Multigrid Methods for Hierarchical Adaptive Finite Elements

S.-A. H. Schneider
Chr. Zenger

Institut für Informatik
Technische Universität München
D-80290 München, Germany
e-mail: {schneids,zenger}@in.tum.de

January 28, 2000

Abstract

The hierarchical tensor product finite element method, using elements with variable aspect ratio, has proven to be useful for the numerical solution of elliptic partial differential equations and allows several concepts for the control of the adaptation process. In addition to the more conventional adaptive grids derived from L_2- or H^0-based error estimates, we study grids which are optimized with respect to the evaluation of linear functionals like the value of the solution at a fixed point. It is well-known that this requires the solution of a dual problem. As for the case of singular solutions, these grids are extremely refined at certain points, yielding different strategies for the solution vector and the right-hand side. This improves the order of the respective error in regard to L_2- or H^0-based adaptive grids, but causes additional difficulties for the design of efficient multigrid solvers.

Key words. self-adaptive solution, a posteriori error estimation, hierarchical finite elements, multigrid method, sparse grids, tensor product approach

AMS(MOS) subject classifications. 65N22, 65N30, 65N50, 65N55

1 Introduction

The first known mathematician to use hierarchical ideas was Archimedes $\mathit{Τετραγωνισμός}$ $\mathit{παραβολῆς}$ (the quadrature of the parabola), see [13]. By inductively exhausting the parabola with triangles, he was able to measure the area given by a parabola. In 1909, Faber [7] introduced the hierarchical basis and explicitly used it for the representation of functions. Yserentant [15] applied the hierarchical basis in 1986 as a preconditioner. In 1990, Zenger [16] directly represented a smooth multivariate function u with a hierarchical tensor product basis instead of a standard nodal basis. The coefficients of this representation, the so-called *hierarchical surplusses*, decrease with the volume of the support of the corresponding basis functions. Consequently, the hierarchical surplus is a very simple criterion for the decision of whether the contribution to the basis representation is important enough or not. These considerations lead to the concept of *sparse grids* in which we order the basis functions in terms of their contribution to the basis representation and, with that, in terms of their support volume. It turns out that sparse grids are a priori L_2- or H^0-adaptive grid structures and lessen the so-called "curse of dimension", see Bungartz [4]. To understand this, let us compare the number of grid points that are necessary to reduce, e.g., the L_2-error of a linear finite element discretization by a factor $1/4$ for a sufficiently smooth problem. In a standard nodal approximation space, we asymptotically need 2^d and in the sparse grid approximation only twice – independent of the dimension d – as many grid points.

Often we are not only interested in the L_2- and H^0-error, but also in other interesting quantities defined by the special application or prescribed by the user, such as the value at a certain point. Thus, it is natural to ask how we can discretize the given problem with a minimum amount of work and memory, on one hand, and ensure the prescribed tolerance, on the other hand. In fact, this is an economic question: efficient use of limited sources (computational time and memory) with respect to the benefit gained (user prescribed tolerance). In non-adaptive methods, the solution has to be estimated before its calculation in order to arrange the grid and to cluster grid points where necessary. We then get a priori optimized grid structures, e.g., the aforementioned sparse grids. However, the a priori grid construction might be a hard job for arbitrary quantities of interest, for, in general, the locations of critical ranges are not known a priori. Often, much experience

is required. A priori estimations as provided, e.g., by the standard error analysis for finite element or finite difference methods, are often insufficient since they only describe the asymptotic error behaviour. Additionally, they require regularity conditions of the solution which are not satisfied in the presence of the singularities, e.g., arising from re-entrant corners. These considerations clearly motivate a *self-adaptive solution process* based on an *a posteriori error estimation*.

Today *multigrid methods* are thought to be the most efficient methods to solve the large systems of linear equations arising in connection with the approximate solution of linear elliptic boundary value problems by finite element or finite difference methods. Griebel and Oswald [8] described an additive Schwarz multilevel preconditioner, Bungartz [3] and Pflaum [10] multiplicative multigrid solvers for sparse grids.

Following [14], one of the most promising and most challenging methodological approaches in today's scientific computing is to bring together multigrid, parallelism and adaptivity methods. In this article, we specially focus on the fruitful combination of the multigrid and the adaptivity aspects. Here, in many respects, the hierarchical polynomial basis of degree $p \geq 1$ has proven to be useful. We introduce a multiplicative multilevel solver for more general adapted grids, e.g. grids which are optimized with respect to the evaluation of the solution at a fixed point. In Section 2, we explain the hierarchical finite elements used for the Galerkin approximation and the adaptation process. In Section 3, we show how the *adaptation* process is done and to which grids the refinement process leads to. Hereby, the dual solution enables us to decompose the discretization error into different contributions arising from the discretization. We estimate these error contributions and seek to balance their influences. For a detailed description of the adaptive hierarchical finite element method we refer to Schneider [12]. We use an a posteriori error estimation, because then, no information about the exact solution is needed before the computation. The a posteriori error estimation is based on a concept in finite element Galerkin methods in which the discretization error is essentially governed by the Green's function. This concept uses duality arguments introduced by Eriksson, Johnson, and their co-workers [6, 5] and further developed by Rannacher and co-workers [2, 11]. A new point of view to define a *regular sparse grid* is given in Section 4. We demonstrate that sparse grids can be classed as a special case of this self-adaptive solution process in which we refine as much as possible. Section 5 deals with

the ingredients of the *multigrid method*. The main idea is to relax only on the discretization points of all possible tensor subgrids of the discretization to get N-independent convergence rates, where N denotes the number of degrees of freedom of the underlying finite element approach. The lion's share of this article is Section 6, where two- and three-dimensional *numerical examples* are presented and discussed in detail. For example, we observe an $O(N^{-2})$ convergence behaviour for the two-dimensional point-value error problem discretized with linear finite elements. This means that we gain one order of magnitude. Another basic result is that the order of magnitude of the computational time is dominated by the calculation of the right-hand side, and not, as one would expect, by the solution of the system of linear equations. Several *extensions* are presented in Section 7: a problem with a root singularity on the boundary is solved with higher order discretizations. Finally, *concluding remarks* are given in Section 8.

2 Hierarchical Finite Elements

Beginning with the one-dimensional case, we construct the *hierarchical basis* B_n of *depth* n for the interval $\Omega^{(1)} := [0,1]$ from the standard hat function $\phi : \mathbb{R} \to \mathbb{R}$,

$$\phi(x) := \begin{cases} 1 - |x| & \text{for } x \in [-1,+1], \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

and the linear transformation $\tau_{x_j} : [x_j - h_{x_j}, x_j + h_{x_j}] \to [-1,1]$ defined by $\tau_{x_j}(x) := (x - x_j)/h_{x_j}$. All piecewise linear basis functions $\phi_{x_j} \in B_n$ can then be constructed by dilation and translation of ϕ

$$\phi_{x_j}(x) := \phi\left(\tau_{x_j}(x)\right) \quad \forall\, x \in [x_j - h_{x_j}, x_j + h_{x_j}],$$

with *support* $\text{supp}\ \phi_{x_j} := \left[\max\{0, x_j - h_{x_j}\}, \min\{x_j + h_{x_j}, 1\}\right] \subset [0,1]$ for certain given discretization points $x_j \in \Omega^{(1)}$ and the corresponding grid width $0 < h_{x_j} \in \mathbb{R}$. We call $x_j \in [0,1]$ the *basis point* $bp(\phi_{x_j})$ of the basis function $\phi_{x_j} \in B_n$.

