Computing 50, 127–148 (1993)

Computing

© Springer-Verlag 1993
Printed in Austria

Multilevel Gauss-Seidel-Algorithms for Full and Sparse Grid Problems

M. Griebel, C. Zenger, and S. Zimmer, München

Received October 8, 1992

Abstract — Zusammenfassung

Multilevel Gauss-Seidel-Algorithms for Full and Sparse Grid Problems. 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 robust for anisotropic model problems. They can be generalized easily to domain-oriented multilevel methods with the same properties.

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

AMS(MOS) Subject Classifications: 65F10, 65N20, 65N30

Key words: Anisotropic problems, block-Gauss-Seidel iteration, Gauss-Seidel iteration, hierarchical basis, multigrid methods, partial differential equations, point-oriented methods, semidefinite system, sparse grids.

Multilevel Gauß-Seidel-Algorithmen für Voll- und Dünngitterprobleme. Wir stellen gitterorientierte und neu entwickelte punktorientierte robuste Multilevelverfahren für Voll- und Dünngitterdiskretisierungen vor. Besonders die punktorientierten Multilevelmethoden sind sehr gut zu parallelisieren und erweisen sich als robust für anisotrope Modellprobleme. Sie erlauben eine einfache Erweiterung auf gebietsorientierte Multilevelmethoden mit denselben Eigenschaften.

Wir berichten die Ergebnisse numerischer Experimente für die Reduktionszahlen dieser neuen Algorithmen.

1. Introduction

Recently, so-called sparse grid techniques for the solution of elliptic PDEs on rectangular grids have been developed. There, in contrast to standard methods, substantially fewer grid points are needed in the discretization process. Instead of $O(h^{-d})$ grid points for the usual full grid on a domain $\Omega^{(d)} \subset \mathbb{R}^d$, now only $O(h^{-1} \cdot (\log h^{-1})^{d-1})$ grid points are involved, where h denotes the employed mesh size. The obtained accuracy, however, is nearly as good as in the full grid case. In [2] it was shown that the accuracy deteriorates only from $O(h^2)$ to $O(h^2 \cdot (\log h^{-1})^{d-1})$ with respect to the L_2-norm and even remains $O(h)$ with respect to the energy norm, provided that the solution u satisfies the smoothness requirement

$$\frac{\partial^{2d} u}{\partial x_1^2 \ldots \partial x_d^2} \in \mathscr{C}^0(\bar{\Omega}^{(d)}).$$

Using certain product type hierarchical basis functions (HB) for the associated sparse grid finite element space, the Galerkin process leads for linear PDEs 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 [16] 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 [5], [6]) 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 [4], 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, 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 sytsem. Gauss-Seidel type iterations for the semidefinite system are equivalent to certain multigrid methods. For details, see [4].

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 [7], 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 [11, 12] and the ideas of Mulder [14] or Naik and van Rosendale [15].

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. 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 methods for the full grid problem that show convergence properties similar to the respective multilevel methods based on standard coarsening only. Additionally, multigrid 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.

Our experiments show that the reduction rate is independent of the mesh size. However, a rigorous convergence analysis of these point-oriented methods is not yet complete. Note that the efficient implementation of the point-oriented algorithms is not as straightforward as for the conventional multigrid methods. But 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.

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 [7]. 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.

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 $\Omega = (0, 1)^2$,

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

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 corresponding variational problem is to find a function $u \in V$ with

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

(In the case of homogeneous Dirichlet boundary conditions, V is the Sobolev space $H_0^1(\Omega)$.) Here, a: $V \times V \to \mathbb{R}$ is a bounded, V-elliptic, symmetric bilinear form, and $(\cdot, \cdot)$ is the usual linear form for the right hand side. Let $\|\cdot\|_a := \sqrt{a(\cdot, \cdot)}$ denote the induced energy norm. The Lax-Milgram lemma guarantees the existence and uniqueness of the solution of (2). If we consider directly the functional $E(u) = 1/2a(u, u) - (f, u)$, 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 $\Omega_{m,n}$

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

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

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

with dimensions

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

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 $\{x_1, \ldots, x_{N_{m,n}}\}$ in $\Omega_{m,n}$, $m, n = 1, \ldots, k$, 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 $\phi_i^{(m,n)}$, $i = 1, \ldots, N_{m,n}$, that are defined by

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

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 $V_{m,n,x_i} := \operatorname{span}\{\phi_i^{(m,n)}\}$, see Fig. 1. Following [18], the sparse grid can be defined by

