (\Gamma_k^{ES})^{1/2}(\Pi_k^{ES})^{-1/2} \hat{L}_k^{ES} (\Pi_k^{ES})^{-1/2}(\Gamma_k^{ES})^{1/2} \tag{80}$$

and

$$C_k^S := \sum_{x\in\Omega_k^S} \gamma_x P_x^S (L_x^S)^{-1} R_x^S. \tag{81}$$

See Table 17 for the results of numerical experiments. It turns out that the choice $\gamma_x := 1$ gives for both the sparse grid case and full grid case the best condition numbers. Note at last that we expect an additional diagonal scaling of $\breve{L}_k^E$ and $\breve{L}_k^{ES}$ to improve the condition numbers a little more. However, with respect to the efficiency, the point-block preconditioner does not achieve the quality of the level-wise scaling according to (69) or (71), anyway.

TABLE 17

Eigenvalues and condition numbers for the point-wise preconditioned system, $\epsilon = 1$

	k	3	4	5	6	7	8
Sparse grid	λ_{min}	1	1	1	1	1	1
$\gamma_x := \|B_x\|$	λ_{max}	13.0	26.5	49.6	81.9	123.3	174.1
Sparse grid	λ_{min}	1	1	1	1	1	1
$\gamma_x := \sqrt{\|B_x\|}$	λ_{max}	6.91	11.3	18.2	25.0	32.0	41.5
Sparse grid	λ_{min}	0.995	0.993	0.988	0.985	0.981	0.979
$\gamma_x := 1$	λ_{max}	4.00	5.45	7.55	8.76	10.5	11.7
	κ	4.02	5.49	7.64	8.89	10.7	11.95
Sparse grid	λ_{min}	0.408	0.315	0.257	0.216	0.187	0.165
$\gamma_x := \sqrt{\|B_x\|^{-1}}$	λ_{max}	2.56	3.06	3.72	3.97	4.45	4.62
	κ	2.27	9.71	14.5	18.4	23.8	28.0
Sparse grid	λ_{min}	0.166	0.0997	0.0663	0.0473	0.0354	0.0275
$\gamma_x := \|B_x\|^{-1}$	λ_{max}	1.83	2.00	2.22	2.29	2.41	2.45
	κ	11.0	20.1	33.5	48.4	68.1	89.1
Full grid	λ_{min}	1	1	1	1	–	–
$\gamma_x := \sqrt{\|B_x\|}$	λ_{max}	14.2	23.7	35.7	49.1	–	–
Full grid	λ_{min}	0.996	0.991	0.986	0.982	–	–
$\gamma_x := 1$	λ_{max}	7.55	10.5	13.2	15.9	–	–
	κ	7.58	10.6	13.4	16.2	–	–
Full grid	λ_{min}	0.333	0.249	0.198	0.164	–	–
$\gamma_x := \sqrt{\|B_x\|^{-1}}$	λ_{max}	4.5	5.6	6.43	7.20	–	–
	κ	13.5	22.5	32.5	43.9	–	–

4. Concluding remarks. In this paper we presented different multilevel algorithms based on the generating system approach. We studied level-oriented techniques where GS methods for the arising semidefinite system turn out to be either multigrid or hierarchical basis methods with semi-coarsening. The application of CG results in BPX type preconditioners. Additionally, we presented point-oriented methods. There, as well as in the generalized case of domain-oriented methods, GS and BGS iterations for the semidefinite system are obtained that exhibit a reduction rate independent of the grid size and number of levels like conventional MG methods. These algorithms possess favorable properties with respect to parallelization and adaptivity. In contrast to the parallelization of a multigrid method, where communication has to take place on every level of discretization, our new algorithms have communication requirements like conventional domain decomposition methods.

Unlike the grid- and point-oriented methods introduced in [Griebel, 1992] that use standard-coarsening, the use of semi-coarsening allows the straightforward extension to the case of sparse grids. Additionally, the resulting algorithms are robust for anisotropic model problems.

We showed that the use of the generating system approach is an excellent guideline for the construction of multilevel- and multigrid-like algorithms. The extension to higher dimensional cases is straightforward. Note at last that an extension to the case of adaptive refinement is quite easy for the point-block methods.