Let us construct the hierarchical basis B_n inductively, starting with $B_0 := \{\phi_0(x) := 1 - x,\ \phi_1(x) := x\}$, where we define $h_0 := h_1 := 1$, see Fig. 1, by two principles:

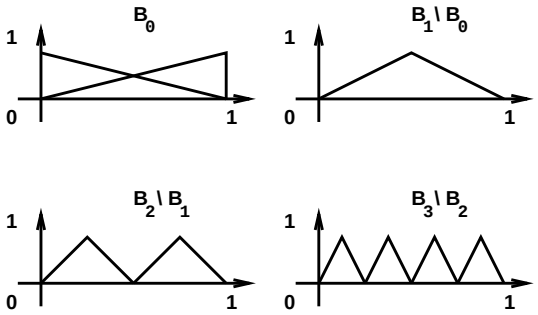

Figure 1: The one-dimensional piecewise linear hierarchical basis: basis functions of the bases $B_0, \ldots, B_3$.

1. the *principle of hierarchy* (P1): the family of bases B_n $\forall n \geq 0$ build a nested sequence of sets by $B_{n-1} \subset B_n$ $\forall n > 0$ and

2. the *principle of surplus* (P2): all basis functions of B_n $\forall n > 0$ may not influence the representation of the function $u \in H^1\left(\Omega^{(1)}\right)$ in any basis point $x_j \in bp(B_{n-1})$ of the basis functions of B_{n-1}.

The second principle gives a hint how to construct the basis functions of B_n $\forall n > 0$. All basis functions $\phi_{x_j} \in B_n \setminus B_{n-1}$ have to fit with their support exactly in the set of intervals one gets by the partition of $\Omega^{(1)}$ by the basis points x_j of the basis functions $\phi_{x_j} \in B_{n-1}$.

In Fig. 1, the first three steps of constructing B_n are given. For example, one deduces the only basis function $\phi_{1/2} \in B_1 \setminus B_0$ by exhausting the interval $[0,1]$ by the support supp $\phi_{1/2}$ of the function $\phi_{1/2}(x) := 1 - |2x - 1|$ with $h_{1/2} := 1/2$. It is easy to deduce that for all basis functions $\phi_{x_j} \in B_n \setminus B_{n-1}$ $\forall n \geq 1$ hold for the grid width $h_{x_j} := 2^{-n}$, and therefore, the support of the basis function ϕ_{x_j} has the length 2^{1-n}.

Now, we exploit the *tensor product* approach for the d-dimensional case and give a recursive formulation of the d-dimensional hierarchical basis $B^{(d)}_{\mathbf{n}^d}$ for $d > 1$

$$B^{(d)}_{\mathbf{n}^d} := B^{(d-1)}_{\mathbf{n}^{d-1}} \otimes B_{n_d},$$

where we define $\mathbf{n}^d := (n_1, \ldots, n_d) \in \mathbb{N}_0^d$. We start the recursion with $B^{(1)}_{n_1} := B_{n_1}$. The indices n_i $i = 1, \ldots, d$ indicate the depths of the basis $B^{(d)}_{\mathbf{n}^d}$ in the directions i. In the following considerations, we suppress the upper dimension

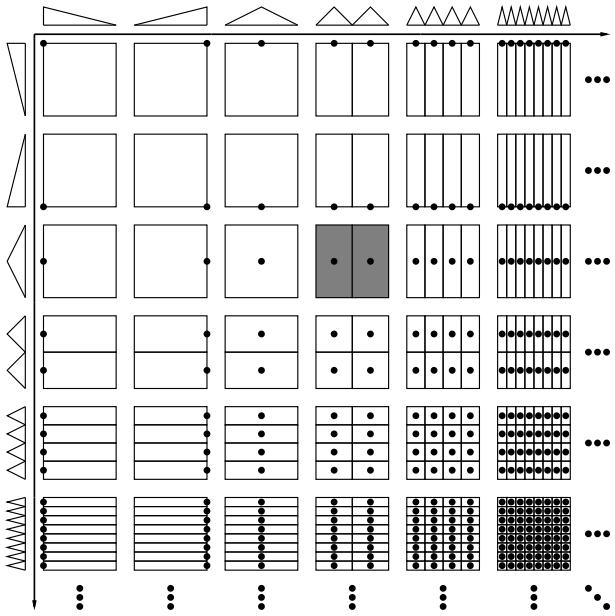

Figure 2: The multidimensional piecewise linear hierarchical basis: two-dimensional subspace scheme with supports supp $\phi_{\mathbf{x_j}}$ and basis points $\mathbf{x_j}$ of the corresponding hierarchical basis function $\phi_{\mathbf{x_j}}$. (For example, the hierarchical basis functions with the grey supports are displayed in Fig. 3.)

index (d) whenever the dimension d is clear from the context. The piecewise multilinear basis functions are defined as

$$\phi^{(d)}_{\mathbf{x_j}}(\mathbf{x}^{(d)}) := \phi^{(d-1)}_{\mathbf{x_j}}(\mathbf{x}^{(d-1)}) \cdot \phi^{(1)}_{x_{j_d}}(x^{(1)}_{j_d}) := \prod_{i=1}^{d-1} \phi_{x_{j_i}}(x_i) \cdot \phi_{x_{j_d}}(x_d) = \prod_{i=1}^{d} \phi_{x_{j_i}}(x_i),$$

where $\boldsymbol{x}^{(d)} := (x_1, \ldots, x_d) \in \Omega^{(d)} := [0,1]^d$. The coordinates of the basis point $\mathbf{x_j} := (x_{j_1}, \ldots, x_{j_d})$ of the d-dimensional basis function $\phi_{\mathbf{x_j}}$ are given by the d basis points of B_{n_i} of the corresponding one-dimensional basis functions in all directions $i = 1, \ldots, d$, see also the *subspace scheme* in Fig. 2. To get an impression of a typical two-dimensional basis function, see Fig. 3. The space spanned by $B^{(d)}_{\mathbf{n}^d}$ is called $V^{(d)}_{\mathbf{n}^d} := < B^{(d)}_{\mathbf{n}^d} > \subset V = H^1(\Omega^{(d)})$. Note that $V^{(d)}_{\mathbf{n}^d}$ is also generated by a classical tensor nodal basis with 2^{n_i+1} basis functions in direction $i = 1, \ldots, d$.

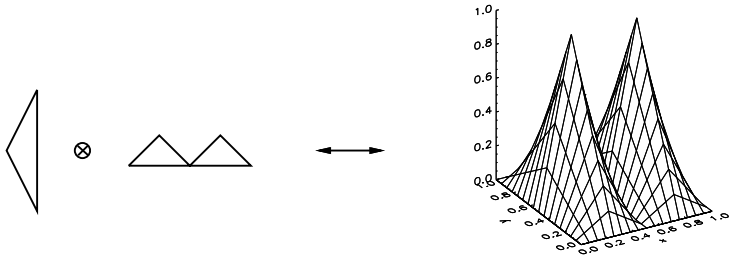

Figure 3: The multidimensional piecewise linear hierarchical basis: illustration of the tensor product approach for piecewise bilinear basis functions. The corresponding supports are shown in Fig. 2 (grey).