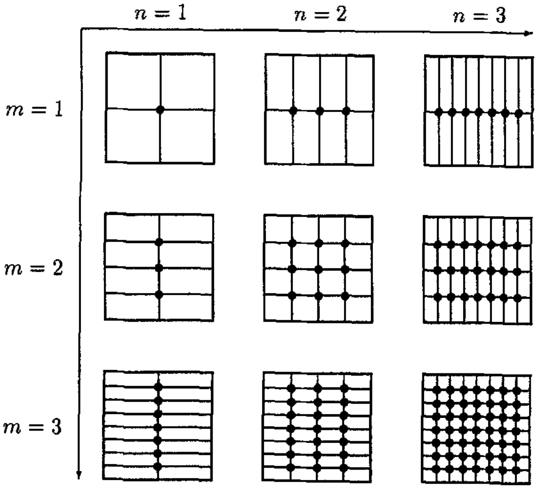

Figure 1. The quadratic scheme of grids $\Omega_{m,n}$, $1 \le m, n \le k$, contained in the full grid, $k = 3$

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

(see Fig. 2), with the corresponding finite element space

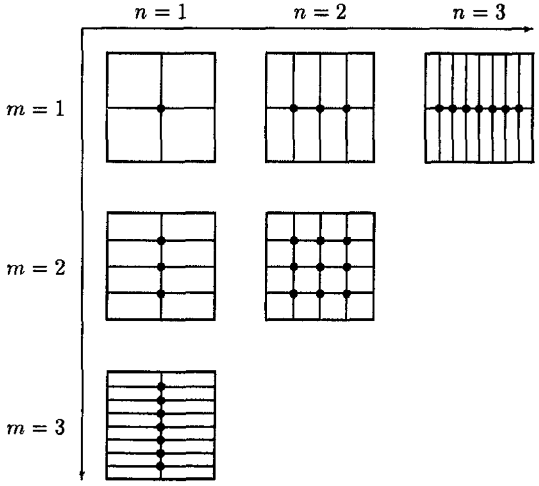

Figure 2. The triangular scheme of grids $\Omega_{m,n}$, $m + n \leq k + 1$, contained in the sparse grid, $k = 3$

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

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 $\dim V_k^S = O(h_k^{-1} \cdot \log(h_k^{-1}))$, while for sufficiently smooth functions 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 [2], [6], [9] and [18].

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 [18], 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

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

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. For example, the linear system resulting from the nodal basis $B_{k,k}$ is not well conditioned and algorithms for its solution have to be accelerated, for example, in the multigrid context by coarse grid correction. Therefore, in the following, we allow, analogously to [4], 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 Fig. 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 (Fig. 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 Fig. 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

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

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

Note that 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 u_k^{HS} 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

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

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

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

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 (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

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

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

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

and, using the hierarchical basis H_k^S, we obtain the subsystem of (31)

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

with

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

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

3. Gauss-Seidel and Block Gauss-Seidel Methods

Now, we use the semidefinite systems (27) and (31) to construct efficient solvers for the full and sparse grid problems, respectively. Here, we will focus on the Gauss-Seidel (GS) and the block Gauss-Seidel (BGS) methods. For BPX-type preconditioned conjugate gradient methods, that can be constructed analogously, see [10]. 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

$$L_k^E = F_k^E + G_k^E + (G_k^E)^T, \tag{35}$$

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

$$W_k^E(u_k^{E,it+1} - u_k^{E,it}) = f_k^E - L_k^E u_k^{E,it} \tag{36}$$

with

$$W_k^E := F_k^E + G_k^E. \tag{37}$$

Note that, despite of the semidefiniteness of L_k^E, its lower diagonal part W_k^E is definite and thus invertible. This can be seen more clearly from the variational formulation (41) below.

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

$$L_k^E = \mathscr{F}_k^E + \mathscr{G}_k^E + (\mathscr{G}_k^E)^T, \tag{38}$$

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

$$\mathscr{W}_k^E(u_k^{E,it+1} - u_k^{E,it}) = f_k^E - L_k^E u_k^{E,it} \tag{39}$$

with

$$\mathscr{W}_k^E := \mathscr{F}_k^E + \mathscr{G}_k^E. \tag{40}$$

Note that $\mathscr{W}_k^E$ is positive semidefinite and can be singular, depending on the respective block partitions. But, like Eq. (27), the Eq. (39) is solvable, because the right hand sides are constructed in a consistent manner, see also the variational formulation (42). Furthermore, all solutions of (39) 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 [17], where we relax a function $u^{(\mu)} \in V_k$ with respect to a $\phi \in E_k$ by

$$u^{(\mu+1)} := u^{(\mu)} + \lambda \cdot \phi \qquad \text{with} \qquad a(u^{(\mu+1)}, \phi) = f(\phi), \tag{41}$$

compare also the PMG method in [13].

