Experience with the Solution of a Finite Difference Discretization on Sparse Grids

P. W. Hemker[1] and F. Sprengel[2]

[1] Centrum voor Wiskunde en Informatica,
P.O.Box 94079, NL–1090 GB Amsterdam, The Netherlands
[2] SCAI Institute for Algorithms and Scientific Computing
GMD German National Research Center for Information Technology
Schloss Birlinghoven, D–53754 Sankt Augustin, Germany

Abstract. In a recent paper [10], we described and analyzed a finite difference discretization on adaptive sparse grids in three space dimensions. In this paper, we show how the discrete equations can be efficiently solved in an iterative process. Several alternatives have been studied before in Sprengel [16], where multigrid algorithms were used. Here, we report on our experience with BiCGStab iteration. It appears that, applied to the hierarchical representation and combined with Nested Iteration in a cascadic algorithm, BiCGStab shows fast convergence, although the convergence rate is not truly independent of the meshsize.

1 Introduction

Recently, the use of sparse grids has drawn considerable attention [4,6,7,10,11,16] because of its prospects for a very efficient treatment of higher dimensional problems. Most attention is directed towards the solution of three-dimensional partial differential equations, because of their importance for scientific and technical problems. The contrast of sparse grids with the classical grids is the fact that on usual regular three-dimensional grids the number of gridpoints grows with $\mathcal{O}(h^{-3})$ with decreasing mesh-width h, whereas the number of mesh-points grows with only $\mathcal{O}(h^{-1}|\log h|^2)$ for sparse grids. For a solution, u, with sufficient smoothness, the loss off accuracy (e.g. with piecewise trilinear approximation) is remarkably small. Viz., with bounded mixed derivatives $D^{2,2,2}u$ (at least in the weak sense) the usual accuracy of $\mathcal{O}(h^2)$ reduces to only $\mathcal{O}(h^2|\log h|^2)$.

Here we should notice that the smoothness requirement is essential, and that, with sufficient smoothness, classical higher order methods may yield even more efficiency. As higher order methods can also be used in combination with sparse grids [4], both regular and sparse grids may have their own areas of application. However, it is clear that proper grid-alignment plays a more important role for sparse grids. Therefore, it is useful to see what grids should be used in practice under what circumstances.

Considering the smoothness conditions required for the different approximations, we see that the usual, regular approximations require $u \in C^k(\Omega)$, i.e., all derivatives up to some constant k should be bounded, whereas the error for

L. Vulkov, J. Waśniewski, and P. Yalamov (Eds.): NAA 2000, LNCS 1988, pp. 402–413, 2001.
© Springer-Verlag Berlin Heidelberg 2001

sparse grids is bounded mainly by the mixed derivatives. This implies that the error estimates in the former case are essentially direction-independent, whereas the error for the sparse grid case is dependent on the grid orientation. This may show the area of application of sparse grids: the cases where significant features of the solution can be captured by grid positioning.

We do not want to go into detailed arguments on grid selection. However, we want to say that the study of sparse grids has led to new insights in the proper application of semi-refinement, hierarchical representation of functions, and the use of partially ordered sets of spaces for mesh-adaptive approximation.

This paper concerns the solution of linear systems as they arise in the finite difference approximation of PDEs in 3D. The FD approach to the solution of PDEs on sparse grids was initiated by Griebel in [7] and worked out in more detail in [13]. More results are found in [10], where we described how the finite difference discretization is constructed and how the discrete functions can be represented on a nodal and on a hierarchical basis. Other relevant papers on the solution of 3D discrete systems on sparse grids are [6,11].

The emphasis of this note is on the experience with several solution algorithms for the finite difference discretization on sparse grids. The algorithms are based on a basic iterative solver (BiCGStab [1]) and Nested Iteration. The work is inspired by [12], where hierarchical basis preconditioners in three dimensions are described in a finite element context. The difference is that in [12] a classical sequence of meshes is used, constructed from tetrahedral elements and quasi-uniform refinement. It has been shown that, in that case, the condition of the matrix based on the hierarchical representation, preconditioned by a coarse grid operator is $\mathcal{O}(h^{-1}|\log h|)$, where h is the mesh size. By diagonal scaling by levels, the condition number could be reduced to $\mathcal{O}(h^{-1})$. Similarly, in the present paper, we observe also that the hierarchical representation gives a better convergence rate than the usual nodal representation.