Any function u of the space $V_{\mathbf{n}^d}^{(d)}$ has the *hierarchical basis representation*

$$u_{\mathbf{n}^d}(\mathbf{x}) = \sum_{\mathbf{x_j} \in bp(B_{\mathbf{n}^d}^{(d)})} u_{\mathbf{n}^d,\mathbf{x_j}}^{\text{hier}} \, \phi_{\mathbf{x_j}}(\mathbf{x}),$$

with $u_{\mathbf{n}^d,\mathbf{x_j}}^{\text{hier}} \in \mathbb{R} \; \forall \phi_{\mathbf{x_j}} \in B_{\mathbf{n}^d}^{(d)}$. The coefficients $u_{\mathbf{n}^d,\mathbf{x_j}}^{\text{hier}}$ correspond to the increments of data coming from the basis function $\phi_{\mathbf{x_j}}$, and therefore, they are also called *hierarchical surplusses*.

As a model problem, we consider Poisson's equation with Dirichlet and Neumann boundary conditions

$$\begin{aligned} -\Delta u &= f \text{ in } \Omega^{(d)}, \\ u &= g \text{ on } \emptyset \neq \Gamma_D \subset \Gamma := \partial\Omega^{(d)}, \\ \partial_\nu u &= m \text{ on } \Gamma_N := \Gamma \setminus \Gamma_D. \end{aligned} \tag{2}$$

Let us denote the standard L_2-inner product by $(.,.)_X$ and by $\|.\|_X$ the corresponding norm on $\Omega^{(d)}$, resp. Γ. The *weak* or *variational formulation* of (2) reads then

$$(\nabla\phi, \nabla u)_\Omega = (\phi, f)_\Omega + (\phi, m)_\Gamma \quad \forall \, \phi \in V. \tag{3}$$

We also call (3) the *continuous primal problem*. Using the finite element method, we obtain an approximation $u_n \in < A_n >$ of the analytical solution $u \in V$ in the *ansatz space* $< A_n > \subset V$ by solving the *discrete primal problem* of (3) given by

$$(\nabla\phi, \nabla u_n)_\Omega = (\phi, f_n)_\Omega + (\phi, m_n)_\Gamma \quad \forall \, \phi \in T_n, \tag{4}$$

where $T_n \subset V$ is called the *test space*. Sticking to a Ritz-Galerkin approach, we choose $T_n = A_n \subset B^{(d)}_{\mathbf{n}^d}$. The discretization underlies the *grid* $G_n := bp(A_n)$.

The function $u_n = \sum_{\mathbf{x_j} \in bp(A_n)} u_{n,\mathbf{x_j}} \cdot \phi_{\mathbf{x_j}}$ interpolates the Dirichlet boundary value function g on $\Gamma \cap A_n$, $f_n = \sum_{\mathbf{x_j} \in bp(A_n)} f_{n,\mathbf{x_j}} \cdot \phi_{\mathbf{x_j}}$ interpolates the source function f in $\Omega^{(d)}$ and $m_n = \sum_{\mathbf{x_j} \in bp(A_n)} m_{n,\mathbf{x_j}} \cdot \phi_{\mathbf{x_j}}$ interpolates the Neumann boundary value function m on $\Gamma_N \cap A_n$. We end up with a system of linear equations $S \cdot u_n = b_n$ for the coordinate vector $u_n := (u_{n,\mathbf{x_j}})_{\mathbf{x_j} \in bp(A_n)} \in \mathbb{R}^N$ of the function u_n. We correspondingly define the coordinate vectors $f_n := (f_{n,\mathbf{x_j}})_{\mathbf{x_j} \in bp(A_n)} \in \mathbb{R}^N$ of the source function f and $m_n := (m_{n,\mathbf{x_j}})_{\mathbf{x_j} \in bp(A_n)} \in \mathbb{R}^N$ of the Neumann boundary function. The vector b_n is called *load vector*. The matrix S is known as the *stiffness matrix* with entries $s_{\mathbf{x_k},\mathbf{x_j}}$ defined by

$$S := \left(s_{\mathbf{x_k},\mathbf{x_j}}\right)_{\mathbf{x_k} \in bp(T_n), \mathbf{x_j} \in bp(A_n)} \in \mathbb{R}^{N \times N} \text{ and } s_{\mathbf{x_k},\mathbf{x_j}} := \left(\nabla \phi_{\mathbf{x_k}}, \nabla \phi_{\mathbf{x_j}}\right)_{\Omega^{(d)}}. \quad (5)$$

3 Adaptation

We only give a very crude description of the adaptation process in this article. In Subsection 3.1, we generally develop an a posteriori error estimation that allows us to distinguish two relevant error contributions, namely the primal error contribution arising from the discretization of the primal problem and the dual error contribution arising from the discretization of the dual problem. We use the primal error contribution for the a posteriori error estimation and the dual error contribution for the grid refinement. In Subsection 3.2, we present the grid refinement technique that makes explicit use of the hierarchical tensor product structure of the approach. For a detailed discussion of both the a posteriori error estimation and the self-adaptive algorithm see Schneider [12].

3.1 The A Posteriori Error Estimation

We like to derive a posteriori error estimates for the model problem (2) with an arbitrary, linear functional, e.g. of the form

$$J(\phi) := (w, \phi)_\Omega \text{ or } J(\phi) := (\mathbf{w}, \nabla \phi)_\Omega \quad (6)$$

where $\phi \in V$ (or a suitable restriction of V). The functions $w(.)$ and $\mathbf{w}(.)$ are suitably chosen weights.

Following the dual approach, we consider another boundary value problem, the so-called *continuous dual problem*, for the corresponding "error propagation" function $z \in V$

$$(\nabla z, \nabla \psi)_\Omega = J(\psi) + (y, \psi)_\Gamma \quad \forall\, \psi \in V, \tag{7}$$

with the functional J as the right-hand side, homogenous Dirichlet boundary conditions[1] $z|_{\Gamma_D} = 0$, and homogenous Neumann boundary conditions $y := \partial_\nu z$ with $y|_{\Gamma_N} = 0$. Since, we will make explicit use of the *continuous dual solution* z, we approximate z by the solution $z_m \in <T_m>$ of the *discrete dual problem*

$$(\nabla z_m, \nabla \psi)_\Omega = J(\psi) + (y_m, \psi)_\Gamma \quad \forall\, \psi \in A_m, \tag{8}$$

where the function $y_m = \sum_{\mathbf{x_k} \in bp(T_m \cap \Gamma)} y_{m,\mathbf{x_k}} \cdot \phi_{\mathbf{x_k}}$ interpolates the function $y = 0$ on $T_m \cap \Gamma_D$. We call z_m the *discrete dual solution*. Analogically, we denote the test space T_m as *dual test space* and the ansatz space A_m as *dual ansatz space*. In the following considerations, we assume that the primal spaces are included in the dual spaces: $T_n \subset T_m$ and $A_n \subset A_m$.