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

$$u^{(\mu+1)} := u^{(\mu)} + \sum_{\phi \in \Phi} \lambda_\phi \cdot \phi \qquad \text{with} \qquad \forall \phi \in \Phi \colon a(u^{(\mu+1)}, \phi) = f(\phi), \tag{42}$$

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

$$\forall \phi \in V_k \colon a(e^{(\mu)}, \phi) = f(\phi) - a(u^{(\mu)}, \phi), \tag{43}$$

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

Now, the GS method with respect to the generating system is the cyclic application of (41) for all $\phi \in E_k$ in a given order, while the BGS method applies (42) to all sets of a disjoint decomposition $E_k = \bigcup \Phi_k$ 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 $E_k^{GS} = \bigcup \Phi_k$ 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.

Note, that 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 [4] to implement the matrix-vector multiplication and the Gauss-Seidel iteration in $O(N_k^E)$ and $O(N_k^{ES})$ operations. However, the Gauss-Seidel method is still tricky to implement. A detailed description of implementation details will be given elsewhere. The basic approach follows the idea in [2] and [18]. 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

$$\begin{aligned} \varepsilon^2 \cdot u_{xx} + u_{yy} &= f(x, y), \qquad (x, y) \in \Omega = (0, 1)^2, \\ u(x, y) &= 0, \qquad (x, y) \in \partial\Omega. \end{aligned} \tag{44}$$

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

$$it := \frac{-10}{\log \rho} \tag{45}$$

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

3.1 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 (41), we obtain the hierarchical basis method from [2]. 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 basis instead of the nodal basis on the finest grid. Note further that in contrast to the HB-MG method in [1] our hierarchical basis contains basis functions with in general distorted rectangular support. It is also suitable for the sparse grid case (with $E_k^{GS} := H_k^S$), see [2] for details.

Ordering the hierarchical basis level-wise

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

we traverse the unknowns associated to H_k according to Fig. 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.

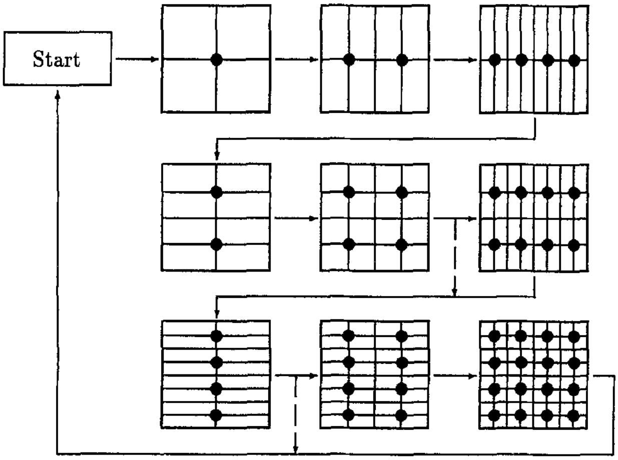

Figure 3. Sequence of relaxations for the hierarchical basis method. The dashed lines mark the short cuts for the sparse grid case

We obtain the algorithm

```
Hierarchical basis method
  for $m = 1 \dots k$
    for $n = 1 \dots k$ (full grid) or $n = 1 \dots k + 1 - m$ (sparse grid)
      for $\phi \in \tilde{B}_{m,n}$
        relax $\phi$ according to (41).
```

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.

Table 1. Reduction rates and numbers of iterations for the HB-Gauss-Seidel, $\varepsilon = 1$

	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

3.2 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 Fig. 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 the algorithm

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

Note the close relation of our algorithm to the methods of Mulder [14], Naik and van Rosendale [15] or Hackbusch [11, 12].