2 Adaptive Function Approximation

For an arbitrary $\mathbf{k} = (k_1, k_2, k_3) \in \mathbb{N}_0^3$, we define a dyadic grid $\Omega_{\mathbf{k}}^+$ over $\Omega \subset \mathbb{R}^3$ by

$$\Omega_{\mathbf{k}}^+ = \{\mathbf{x}_{\mathbf{k},\mathbf{j}} \mid \mathbf{x}_{\mathbf{k},\mathbf{j}} = \mathbf{j} \cdot \mathbf{h}_{\mathbf{k}} = (j_1 2^{-k_1}, j_2 2^{-k_2}, j_3 2^{-k_3})\} \cap \overline{\Omega},$$

and we consider tensor-type basis functions $\varphi_{\mathbf{k},\mathbf{j}}(\mathbf{x}) = \prod_{i=1}^{3} \varphi(x_i/h_{k_i} - j_i)$, where $\varphi(x) = \max(0, 1 - |x|)$ is the usual hat function. Given a continuous function $u \in C(\Omega)$, we can approximate it by $u_{\mathbf{n}} \in V_{\mathbf{n}} = \operatorname{Span}\{\varphi_{\mathbf{n},\mathbf{j}}\}$ by interpolation on $\Omega_{\mathbf{n}}^+$. Obviously, the function $u_{\mathbf{n}}$ on $\Omega_{\mathbf{n}}$ is given by

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

We can make an approximation (1) for all grids $\Omega_{\mathbf{n}}^+$ with $\mathbf{n} \geq \mathbf{0}$. For large enough $\mathbf{n}$, the approximation can be arbitrarily accurate, but the number of degrees of freedom increases geometrically with $|\mathbf{n}| = n_1 + n_2 + n_3$. Therefore, in practice we select a 'smallest' $\mathbf{n}$ such that an accuracy criterion is satisfied. Notice that keeping the representations in *all* coarser $V_\mathbf{k}$ (all $V_\mathbf{k}$, $\mathbf{0} \leq \mathbf{k} \leq \mathbf{n}$) does not take essentially more coefficients than the representation on the finest grid (i.e., in $V_\mathbf{n}$) alone.

In order to obtain an efficient approximation, we can distinguish different areas in the domain Ω, in each of which we make the finest approximation of u in different $V_\mathbf{n}$. We make full and efficient use of the system $\{V_\mathbf{n} \mid \mathbf{n} \in \mathbb{N}_0^3\}$, by *in principle* approximating a given function $u \in C(\Omega)$ in *all* $\{V_\mathbf{n} \mid \mathbf{n} \in \mathbb{N}_0^3\}$, but using *in practice* only those coefficients that contribute to a sufficiently accurate representation. This implies that in practice the function u is represented in a particular $V_\mathbf{n}$ only on part of the domain Ω. To introduce a (minimal) structure in the family of approximating basis functions $\{\varphi_{\mathbf{n},\mathbf{j}}\}$, we introduce the following condition **H.** (*The H condition:*) If a basis function $\varphi_{\mathbf{n},\mathbf{j}}(\mathbf{x})$ is used in the representation (1), then all corresponding coarser basis functions (i.e., functions $\varphi_{\mathbf{k},\mathbf{i}}$ for which $\mathrm{supp}(\varphi_{\mathbf{k},\mathbf{i}}) \supset \mathrm{supp}(\varphi_{\mathbf{n},\mathbf{j}})$) are also used for the representation.

E- and H-Representation. We call the representation of the approximation of a function $u \in C(\Omega)$ by a collection of such (partial) approximations (1) in the family of spaces $\{V_\mathbf{n}\}$, the *nodal representation*, or the *E-representation* of the approximation. This E-representation requires the coefficients $a_{\mathbf{n},\mathbf{j}} = u(\mathbf{x}_{\mathbf{n},\mathbf{j}})$ corresponding with grid-points $\mathbf{x}_{\mathbf{n},\mathbf{j}}$, to be equal on the different grids $\Omega_\mathbf{n}^+$ at coinciding grid-points $\mathbf{x}_{\mathbf{n},\mathbf{j}}$. Thus, because points from coarser grids coincide with those from finer ones, a certain consistency is required (and a redundancy exists) in the E-representation of an approximation.