The aim is to create a primal ansatz space $A_n \subset V$ in order to minimize the value $|J(e)|$ for the error $e := u - u_n \in V$ of the discrete primal problem (4). The error $J(e)$ can be decomposed into two components: the *primal error contribution* $\eta_m \in \mathbb{R}$ and the *dual error contribution* $\theta_m \in \mathbb{R}$

$$\eta_m := (\nabla z_m, \nabla(u - u_n))_\Omega - (y_m, g - g_n)_\Gamma \text{ and} \tag{9}$$

$$\theta_m := (\nabla(z - z_m), \nabla u)_\Omega - (y - y_m, g)_\Gamma\,. \tag{10}$$

The primal error contribution η_m arises from the discretization of the primal problem involving the discrete dual solution z_m, and the dual error contribution θ_m arises from the dual problem involving the continuous primal solution u. Again, these dependencies reflect the duality of the problems. The total of the discretization error then is

$$J(e) = \eta_m + \theta_m. \tag{11}$$

[1]Note that the continuous dual function vanishes where the continuous primal function has prescribed Dirichlet boundary conditions, and the normal derivative of the continuous dual function vanishes where the continuous primal function has prescribed Neumann boundary conditions.

Equation (11) is a quantitative relation between the discretization error $J(e)$ and the primal and dual error contributions η_m and θ_m.

We like to estimate the error $J(e)$ by an appropriate error estimation in an economic way. Since the discretization of the dual problem in (8) is consistent with the continuous dual problem in (7), the dual error contribution θ_m tends to zero with higher resolution of the discrete dual solution z_m. This means that only the *primal error contribution* $\eta \in \mathbb{R}$ remains in (11)

$$\eta := (\nabla z, \nabla(u - u_n))_\Omega - (y, g - g_n)_\Gamma = \lim_{m \to \infty} \eta_m = J(e). \tag{12}$$

We learn two important things from equation (12). First, the continuous dual solution z rules the discretization error $J(e)$. Therefore, the dual solution z has the key position in the presented approach. Second, for a "sufficiently good" approximation z_m of the continuous dual solution z, we may use η_m as an error estimation for the total of the discretization error $J(e)$. Here, a "sufficiently good" approximation of z has two aspects: on one hand, because of an accurate error estimation, the computation of z_m is carried out on a not too coarse grid and, on the other hand, because of the computational costs, the computation of z_m is carried out on a not too fine grid. Because we have already postulated $T_n \subset T_m$ and $A_n \subset A_m$, we define in the numerical examples $T_m := T_n$ and $A_m := A_n$.

3.2 The Grid Refinement

The grid refinement process depends on data from the continuous primal solution u, e.g. the locations of the singularities. As presented in Schneider [12], the primal error contribution η_m can be calculated from the discrete primal and discrete dual solution u_n and z_m, the given source data f, the given Neumann data m, and the given Dirichlet data g. Therefore, we focus on the dual error contribution θ_m for the necessary data for the grid refinement. Due to the hierarchical approach, we only consider elements of the current approximation that can still be refined, so-called *leave elements*. Because the supports of these leave elements are by construction disjunct, we can estimate their contribution to the dual error contribution θ_m by a value that is proportional to the product of the corresponding surplusses. This

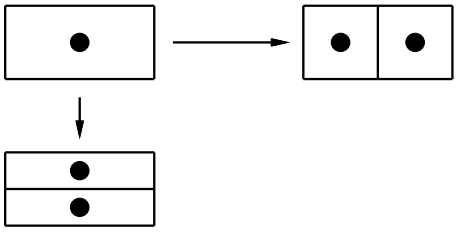

Figure 4: The refinement of an element: geometrical illustration for the subdivision of an element to be refined in the two-dimensional case.

motivates to introduce the *local dual error contributions*

$$\theta_{m,\mathbf{x_k}} := z^{\text{hier}}_{m,\mathbf{x_k}}\, u^{\text{hier}}_{m,\mathbf{x_k}} \left[(\nabla\phi_{\mathbf{x_k}}, \nabla\psi_{\mathbf{x_k}})_\Omega - \left(\partial_\nu \phi_{\mathbf{x_k}}|_\Gamma \,,\, \psi_{\mathbf{x_k}}|_\Gamma \right)_\Gamma \right] \quad \forall\, \mathbf{x_k} \in bp(T_m). \tag{13}$$

Here, we make the basic assumption that the hierarchical surplus asymptotically decays in regions where the concerning functions are sufficiently smooth. However, the local dual error contributions $\theta_{m,\mathbf{x_k}}$ are also appropriate in regions where the corresponding functions are not smooth enough, as the numerical examples in the Sections 6 and 7 show.

We create new ansatz basis functions in the following way: we suppose that an indicated element corresponds with the ansatz basis function $\psi_{\mathbf{x_j}} \in B^{(d)}_{\mathbf{n}^d}$. Then, we add to the refined discretization A_{n+1} all ansatz basis functions of the next hierarchical level $\psi_{\tilde{\mathbf{x}}_\mathbf{j}} \in B^{(d)}_{\tilde{\mathbf{n}}^d} \setminus B^{(d)}_{\mathbf{n}^d}$ with $\text{supp}(\psi_{\tilde{\mathbf{x}}_\mathbf{j}}) \subset \text{supp}(\psi_{\mathbf{x_j}})$, in which we use the usual multi-index notation $\tilde{\mathbf{n}}^d = \mathbf{n}^d + \mathbf{1}$. Geometrically speaking, subdividing an element means to create in each coordinate direction $i = 1, \ldots, d$ two new finite elements with support of half size in the direction i and the same support size in all other directions k with $1 \leq k \neq i \leq d$. Fig. 4 illustrates this creation of new elements in the two-dimensional case.

In order to balance all local dual error contributions $|\theta_{m,\mathbf{x_k}}|$, we indicate all elements with basis point $\mathbf{x_k} \in G_n$ satisfying

$$|\,\theta_{m,\mathbf{x_k}}| \geq \gamma \max_{\mathbf{x_k} \in G_n} |\theta_{m,\mathbf{x_k}}|. \tag{14}$$

This means that we refine all elements of the set of all leave elements with expected comparatively large contribution to the discretization error. The real number $0 \leq \gamma \leq 1$ prescribes a threshold depending on the error functional J and the discretization order p. The value of γ controls how many

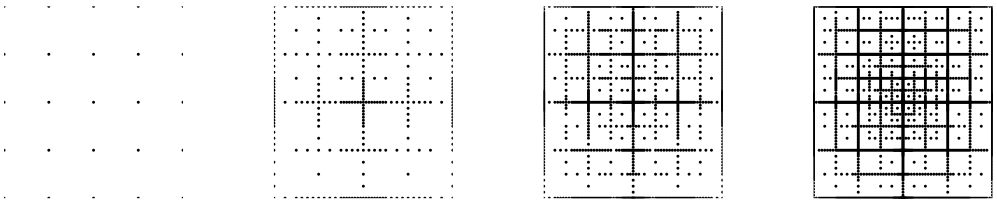

Figure 5: A two-dimensional example for a mid-point value adaptation: initial and adaptive grids after 8, 12, and 18 refinement steps.

new grid points are generated per refinement step and, therefore, we define the *grid point quotient of a refinement step*

$$q_n := \frac{N(A_{n+1})}{N(A_n)} \in \mathbb{R} \quad \forall\, n \geq 0. \tag{15}$$

Hereby, we have to take care: on one hand, if we create too few new grid points ($q_n \to 1$), the solution of the primal and dual discrete problems is relatively too expensive in comparison with the calculations of the adaptation process. On the other hand, if we create too many new grid points, the self-adaptive grid refinement cannot find the crucial regions corresponding to the error functional J because of over-adaptation. When we refine all grid points in every refinement step, we end up with the so-called regular sparse grids ($q_n \to 2$), see Section 4. For the complete self-adaptive algorithm see again Schneider [12].