REFERENCES

[Bramble et al., 1990] Bramble, J., Pasciak, J., & Xu, J. (1990). Parallel multilevel preconditioners. *Math. Comp.*, 31, 333–390.

[Bungartz, 1992] Bungartz, H.-J. (1992). *Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung.* PhD thesis, Institut für Informatik, TU München.

[George, 1973] George, A. (1973). Nested dissection of a regular finite element mesh. *SIAM, J. Numer. Anal.*, 10, 345–363.

[Griebel, 1991a] Griebel, M. (1991a). *Multilevel algorithms considered as iterative methods on indefinite systems.* SFB-Report 342/29/91 A; TUM-I9143, TU München, Institut f. Informatik.

[Griebel, 1991b] Griebel, M. (1991b). Parallel multigrid methods on sparse grids. In *Int. Series of Numer. Mathematics, Vol. 98* Basel: Birkhäuser Verlag.

[Griebel, 1991c] Griebel, M. (1991c). A parallelizable and vectorizable multi-level algorithm on sparse grids. In W. Hackbusch (Ed.), *Parallel Algorithms for Partial Differential Equations, Proceedings of the Sixth GAMM-Seminar, Kiel, January 19-21, 1990*: Vieweg-Verlag.

[Griebel, 1992] Griebel, M. (1992). Grid- and point-oriented multilevel algorithms. In W. Hackbusch (Ed.), *Incomplete Decomposition (ILU): Theory, Technique and Application, Proceedings of the Eigth GAMM-Seminar, Kiel January 24-26, 1992*: Vieweg-Verlag.

[Griebel & Oswald, 1992] Griebel, M. & Oswald, P. (1992). *On Additive Schwarz Preconditioners for Sparse Grid Discretizations.* Report Math/92/7, Friedrich-Schiller-Universität Jena. (Submitted to Numerische Mathematik).

[Griebel et al., 1991] Griebel, M., Schneider, M., & Zenger, C. (1991). *A Combination Technique for the Solution of Sparse Grid Problems.* SFB-Report 342/19/90 A; TUM-I9038, TU München, Institut f. Informatik.

[Hackbusch, 1987] Hackbusch, W. (1987). *A New Appoach to Robust Multi-Grid Solvers.* Report, Institut für Informatik und praktische Mathematik, CAU Kiel.

[Hackbusch, 1989] Hackbusch, W. (1989). The frequency decomposition multi-grid method, part I: Applications to anisotropic equations. *Numerische Mathematik*, 56.

[Linner, 1992] Linner, T. (1992). Konvergenz iterativer Verfahren mit dünnen Gittern. Master's thesis, TU München, Institut für Informatik.

[McCormick, 1992] McCormick, S. F. (1992). Multilevel projection methods for partial differential equations. In *CBMS-NSF Regional Conference Series in Applied Mathematics.* SIAM.

[Mulder, 1988] Mulder, W. (1988). *A New Multigrid Approach to Convection Problems.* Report 88-04, CAM.

[Naik & van Rosendale, 1991] Naik, N. H. & van Rosendale, J. (1991). *The Improved Robustness of Multigrid Elliptic Solvers Based on Multiple Semicoarsened Grids.* Report 55-70, ICASE.

[Oswald, 1991] Oswald, P. (1991). Approximationsräume und Anwendungen in der Theorie der Multilevel-Verfahren. Manuskript, FSU Jena.

[Oswald, 1992] Oswald, P. (1992). Preliminary results on preconditioners for sparse grid discretizations. FSU Jena.

[Xu, 1990] Xu, J. (1990). *Iterative Methods by Space Decomposition and Subspace Correction: A Unifying Approach.* Report No. AM67, Department of Mathematics, Penn State University. (revised 1992).

[Yserentant, 1986] Yserentant, H. (1986). On the multilevel splitting of finite element spaces. *Numerische Mathematik*, 49, 379–412.

[Zenger, 1991] Zenger, C. (1991). Sparse grids. In W. Hackbusch (Ed.), *Parallel Algorithms for Partial Differential Equations, Proceedings of the Sixth GAMM-Seminar, Kiel, January 19-21, 1990*: Vieweg-Verlag.

[Zhang, 1992] Zhang, X. (1992). *Multilevel additive Schwarz methods.* Report CS-TR-2894, UMIACS-TR-92-52, University of Maryland, Department of Computer Science.