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

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

In this case, of course, the set of coefficients $\{a_{\mathbf{n},\mathbf{j}}\}$ always determines a unique function u_h. However, for a given function u_h, now the coefficients $\{a_{\mathbf{n},\mathbf{j}}\}$ are not uniquely determined because the $\{\varphi_{\mathbf{n},\mathbf{j}}\}$ are linearly dependent. One way to select a special unique representation is by choosing the coefficients $a_{\mathbf{n},\mathbf{j}}$ such that $a_{\mathbf{n},\mathbf{j}} \neq 0$ only for those $(\mathbf{n},\mathbf{j})$ for which $|\!|\!|\mathbf{j}|\!|\!| = j_1 \cdot j_2 \cdot j_3$ is odd[1]. This implies that $a_{\mathbf{n},\mathbf{j}} = 0$ except for a pair $(\mathbf{n},\mathbf{j})$ for which $\Omega_\mathbf{n}^+$ is the coarsest grid which contains the nodal point $\mathbf{x}_{\mathbf{n},\mathbf{j}}$. This representation

$$u_h = \sum_{(\mathbf{n},\mathbf{j}),|\!|\!|\mathbf{j}|\!|\!| \text{ odd}} a_{\mathbf{n},\mathbf{j}} \varphi_{\mathbf{n},\mathbf{j}} \tag{2}$$

we call the *H-representation* because it represents the approximation in the *hierarchical basis*

$$\{\varphi_{\mathbf{n},\mathbf{j}} \mid \mathbf{n} \in \mathbb{N}_0^3, \mathbf{j} \in \mathbb{Z}^3, \ |\!|\!|\mathbf{j}|\!|\!| \text{ odd}, \mathbf{x}_{\mathbf{n},\mathbf{j}} \in \Omega_\mathbf{n}^+\} \ , \tag{3}$$

[1] More precisely, with "$|\!|\!|\mathbf{j}|\!|\!|$ is odd" we mean: for all $i = 1, 2, 3$, either j_i is an odd integer, or $k_i = 0$ (i.e., j_i lives on the coarsest grid in the i-direction).

and the part of u_h in

$$W_{\mathbf{n}} = \operatorname{Span}\{\varphi_{\mathbf{n},\mathbf{j}} \mid \mathbf{j} \in \mathbb{Z}^3_, \ |||\mathbf{j}||| \text{ odd}, \mathbf{x}_{\mathbf{n},\mathbf{j}} \in \Omega^+_{\mathbf{n}}\}$$

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

$$V_{\mathbf{n}} = W_{\mathbf{n}} + \sum_{j=1}^{3} V_{\mathbf{n}-\mathbf{e}_j} = \sum_{0\leq\mathbf{m}\leq\mathbf{n}} V_{\mathbf{m}} \,,$$

and the sparse grid space is defined by

$$V_L = \sum_{0\leq|\mathbf{m}|\leq L} V_{\mathbf{m}} \,,$$

corresponding to a sparse grid $\Omega^+_L = \bigcup_{0\leq|\mathbf{m}|\leq L} \Omega^+_{\mathbf{m}}$. Interpolating the function u at the nodal points $\mathbf{x}_{\mathbf{n},\mathbf{j}}$, the hierarchical coefficients $a_{\mathbf{n},\mathbf{j}}$ in

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

are determined by (cf. [9])

$$a_{\mathbf{n},\mathbf{j}} = \prod_{i=1}^{3} \left[-\frac{1}{2}, 1, -\frac{1}{2}\right]_{h_{n_i}\mathbf{e}_i} u(\mathbf{j}\mathbf{h}_{\mathbf{n}}) \,,$$

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

For piecewise multilinear functions, it is often described [5,6,7] how a pyramid algorithm can be used to convert an E-representation to a H-representation, and vice versa. Such a conversion can be executed in $\mathcal{O}(N)$ operations, where N is the total number of coefficients.

The Data Structure. The data structure to implement all the above possibilities of an adaptive (sparse) grid representation can be efficient and relatively simple. For the d-dimensional case ($d = 1, 2, 3$), we use the data structure BASIS3 [8] that takes the 'patch' $P_{\mathbf{n},\mathbf{j}}$ as an elementary entity. This $P_{\mathbf{n},\mathbf{j}}$ takes all information related to a right-open left-closed cell

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

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

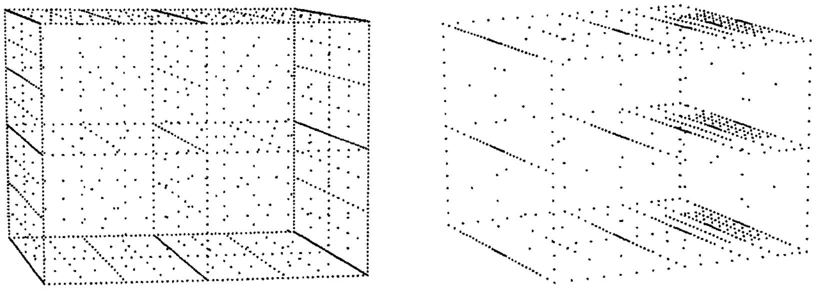

Fig. 1. Regular sparse grid Ω_6^+ for $\Omega = (0,1)^3$ (left) and an adaptive sparse grid (ASG) (right)