We give an example for the two-dimensional Dirichlet model problem (2) with solution $u(x_1, x_2) := \sin \pi x_1 \cdot \sinh \pi x_2 / \sinh \pi$. We choose the point-value $J(u) := u\,(1/2, 1/2)$ in the mid-point as error functional by defining in (6) the Dirac's delta distribution as weight $w(\mathbf{x}) := \delta\,(1/2 - x_1, 1/2 - x_2)$. The adaptation process creates a sequence of grids extremely refined towards the mid-point, see Fig. 5.

4 Regular Sparse Grids

In this section, we newly define regular sparse grids by mathematical induction. The regular sparse grids were firstly introduced by Zenger [16]. We

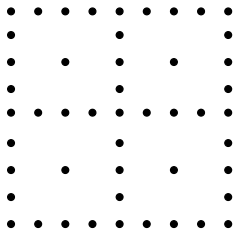

Figure 6: Regular sparse grids: the two-dimensional regular sparse grid G_3 with depth $t = 3$ (49 grid points).

start with a regular full grid G_0 of depth $t = 0$ with 2^d ansatz basis functions and set the local dual error contributions $\theta_{m,\mathbf{x_k}} := 1 \quad \forall\, \mathbf{x_k} \in T_m$ in every refinement step. This leads to a maximum grid refinement ($q_n \to 2$) in which all grid points are indicated, see equation (14). Consequently, they are all refined. In Fig. 6, a two-dimensional regular sparse grid G_3 is presented. We recognize the recursive grid pattern.

5 Multigrid Method

In this chapter, we design an efficient multigrid solver for problems discretized on grids that have no longer a regular structure, e.g. on grids that are extremely refined at certain points.

The basic idea is to relax on all full (semi-)coarsened grids involved in the discretization, see [9]. For two-dimensional regular sparse grids, this was already implemented by Pflaum, see [10]. Here, we provide all subgrids of the ansatz basis functions A_n that have no more refined elements in A_n in any direction. We do a classical correction scheme V-cycle with one pre- and post-smoothing step, in which we semi-coarsen only in one direction, say x_1, see also Fig. 7 for a two-dimensional example. The grey subgrids are the starting grids for the V-cycle marked by the arrows.

We smooth with a Jacobi relaxation with damping parameter $\omega = 2/3$ in the nodal basis of the subgrid. The correction δu_n of the coefficients u_n on the current subspace $\tilde{A}_n$ influences the current residual $r_n := b_n - S u_n$ by $\delta r_n = -\tilde{S} u_n$, where $\tilde{S}$ is a suitable restriction of the stiffness matrix (5) to

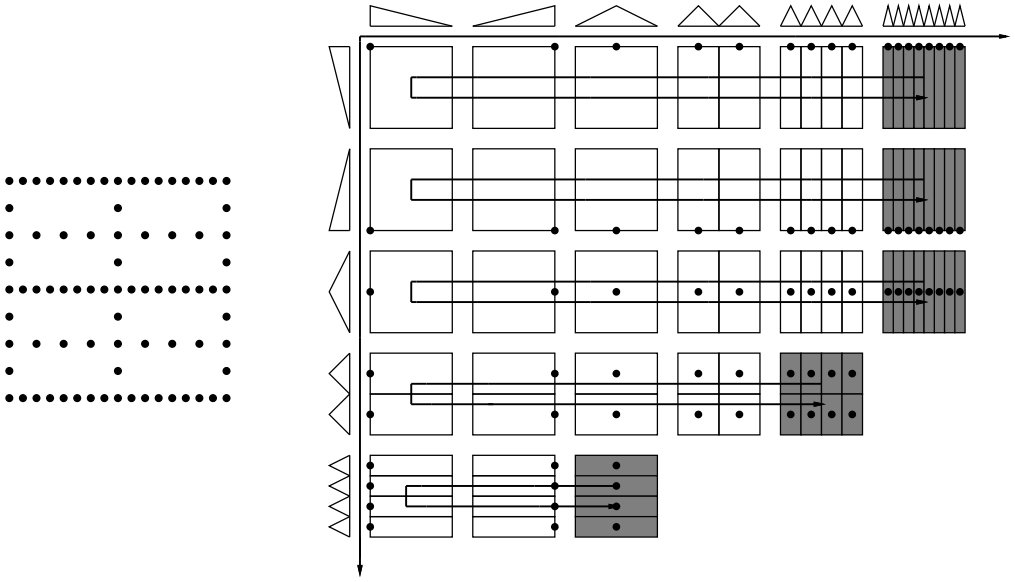

Figure 7: The error smoothing: two-dimensional example for an adaptive grid (left) and subgrids introduced by the hierarchical basis (right). The arrows indicate the V-cycle with one pre- and post-smoothing step.

the current subspaces $\tilde{T}_n$ and $\tilde{A}_n$. Note that in general not all points of the subgrids will be used in the discretization. For reasons of computational time, we relax only on those points that are taken into account in the discretization.

The residual r_n is computed in the hierarchical basis of the current test subspace $\tilde{T}_n$. Because we correct u_n in the nodal basis of the current subspace $\tilde{A}_n$, we have to dehierarchise the current residual r_n. The d-dimensional transformation for dehierarchising and also for hierarchising a function u_n is due to the tensor product approach reduced to a sequence of one-dimensional dehierarchising or hierarchising transformations, see Fig. 8. For the one-dimensional hierarchical basis transformations, see Balder [1].

The fine-to-coarse transfer is done by simple restriction, whereas the coarse-to-fine transfer is done by multilinear interpolation.

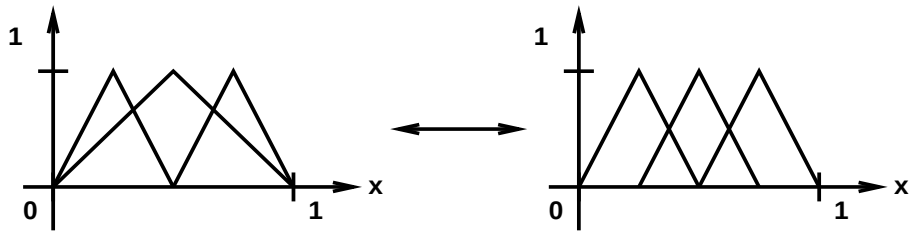

Figure 8: Involved basis functions ϕ of the one-dimensional basis transformation from hierarchical (left) to nodal (right) representation and vice versa.

6 Numerical Examples

The algorithms for the adaptation process and multigrid solver are implemented as a C program, based on linearized binary trees as underlying data structure; for algorithmic details see Schneider [12].

We study the influence of the regarded error functional on the refinement process in two ways: first, we are looking at the grid generation and the behaviour of the error-complexities ε, and second, we focus on the reduction of the l_2-norm of the residuum per iteration ρ with respect to the number of degrees of freedom N

$$\rho := \frac{\|r_{n,\text{new}}\|_2}{\|r_{n,\text{old}}\|_2}.$$

We always compare the *convergence rate* ρ for the adapted grid with the convergence rate ρ for full grids.