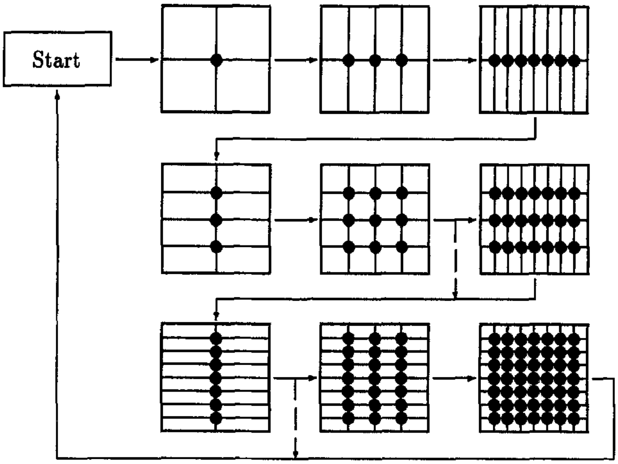

Figure 4. Sequence of relaxations for the multilevel method (outer loop). Again the dashed lines mark the short cuts for the sparse grid case

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

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

denotes the coefficients belonging to $B_{m,n}$. In the sparse grid case, we obtain the levelwise 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 (36)–(37) 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 Fig. 4. For example, a symmetric version of this method is obtained by a sweep as described in Fig. 4 followed by s sweep in reverse ordering of the unknowns, 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 Fig. 4.

Table 2 shows the results for the full grid case. Here, a four color Gauss-Seidel relaxation is applied v 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 of the subproblems arising on each level of the sparse grid discretization for E_k^S. This is the block Gauss-Seidel iteration (39)–(40) 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 $\mathscr{F}$ contain

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

	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, $\varepsilon = 1$

	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	—

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 $V_{m,n}$, $1 \leq m, n \leq k$ 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 $\varepsilon \to 0$ and $\varepsilon \to \infty$, they are still bounded. This demonstrates the robustness of our level-oriented method. Compare also the discussion in [12] and [15].

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

3.3 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 partitioned system. Thus, as blocks in the iteration (39)–(40), we consider the unknowns that belong to the functions

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

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

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

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

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

For the sparse grid case, we define analogously

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

This corresponds to the decomposition

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

and the partition

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

For an illustrating example, see Fig. 5.

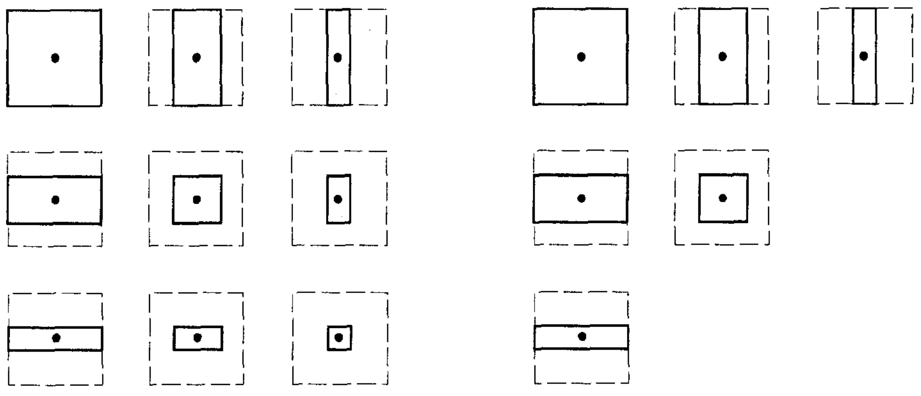

Figure 5. Supports of the generating functions $B_{(1/2,1/2)}$ (left) and $B^S_{(1/2,1/2)}$ (right) for $k = 3$

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 (42) result in systems of linear equations for every grid point $x \in \Omega_{m,n} \backslash (\Omega_{m-1,n} \cup \Omega_{m,n-1})$ 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 $\mathscr{G}_k^E$, or, for the formulation as subspace corrections, by the evaluation of $a(u^\mu, \phi)$ in (42). 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 Fig. 3), and obtain the following algorithm

```
Point-block method
  for $m = 1 \dots k$
    for $n = 1 \dots k$ (full grid) or $n = 1 \dots k + 1 - m$ (sparse grid)
      for $x \in \Omega_{m,n} \backslash (\Omega_{m-1,n} \cup \Omega_{m,n-1})$
        relax $B_x$ or $B_x^S$, respectively, according to (42).
```

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), $\varepsilon = 1$

	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, the numbers for the BGS relaxations indicate that the reduction rates are bounded independently 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 $\varepsilon \to 0$ and $\varepsilon \to \infty$,

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

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

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 it	0.012 5.2	0.022 6.0	0.025 6.2	0.025 6.2	0.024 6.2	0.024 6.2	0.023 6.1	0.022 6.0	0.022 6.0	0.020 5.9	0.012 5.2
Sparse grid	ρ it	0.012 5.2	0.012 5.2	0.018 5.7	0.020 5.9	0.019 5.8	0.019 5.8	0.020 5.8	0.020 5.8	0.020 5.8	0.013 5.3	0.012 5.2

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.