3 Difference Operators for ASG Functions

Although finite element discretization of a PDE on a sparse grid is feasible for a constant coefficient problem in two dimensions, finite elements for more-dimensional problems and variable coefficients give problems. The difficulty arises because — with the hierarchical basis (3) for test and trial space — the computational complexity of the evaluation of the discrete operator becomes too large. This is caused by the fact that the intersection of the supports of an arbitrary trial and test function is much smaller than the supports of these functions themselves. This has as a consequence that the advantage of sparse grids is lost if the FEM discrete operator is evaluated.

The alternative, as it was already suggested in [7,13], is the use of a finite difference discretization. Therefore, in order to solve PDEs on sparse grids, we should be able to apply (approximate) differentiation to discrete representations of approximations as described in [10]. The application of linear difference operators of the form

$$L_h u_h \equiv \sum_{i,j} \frac{\partial}{\partial x_i}\left(A_{ij}(\mathrm{x})\frac{\partial}{\partial x_j}\right) u_h(\mathrm{x}) + \sum_i B_i(\mathrm{x})\frac{\partial}{\partial x_j} u_h(\mathrm{x}) + C(\mathrm{x})u_h(\mathrm{x}) \quad (4)$$

comes down to the construction of linear combinations, the pointwise multiplication, and the differentiation of functions (2). In both representations the construction of a linear combination over the real numbers is directly computed by application of the linear combination to the coefficients. Pointwise multiplication is only possible in the E-representation, in which the function values at grid-points are directly available. For a description of the evaluation of first and second order derivatives we again refer to [10].

First and Second Order Interpolation. Because we use piecewise tri-linear basis functions $\varphi_{\mathrm{nj}}(\mathrm{x})$ on the grid Ω_{n}^+, truncating at a particular level corresponds with tri-linear interpolation between the nodal points included. In this way, piecewise tri-linear interpolation is natural in the finite hierarchical representation.

For $C^{2,2,2}(\Omega)$-functions, the behavior of the coefficients $a_{\mathbf{nj}}$ is rather predictable for higher levels of approximation because Lemma [9, Lemma 3.2]. gives a precise relation with the second order cross derivatives, or in lower dimensional manifolds (at the coarsest level, at the boundaries, or in mixed H-E-representations over the different coordinate directions) with the second order derivatives. This allows for an efficient quadratic interpolation procedure when a finite hierarchical representation of a discrete function is available. To interpolate the function

$$u_h^\ell(\mathbf{x}) = \sum_{|\mathbf{n}|\le\ell} \sum_{\mathbf{j},\|\mathbf{j}\|\ \mathrm{odd}} a_{\mathbf{n},\mathbf{j}}\varphi_{\mathbf{n},\mathbf{j}}(\mathbf{x})\,. \tag{5}$$

with second order accuracy to a function $u_h^{\ell+1}(\mathbf{x})$, the coefficients $\{a_{\mathbf{n},\mathbf{j}} \mid |\mathbf{n}| = \ell+1\}$ can be derived from the coefficients $\{a_{\mathbf{n},\mathbf{j}} \mid |\mathbf{n}| = \ell\}$ by taking the new coefficients $a_{\mathbf{m},\mathbf{k}} = a_{\mathbf{n},\mathbf{j}}/4$, where $|\mathbf{m}| = |\mathbf{n}|+1$ and $\mathbf{m}$ and $\mathbf{j}$ satisfy $|\mathbf{x}_{\mathbf{m},\mathbf{k}} - \mathbf{x}_{\mathbf{n},\mathbf{j}}| \le 2^{-\ell}$. This corresponds with the extrapolation assumption that the second order derivative is slowly varying (constant) over the smallest covering cell $\Omega_{\mathbf{n},\mathbf{j}}$. In order to maintain symmetry over the coordinate directions, in the case of a non-unique smallest covering cell one may take the mean value of the coefficients of all (at most $d-1$) smallest covering cells. In this way, we introduce the second order interpolation operator $P_{\ell+1,\ell}$, defined by

$$u_h^{\ell+1} = P_{\ell+1,\ell}\, u_h^\ell\,, \tag{6}$$

where both u_h^ℓ and $u_h^{\ell+1}$ are described by (5). First order interpolation is simply achieved by setting $a_{\mathbf{m},\mathbf{k}} = 0$ for $|\mathbf{m}| = |\mathbf{n}| + 1$.