In the numerical examples we restrict ourselves to Dirichlet boundary problems ($\Gamma_N = \emptyset$) and focus on four different error measurements: the discrete maximum norm, indicated in all figures by ($\diamond$), taking the maximum absolute value over all grid points; the discrete l_2-norm ($*$) and the discrete energy-norm ($\triangle$) are calculated as means of all squared values of the discrete function u_n or Su_n, resp., on an auxiliary grid where all elements are subdivided as it is described in Section 3. Last, we study the error in the mid-point $P = (1/2, ..., 1/2)$ of the d-dimensional unit cube, indicated by ($\cdot$). In all diagrams, the respective error is plotted with respect to the number of degrees of freedom N. Also a solid reference curve is given. The position of this curve is chosen for reasons of clarity.

6.1 The Two-Dimensional Poisson Problem

Let us consider the two-dimensional model problem (2) with the solution $u(x_1, x_2) := 10 \cdot \sin(\, 4x_1 + 2x_2 - 1\,)$ studied by Rannacher [2], see the left-hand side of Fig. 9.

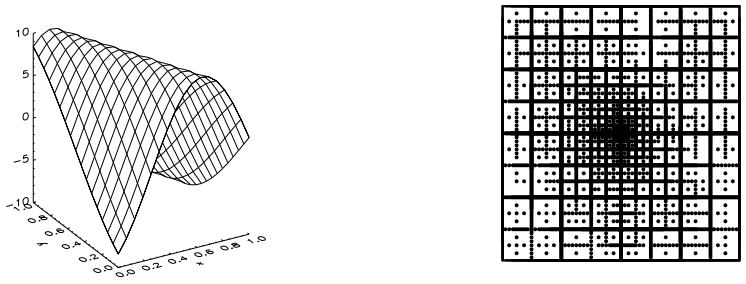

Figure 9: The two-dimensional Poisson problem: model problem (2) with solution $u(x_1, x_2) := 10 \cdot \sin(\, 4x_1 + 2x_2 - 1\,)$ (left) and refined grid for computing $u\,(1/2, 1/2)$ with best accuracy with 12 624 grid points (right).

6.1.1 The Full Grid Approximation

In Fig. 10, we present the error-plot for a full grid discretization. On the left-hand side we see the convergence behaviour for the l_2-, mid-point, and maximum error of order of magnitude $O(N^{-1})$, and for the energy-error of order of magnitude $O(N^{-1/2})$. Both results are well known. The convergence rate ρ is bounded by 0.1111 and therefore N-independent, as shown on the right-hand side.

6.1.2 The Adaptive Point-Value Approximation

Now, we like to compute the point-value for the mid-point P with best accuracy. Therefore, we again define in (6) Dirac's delta distribution as weight $w(\mathbf{x}) := \delta\,(1/2 - x_1, 1/2 - x_2)$ and get as error functional $J(e) := e\,(1/2, 1/2)$. Fig. 11 provides the results. The convergence rate ρ is bounded by 0.3112. The convergence behaviour of the interesting quantity mid-point error is exactly the same as for the interpolant, see Bungartz [4], and has

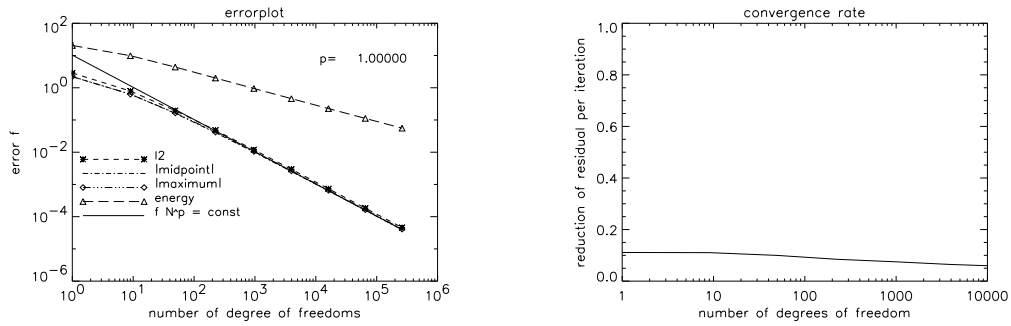

Figure 10: The two-dimensional Poisson problem: error-plot with $O(N^{-1})$ reference curve (left) and the convergence rate ρ (right) for the full grid discretization.

the order of magnitude $O\left(N^{-2}(\log N)^3\right)$ indicated by the sparse grid convergence (sgc) reference curve. The log-terms appear due to the fact, that the right-hand side is evaluated exactly on the same grid structure as used for the calculation of the numerical solution u_n.

If we compute the right-hand side on a grid with higher resolution, say on an suitably chosen regular sparse grid, we are able to suppress the log-terms, see Fig. 12. Note that the order of the computation time is then no longer given by the solution of the elliptic problem, but dominated by the calculation of the right-hand side. The convergence behaviour of the mid-point error is then $O(N^{-2})$ like the reference curve $O(N^{-2})$ is indicating, and the convergence rate ρ is bounded by 0.3106. This means that, if we compute the right-hand side carefully, we are able to get an $O(N^{-2})$ convergence behaviour for a point-value error using linear finite elements for both the primal and the dual problem. Note that in comparison to the full grid discretization, we gain one order of magnitude in accuracy.

A typical grid of the mid-point adaptation process is shown on the right-hand side in Fig. 9. As expected, the grid is extremely refined towards the mid-point and reflects the shape of the primal solution u, because the order of magnitude of the function u also influences the order of magnitude of the hierarchical surplus, and with that the order of magnitude of the corresponding error contribution. Here, the average grid point quotient of a refinement step q defined in (15) is 1.306.

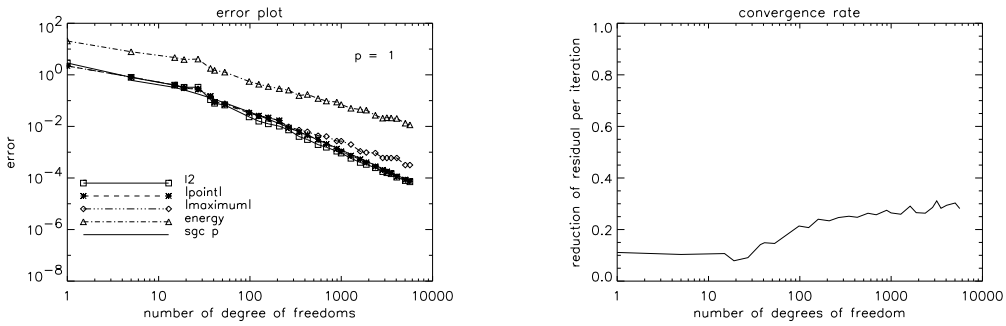

Figure 11: The two-dimensional Poisson problem: point-value adaptation with evaluation of the right-hand side on the same grid structure used for the calculation of the numerical solution u: error-plot with $O\left(N^{-2}(\log N)^3\right)$ reference curve (left) and the convergence rate ρ (right).