3.4 *Domain-Oriented Methods*

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

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

of the grid points $\Omega_{k,k}$. 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 Ω^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 [3] of the grid points into subdomains is advisable. Compare also Fig. 6. 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.

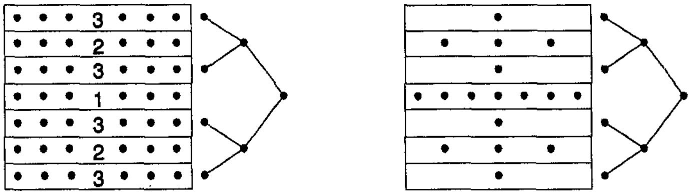

Figure 6. Nested dissection decomposition, full and sparse grid case, $k = 3$

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 Fig. 7. All functions of E_k associated to the grid points of different stripes with the same number are mutually orthogonal with respect to $a(\cdot, \cdot)$, 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 Fig. 7.

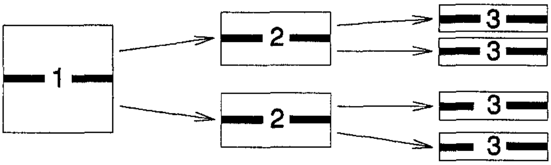

Figure 7. Domain decomposition of the unit square: in each subdomain all basis functions belonging to the centerline are relaxed simultaneously

Table 8 shows the resulting reduction rates for the line-wise BGS method. Once 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 pointwise 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 $v = 1$). Now, the number of operations involved is comparable to the

Table 8. Reduction rates and numbers of iterations for the line-wise BGS, $\varepsilon = 1$

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

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.

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. 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. 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 [7] 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 multigrid-like algorithms. Following [8] and [10], BPX-like additive variants can be constructed following the same principle. 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

[1] Bank, R., Dupont, T., Yserentant, H.: The hierarchical basis multigrid method. Num. Math. *52*, 427–458 (1988).

[2] Bungartz, H.-J.: Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung. Dissertation, Institut für Informatik, TU München, 1992.

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

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

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

[6] Griebel, M.: A parallelizable and vectorizable multi-level algorithm on sparse grids. In: Hackbusch, W. (ed.) Parallel algorithms for partial differential equations. Proceedings of the Sixth GAMM-Seminar, Kiel, January 19–21, 1990. Braunschweig: Vieweg 1991.

[7] Griebel, M.: Grid- and point-oriented multilevel algorithms. In: Hackbusch, W. (ed.) Incomplete decomposition (ILU): Theory, technique and application. Proceedings of the Eigth GAMM-Seminar, Kiel January 24–26, 1992. Braunschweig: Vieweg 1993.

[8] Griebel, M., Oswald, P.: On additive Schwarz preconditioners for sparse grid discretizations. Report Math/92/7, Friedrich-Schiller-Universität Jena, 1992. (Submitted to Numerische Mathematik).

[9] Griebel, M., Schneider, M., Zenger, C.: A combination technique for the solution of sparse grid problems. SFB-Report 342/19/90 A; TUM-I9038, TU München, Institut f. Informatik, 1991.

[10] Griebel, M., Zimmer, S., Zenger, C.: Improved multilevel algorithms for full and sparse grid problems. SFB-Report 342/15/92 A; TUM-I9225, TU München, Institut f. Informatik, 1992.

[11] Hackbusch, W.: A new approach to robust multi-grid solvers. Report, Institut für Informatik und praktische Mathematik, CAU Kiel, 1987.

[12] Hackbusch, W.: The frequency decomposition multi-grid method, part I: Applications to anisotropic equations. Numerische Mathematik *56*, 229–245 (1989).

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

[14] Mulder, W.: A new multigrid approach to convection problems. Report 88-04, CAM, 1988.

[15] Naik, N. H., van Rosendale, J.: The improved robustness of multigrid elliptic solvers based on multiple semicoarsened grids. Report 55–70, ICASE, 1991.

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

[17] Xu, J.: Iterative methods by space decomposition and subspace correction: A unifying approach. Report No. AM67, Department of Mathematics, Penn State University, 1990 (revised 1992).

[18] Zenger, C.: Sparse grids. In: Hackbusch, (ed.) Parallel algorithms for partial differential equations. Proceedings of the Sixth GAMM-Seminar, Kiel, January 19–21, 1990, Braunschweig: Vieweg 1991.

Dr. Michael Griebel
Prof. Dr. Christoph Zenger
Stefan Zimmer
Institut für Informatik
Technische Universität München
Arcisstrasse 21
D-W-8000 München 2
Federal Republic of Germany