4 Solution of the Finite Difference Discretization for the Laplacian

In the remaining part of this paper, as an example of (4), we solve the discretized operator equation as it was described in detail in [10]. For simplicity, we restrict ourselves to the model problem of Poisson's equation with homogeneous Dirichlet boundary conditions,

$$\begin{aligned} -\Delta u &= f \qquad \text{in } \Omega, \\ u|_{\delta\Omega} &= 0, \end{aligned} \tag{7}$$

on the cube $\Omega = (0,1)^3$ and a regular sparse grid.

Iteration Based on a Galerkin Relation. In [10], an analysis of the discretization was made and multilevel-type algorithms, based on the Galerkin structure of the equations were proposed. The coarse grid operators involved were no longer finite difference operators. In an obvious way, the Galerkin relations lead to iterative (defect correction) solution algorithms that are applied in a multilevel setting. However, no spectral equivalence could be established, and the convergence of the iterative schemes appears to depend on the maximum discretization level used, so that the convergence rate slows down on finer grids. The algorithm is briefly characterized in Figure 2 (for details see [10]). Applied to the 3D-problem (7) with the right-hand side $f(\mathbf{x}) = -3\pi^2(\prod_{i=1}^3 \sin \pi x_i + 8\prod_{i=1}^3 \sin 8\pi x_i)$ and starting from the zero function $u_L^{(0)} \equiv 0$, we obtain the convergence behavior shown in Figure 3. We see that we get better convergence if we include also lower levels (right). In both cases, however, the speed of convergence slows down with growing levels. Approximately, the reduction factor gets worse with L^2, the square of the highest level. The slow convergence motivates us to see if better convergence could be obtained by cascadic iteration.

```
for $\ell$ from $L_0$ to $L$
do  for $i = 1$ to $\nu$
    do  for all $|\mathbf{n}| = \ell$
        do  $u_h := u_h + P_{L,\mathbf{n}}\ L_{\mathbf{n}}^{-1} \tilde{R}_{\mathbf{n},L}\ (f_h^L - L_h^L u_h)$ enddo        (G)
    enddo
enddo
```

Fig. 2. The Galerkin algorithm (G)

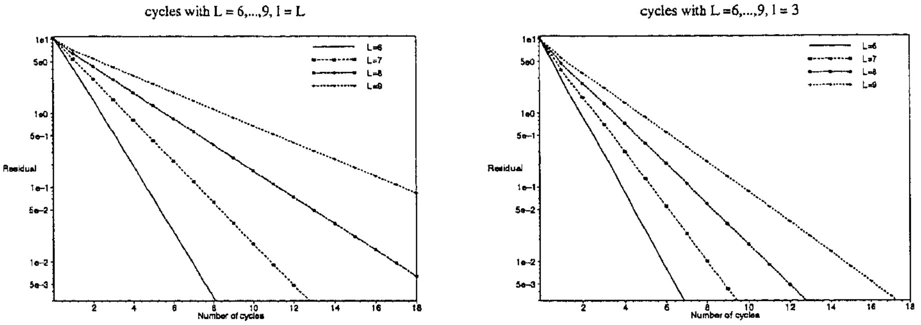

Fig. 3. Left: Convergence of Algorithm (G) for the levels $L = L_0 = 6, \ldots, 9$. Right: Convergence of Algorithm (G) for the levels $L = 6, \ldots, 9$, $L_0 = 3$ with $\nu = 1$

Cascadic Iteration. By construction, the sparse grids and the sparse grid spaces are provided with a multilevel structure, i.e., $\Omega_\ell^+ \subset \Omega_{\ell+1}^+$ and $V_\ell \subset V_{\ell+1}$. Moreover, in [10], we could prove a Galerkin relation

$$L_h^\ell = R_{\ell,\ell+1}\, L_h^{\ell+1}\, \tilde{P}_{\ell+1,\ell}$$

for the discrete Laplace operator L_h^ℓ in hierarchical representation. Here, $R_{\ell,\ell+1}$ denotes the natural hierarchical restriction and $\tilde{P}_{\ell+1,\ell}$ is the first order interpolation. This will be used in a cascadic iteration.