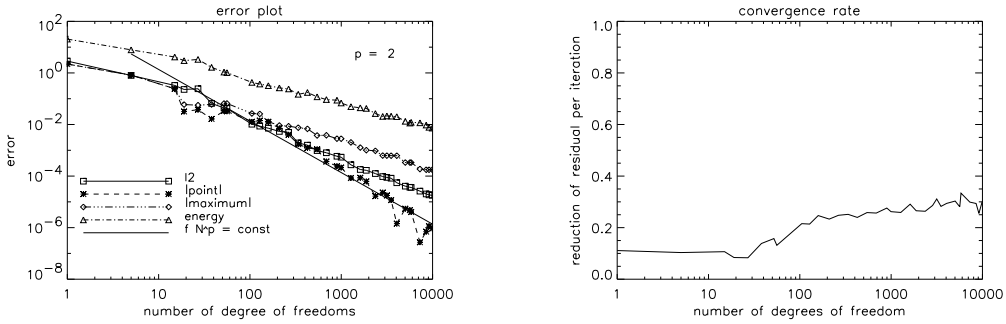

Figure 12: The two-dimensional Poisson problem: point-value adaptation with evaluation of the right-hand side on an suitably chosen sparse grid: error-plot with $O(N^{-2})$ reference curve (left) and the convergence rate ρ (right).

6.2 The Three-Dimensional Laplace Problem

Due to the tensor product finite element approach all theoretical results hold for arbitrary dimension d. E.g., let us focus on the three-dimensional model problem (2) with the solution

$$u(x_1, x_2, x_3) := \frac{\sinh\left(\sqrt{2.0}\,\pi x_1\right)}{\sinh\left(\sqrt{2.0}\,\pi\right)} \sin(\pi x_2) \sin(\pi x_3),$$

i.e. Laplace's equation to neglect the influence of the right-hand side. By defining $\mathbf{w}(\mathbf{x}) := \|\nabla e\|^{-1} \nabla\phi(\mathbf{x})$ in (6) we choose the energy-norm as error functional $J(e) = \|\nabla e\|$. In Fig. 13 we present the convergence rate ρ for the full grid discretization, bounded by 0.3763 (above left), and the energy adaptation process (above right), bounded by 0.4632, which is only little worse than for the full grid approximation. Below, we see the $O(N^{-1})$ convergence of the energy error, indicated by the solid reference curve.

7 Extensions

A finite element discretization with basis polynomials of order $p \geq 1$ is presented in Bungartz [4]. The main idea is to use no longer the classical linear hat function ϕ defined in (1), but appropriate polynomials of higher and arbitrary degree p. To be more precise, in the construction of the set of basis functions, we still keep the linear functions for the boundary influence from B_0. However, for all basis functions $\phi \in B_n \setminus B_{n-1}\, \forall n > 1$, we choose polynomials of degree $\max\{p, n+1\}$ with zeros in all corresponding hierarchical ancestor points of $bp(B_{n-1})$ that are normalized in the basis point by $\phi_{x_j}(x_j) = 1$. An example of a basis polynomial of degree $p = 4$ is shown on the left-hand side of Fig. 14.

As solution of the model problem (2), we choose an example with a root singularity on the boundary $u(x_1, x_2) := Re\left(\sqrt{z - 1/2}\right)$, see also the right-hand side of Fig. 14. We are interested to calculate again the mid-point with best accuracy.

The weak formulation (4) now leads to a Petrov-Galerkin approach, using for the test space still the linear space T_n introduced in Section 2 and for the ansatz space an appropriate space A_n^p based on the finite elements of order p.

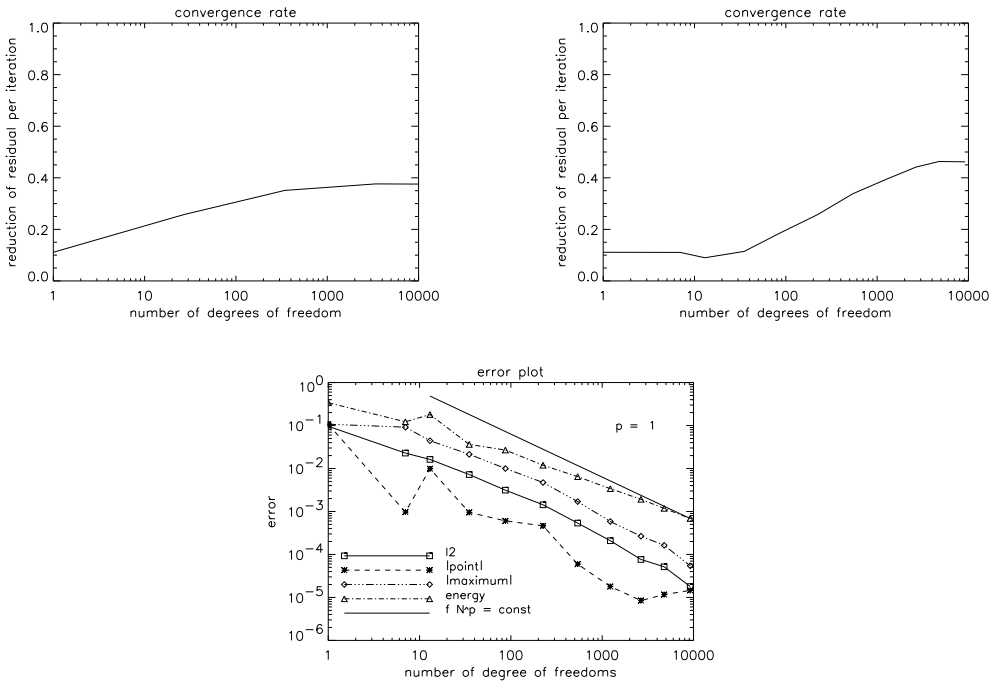

Figure 13: The three-dimensional Laplace problem: the convergence rate ρ for the full grid approximation (above left) and for the energy-error adaptation (above right); the error-plot for the energy-error adaptation with $O(N^{-1})$ reference curve (below).

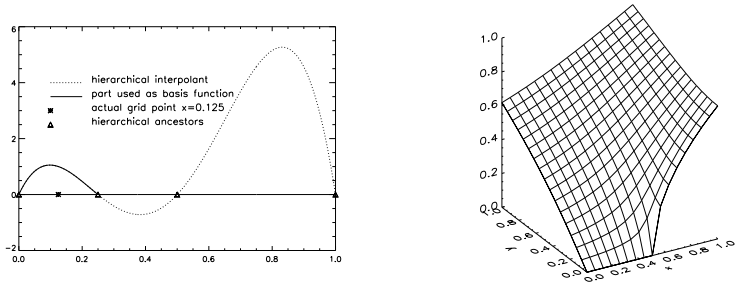

Figure 14: An example for the construction of a one-dimensional basis polynomial with degree $p = 4$ (left) and the solution u of the two-dimensional model problem (2) with root singularity on the boundary in the point with the coordinates $(1/2, 0)$ (right).

Fig. 15 illustrates the error-plots and the convergence rate ρ for the different polynomial degrees $p = 1, 2, 3$. One clearly sees the mid-point error behaviour of order of magnitude $O\left(N^{-2p}\right)$ for $p = 1, 2, 3$, indicated by the corresponding solid reference curves. Note that we only need about 2000 grid points to reach machine accuracy (single precision) for the discretization order $p = 3$. The convergence rate ρ stagnates p-independently and is bounded by 0.4506 for $p = 1$, by 0.3890 for $p = 2$ and by 0.4795 for $p = 3$. The average grid point quotient of a refinement step q defined in (15) is about 1.39.

8 Concluding Remarks

In the previous sections, we introduced concepts for an a posteriori grid-refinement technique and for a multiplicative multigrid solver independently from the dimension d, for details see again [12].