In [2,3], Bornemann and Deuflhard proposed the cascadic multigrid method. In this method, a solution is computed by nested iteration on a sequence of refining grids, without coarse grid corrections applied on the finer grids. In cascadic MG, more basic iterations are used on the coarser than on the finer levels. It has been proved [3,14] that cascadic MG applied to a FEM discretization using P1-conforming elements for the second order 3D problem is accurate with an optimal computational complexity for all conventional iterative methods, like Jacobi or Gauss–Seidel iteration, as well as for the conjugate gradient method as a smoother. However, in the 2D case the cascadic MGM gives accurate solution with optimal complexity for the CG method, but only nearly optimal complexity for the other conventional smoothers.

In [15], it is shown that that this is also true for other conforming or nonconforming elements, provided that $m_l \geq \beta^{L-l} m_L$, with m_l the number of iterations on level l and some constant β depending on the relaxation method.

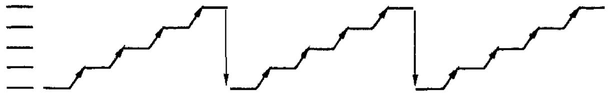

Fig. 4. Cascadic iteration: the problem is approximately solved on a coarser (lower) grid before interpolation to a finer (higher) grid is made. The cycle over all levels is repeated in an outer defect correction (iterative refinement) process. The levels used are the union of the grids $\Omega_{\mathbf{n}}$, with $|\mathbf{n}| = k$, $k = 1, 2, \ldots, 10$. The number of points at each level is given in Table 1

For iteration, we use a cascadic application of the BiCGStab algorithm [1] for the solution of $L_h^L u_h = f_h^L$. The algorithm is shown in Figure 5. In the algorithm $R_{\ell,L}$ denotes the natural hierarchical restriction and $P_{\ell+1,\ell}$ is the first order prolongation $\tilde{P}_{\ell+1,\ell}$ or the second order prolongation (6). Computations are made with this algorithm on meshes up to 10 levels. The corresponding number of gridpoints is given in Table 1.

Table 1. The number of points on the different levels

levels k:	0	1	2	3	4	5	6	7	8	9	10
points #:	8	44	158	473	1286	3302	8170	19699	46594	108568	249910

The working horse of the solution algorithm is BiCGStab iteration. Because of the non-sparse structure of the matrix representation of the sparse grid dis-

```
until a convergence criterion is satisfied
do  $f_h^* := f_h^L - L_h^L u_h$
    $c_h := 0$
    for $\ell$ from $L_0$ to $L$
    do  int $i$=0, $i_c$=0;
        until $i > i_{\max}$ do
            real $n$,$\rho$,$\beta$,$\omega$, $\alpha = 0.0$, $\rho_0 = 1.0$, $\omega_0 = 1.0$;
            $r_h := R_{\ell,L} f_h^* - L_h^\ell c_h$
            $n = (r_h, r_h)$
            if j=0 then $n_0 = n$ endif
            if $n < \varepsilon$ then break endif
            $v_h = 0$
            $p_h = v_h$
            $\tilde{r}_h = r_h$
            until $i > i_{\max}$ do
                $\rho = (\tilde{r}_h, r_h)$
                if $|\rho| < \varepsilon$ then break endif
                $\beta = (\rho/\rho_0)(\alpha/\omega_0)$
                $p_h = r_h + \beta(p_h - \omega v_h)$
                $v_h = L_h^\ell p_h$
                $d = (\tilde{r}_h, v_h)$
                if $|d| < \varepsilon$ then $d = 1.0$ endif
                $\alpha = \rho/d$
                $r_h = r_h - \alpha v_h$
                $t_h = L_h^\ell r_h$
                $d = (t_h, t_h)$
                if $|d| < \varepsilon$ then break endif
                $\omega = (t_h, r_h)/d$
                $c_h = c_h + \alpha p_h + \omega r_h$
                $r_h = r_h - \omega t_h$
                $\rho_0 = \rho$
                $\omega_0 = \omega$
                $n = (r_h, r_h)$
                if $|\omega| < \varepsilon$ then break endif
                if $i_c > i_{c,\max}$ then $i_c$=0; break; endif
                $i$=$i$+1
            enddo
        enddo
        $c_h := P_{\ell+1,\ell}\, c_h$
    enddo
    $u_h := u_h + c_h$
enddo
```

Fig. 5. The cascadic iteration algorithm with BiCGStab

crete operators, we are only interested in matrix-free methods. This restricts the choice of the applicable preconditioning methods. In fact, for preconditioning we restrict ourselves to diagonal scaling and transformation between E- and H- representation. We exploit the available hierarchical structure of the approximate solution by the computation of a good initial approximation on a given level by interpolation of a sufficiently accurate solution that is computed on a coarser level. Thus, starting from a coarsest grid, we obtain the cascadic algorithm.

First, the algorithm was applied both to the E-representation and to the H-representation of the solution, and it appeared that the solution of the H-representation is much faster. This is in agreement with the findings of Ong [12] for the solution of a FEM discretization with the tetrahedral element and quasi-uniform refinement, as discussed in the introduction. As a consequence we further only considered iteration with the H-representation.

By itself the BiCGStab is not a very efficient solver, but combined with cascadic switching between the levels we obtain an algorithm that solves the equation up to truncation error accuracy in only a few (outer) cycles. This is shown in the Figures 6 and 7. In the Fig. 6, we see the difference between using a large number of (inner) BiCGStab iterations vs using a small number. In Fig. 7, on level 10, we see the difference between the use of the first order prolongation $\tilde{P}_{\ell+1,\ell}$ (left) or the second order formula (6) (right). We clearly see that second order interpolation gives a much better convergence, so that truncation error accuracy is obtained in a small number of (4) outer iteration cycles.

Legend to Figures 6 and 7. Top figures: the logarithm of the two-norm of the measured residual at different levels and in the inner loop, against the number of inner iterations. Bottom figures: logarithm of the residual and the global discretization error of the solution of the target equation against the number of elementary operations (flops). The constant lines indicate the approximation error and the local truncation error.

5 Conclusion

Because the evaluation of finite element stiffness matrices for variable coefficient equations on sparse grids in three dimensions still yields difficulties, finite differences are an interesting alternative instead. In this paper, we show how a cascadic multigrid application of BiCGStab yields an efficient solution method for the resulting discrete equations.

The method applies the BiCGStab-iteration to the H-representation of the discrete solution, it uses second order interpolation between the different levels of discretization and it applies global defect correction (iterative refinement) as an outer iteration cycle. Results for this solution method are presented which show that 3 or 4 iteration cycles may be sufficient to solve the discrete equations up to local truncation error accuracy.

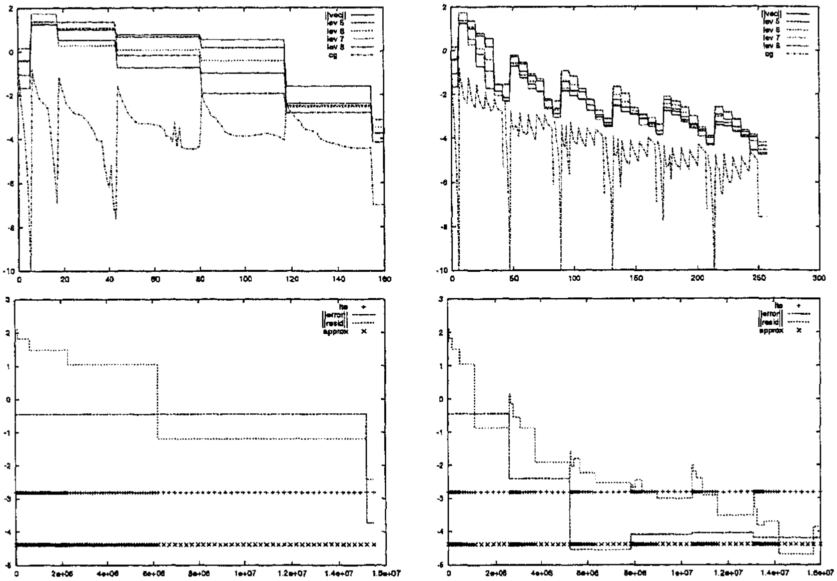

Fig. 6. The advantage of spreading inner iterations over more outer iterations. Left: a single outer iteration with 36 inner iterations at each level. Right: 6 outer iterations with 6 inner iterations each