Both concepts are based upon hierarchical bases and have been successfully combined even for singularities. This allows us to solve elliptic partial differential equations in optimal time, with respect to both the order of magnitude of the number of degrees of freedom N and the computational time per unknown. We have seen in Sections 6 and 7, that the convergence rate ρ is N-independent and approximately bounded by $1/2$ for one pre- and post-

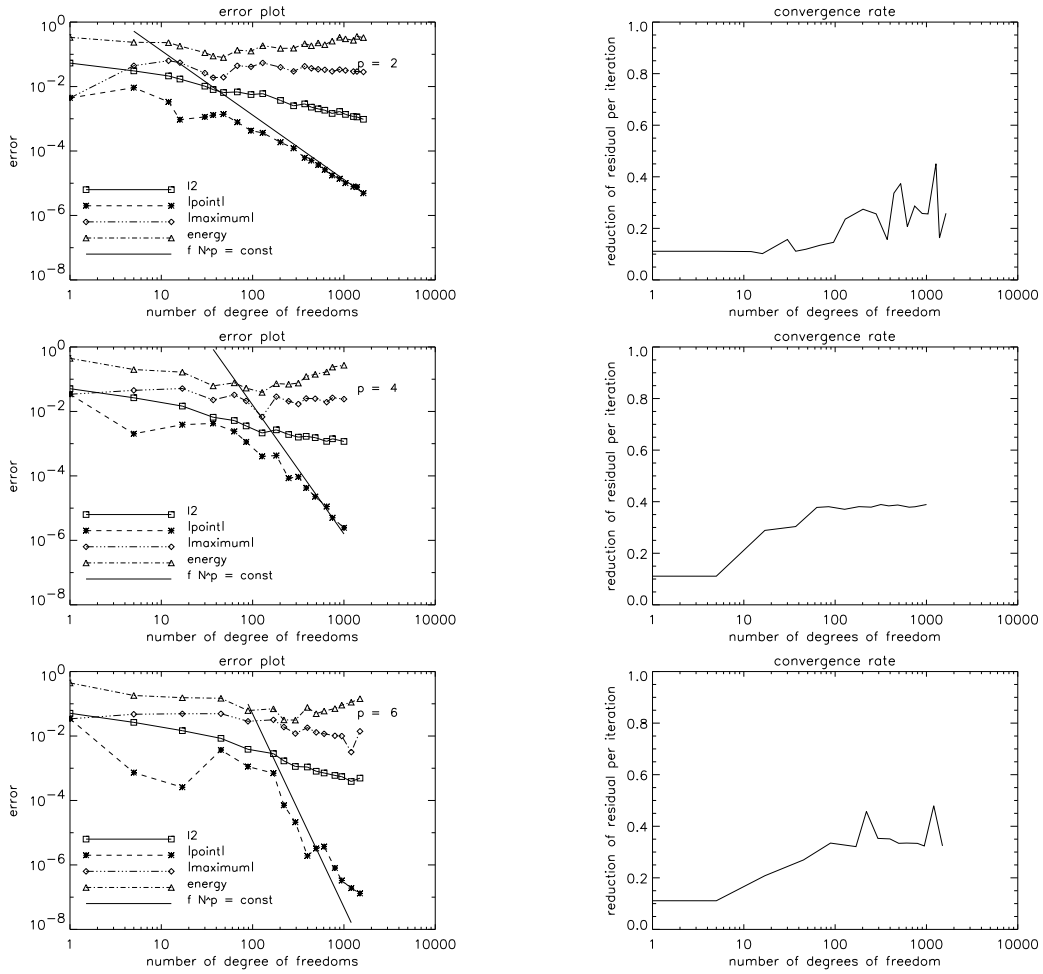

Figure 15: The two-dimensional Laplace problem with root singularity on the boundary: error-plots with reference curves $O(N^{-2p})$ (left) and the corresponding convergence rate ρ (right) for higher order point-value adaptation for the discretization orders $p = 1$ (above), $p = 2$ (center) and $p = 3$ (below).

smoothing step. In other words, we gain a reduction of the l_2-norm of the residuum of approximately 0.7 per relaxation step, independently from the number of degree of freedoms N, the dimension d, the order of approximation p, and the particular grid structure. This is a satisfactory result, even though the numerical value of ρ still has to be improved. Another very promising result is that we achieve for the two-dimensional mid-point error problem the convergence behaviour of order of magnitude $O(N^{-2p})$, using only finite elements of order p. We also gave an example where the order of the computational time is dominated by the calculation of the right-hand side and not, as one might expect, by the solution of the elliptic problem. In the opinion of the authors, there is a huge potential in the method of adaptive hierarchical tensor product finite elements.

References

[1] R. Balder and C. Zenger. The Solution of Multidimensional Real Helmholtz Equations on Sparse Grids. *SIAM J. Sci. Comp.*, 17(3):631–646, 1996.

[2] R. Becker and R. Rannacher. A feedback approach to error control in finite elements methods: basic analysis and examples. *East-West J. Numer. Math.*, 4:237–264, 1996.

[3] H.-J. Bungartz. A Multigrid Algorithm for Higher Order Finite Elements on Sparse Grids. *ETNA*, 6:63–77, 1997.

[4] H.-J. Bungartz. *Finite Elements of Higher Order on Sparse Grids.* Habilitationsschrift, Institut für Informatik, TU München, 1998.

[5] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to Adaptive Methods for Differential Equations. *Acta Numerica*, pages 105–158, 1995.

[6] K. Eriksson and C. Johnson. An adaptive finite element method for linear elliptic problems. *Math. Comp.*, 50:361–3838, 1988.

[7] G. Faber. Über stetige Funktionen. *Mathematische Annalen*, 66:81–94, 1909.

[8] M. Griebel and P. Oswald. On additive Schwar preconditioners for the sparse grid discretizations. *Numer. Math.*, 66:449–463, 1994.

[9] N. H. Naik and J. R. van Rosendale. The improved robustness of multigrid elliptic solvers based on multiple semicoarsened grids. *SIAM J. Numer. Anal.*, 30:215–229, 1993.

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

[11] R. Rannacher. Error control in finite element computations. In H. Bulgak and Chr. Zenger, editors, *Proc. 3^{rd} Europ. PVM Conf., Munich, 1996*, volume 536 of *NATO Advanced Study Institute, Series C: Mathematical and Phisical Sciences*, pages 247–278. Kluwer academic publishers, the Netherlands, 1999.

[12] S.-A. H. Schneider. *Adaptive Solution of Elliptic Partial Differential Equations by Hierarchical Tensor Product Finite Elements.* In preparation, 1999.

[13] K. Simony. *Kulturgeschichte der Physik.* Akadémiai Kiadó, Budapest, 1990.

[14] U. Trottenberg, K. Oosterlee, and A. Schüller. Durchbruch im Wissenschaftlichen Rechnen durch adaptive Mehrgitterverfahren auf Parallelrechner. *Der GMD-Spiegel*, 3:15–17, 1997.

[15] H. Yserentant. On the multi-level splitting of finite element spaces. *Numer. Math.*, 49:379–412, 1986.

[16] C. Zenger. Sparse Grids. In W. Hackbusch, editor, *Parallel Algorithms for Partial Differential Equations*, volume 31 of *NNFM*. Vieweg, Braunschweig/Wiesbaden, 1991.