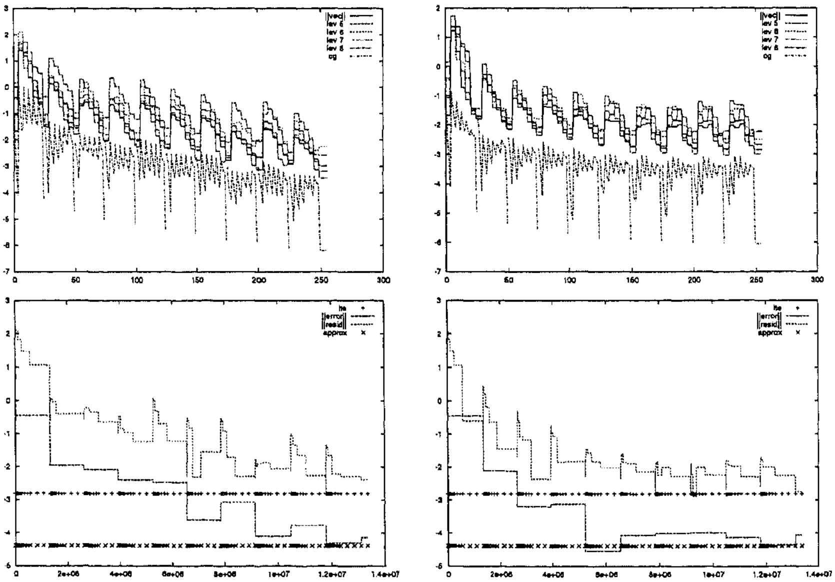

Fig. 7. Convergence at level $k = 10$. Left: first order interpolation between the levels. Right: second order interpolation

References

1. R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the solution of linear systems: building blocks for iterative methods.* SIAM Books, Philadelphia, 1994.
2. F. Bornemann and P. Deuflhard. The cascadic multigrid method for elliptic problems. *Numerische Mathematik*, 75:135–152, 1996.
3. F. Bornemann and P. Deuflhard. The cascadic multigrid method. In R. Glowinsk, J. PÄéiaux, Z. Shi, and O. Widlund, eds., *The 8th International Conference on Domain Decomposition Methods for Partial Differential Equations.* John Wiley & Sons Ltd., 1997.
4. H.-J. Bungartz and T. Dornseifer. Sparse grids: Recent developments for elliptic partial differential equations. In W. Hackbusch and G. Wittum, eds., *Multigrid Methods V*, Lecture Notes in Computational Science and Engineering 3. Springer, Berlin, 1998.
5. M. Griebel. A parallelizable and vectorizable multi-level algorithm on sparse grids. In *Parallel algorithms for partial differential equations*, Notes Numer. Fluid Mech. 31, 94–100. Vieweg, Wiesbaden, 1991.
6. M. Griebel. *Multilevelmethoden als Iterationsverfahren über Erzeugendensystemen.* Teubner Skripten zur Numerik. Teubner, Stuttgart, 1994.
7. M. Griebel. Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences. *Computing*, 2000. to appear.
8. P. W. Hemker and P. M. de Zeeuw. BASIS3, a data structure for 3-dimensional sparse grids. In H. Deconinck, ed., *Euler and Navier-Stokes Solvers Using Multi-dimensional Upwinds Schemes and Multigrid Acceleration*, Notes Numer. Fluid Mech. 57, 443–484. Vieweg, Wiesbaden, 1997.
9. P. W. Hemker and C. Pflaum. Approximation on partially ordered sets of regular grids. *Appl. Numer. Math.*, 25:55–87, 1997.
10. P. W. Hemker and F. Sprengel. On the representation of functions and finite difference operators on adaptive sparse grids. submitted (CWI Report MAS-R9933), 1999.
11. J. Noordmans and P. W. Hemker. Convergence results for 3D sparse grid approaches. In *PRISM'97, Proceedings of the Conference on Preconditioned Iterative Solution Methods for Large Scale Problems in Scientific Computations*, 11–22, Nijmegen, The Netherlands, 1997. Nijmegen University.
12. M. E. Ong. Hierarchical basis preconditioners in three dimensions. *SIAM Journal on Scientific Computing*, 18:479–498, 1997.
13. T. Schiekofer. *Die Methode der Finiten Differenzen auf dünnen Gittern zur Lösung elliptischer und parabolischer partieller Differentialgleichungen.* PhD thesis, Universität Bonn, 1998.
14. V. Shaidurov. Some estimates of the rate of convergence for the cascadic conjugate gradient method. *Comp. Math. Appls*, 31:161–171, 1996.
15. Z.-C. Shi and X. Xu. Cascadic multigrid method for elliptic problems. *East-West J. Numer. Math*, 7:199–209, 1999.
16. F. Sprengel. Some remarks on multilevel algorithms for finite difference discretizations on sparse grids. CWI report MAS-R9924, 1999.