TECHNISCHE
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR INFORMATIK

Sonderforschungsbereich 342:
Methoden und Werkzeuge für die Nutzung
paralleler Rechnerarchitekturen

Sparse Grids: Recent Developments For Elliptic Partial Differential Equations

Hans-Joachim Bungartz and Thomas Dornseifer

TUM-I9702
SFB-Bericht Nr. 342/02/97 A
Februar 97

TUM–INFO–02-I9702-200/1.–FI

Anforderungen an: Prof. Dr. A. Bode
Sprecher SFB 342
Institut für Informatik
Technische Universität München
D-80290 München, Germany

Druck: Fakultät für Informatik der
Technischen Universität München

Sparse Grids: Recent Developments For Elliptic Partial Differential Equations

Hans-Joachim Bungartz and Thomas Dornseifer*

February 11, 1997

Abstract

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

In this paper, we discuss two new algorithmic developments concerning the sparse grid finite element discretization of elliptic partial differential equations. First, a method for the numerical treatment of the general linear elliptic differential operator of second order is presented which, with the help of mapping techniques, allows to tackle problems on more complicated geometries. Second, we leave the approximation space of the piecewise multilinear functions and introduce hierarchical polynomial bases of piecewise arbitrary degree that lead to a very straightforward and efficient access to an approximation of higher order on sparse grids.

Both algorithms discussed here have been designed in a unidirectional way that allows the recursive reduction of the general d-dimensional case to the simpler 1 D one and, thus, the formulation of programs for arbitrary d.

1 Introduction

Though the idea of using hierarchical representations of functions for tasks like interpolation or numerical quadrature has a long tradition that at least goes back to Archimedes' attempts to integrate $1 - x^2$ over $[-1, 1]$, it was not before the eighties of our century that a hierarchical approach was studied in detail for a p. d. e. or, to be more precise, a finite element context [5, 29]. One of the main advantages of hierarchical bases compared with standard nodal point bases is probably the fact that the basis gets a multilevel structure. I. e., we can now distinguish between high-level basis functions with a large support that usually (in the

*Institut für Informatik, Technische Universität München, D - 80290 München, Germany; e-mail: {bungartz,dornseif}@informatik.tu-muenchen.de. This work has been supported by the Bayerische Forschungsstiftung via FORTWIHR – The Bavarian Consortium for HPSC.

smooth case, at least) already contain a significant part of the information, and functions living on lower levels whose contribution to an interpolant or a finite element approximation is rather small. The decrease of the hierarchical coefficients from level to level can be used, of course, to control adaptive grid refinement, but, if it is combined with a tensor product approach for the higher dimensional case, it can be used for an a priori reduction of the number of grid points involved in the calculation, too.

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

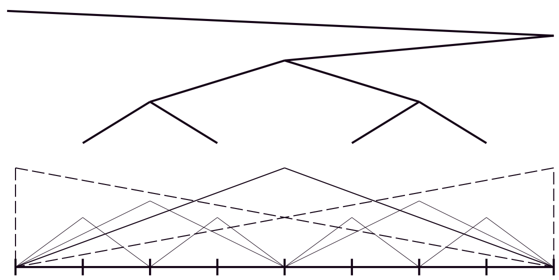


Figure 1: Piecewise linear hierarchical basis and corresponding binary tree

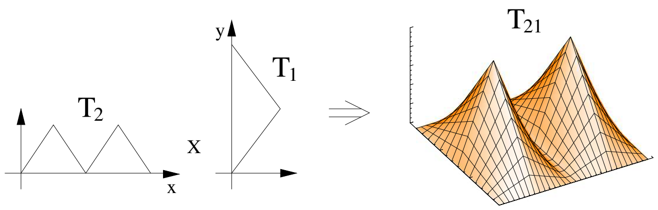
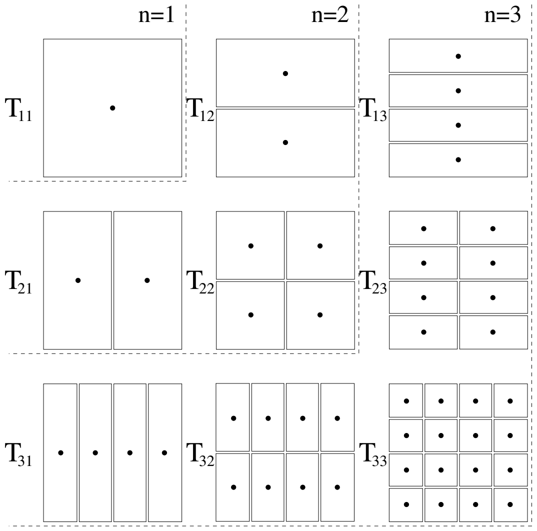


Figure 2: Tensor product approach for two piecewise bilinear hierarchical basis functions

In Fig. 3, for $d = 2$, we show a sector of the theoretically infinite scheme of subspaces. Here, a standard full grid $G_n^{(2)}$ with $(2^n - 1)^2$ inner grid points corresponds to a square sector of n^2 subspaces T_{i_1,i_2}, and T_{i_1,i_2} contains all basis functions with congruent supports of the same aspect ratio. Obviously, the dimension (i. e. the number of grid points) of all subspaces T_{i_1,i_2} with $i_1 + i_2 = c$ is just 2^{c-2}. Furthermore, for functions u continuous on the unit square $\bar{\mathbf{Q}}$, it has been shown that the contribution of all T_{i_1,i_2} with $i_1 + i_2 = c$ to the interpolant of u is of the same order $O(2^{-2c})$ with respect to the L_2- or L_∞-norm and $O(2^{-c})$ with regard to the H^1-norm, if $\frac{\partial^4 u}{\partial x_1^2 \partial x_2^2}$ and some lower mixed derivatives of u are continuous on $\bar{\mathbf{Q}}$ (see [7, 8, 26, 30]). For general d, analogous results have been shown for subspaces $T_{i_1,\ldots,i_d}$ with $i_1 + \ldots + i_d = c$, if $\frac{\partial^{2d} u}{\partial x_1^2 \ldots \partial x_d^2}$ and some lower mixed derivatives of u are continuous on $\bar{\mathbf{Q}} = [0,1]^d$ (see [8, 26]). Due to these properties concerning cost (number of grid points) and benefit (order of approximation), it turns out to be more reasonable to deal with triangular schemes of subspaces as given in Fig. 4 instead of using square ones. The grids or patterns of grid points resulting from such triangular sections are called *sparse grids*.



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

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

For a formal definition of sparse grids, see [8, 30]. Besides the regular sparse grids that result from skipping certain subspaces according to Fig. 4, there is a very straightforward access to adaptive grid refinement. The hierarchical coefficient or *hierarchical surplus* itself can be used to indicate the smoothness of u at the corresponding grid point and, consequently, the necessity to refine the grid here. Figure 5 shows a regular 2 D sparse grid and an adaptively refined 3 D one with singularities at the re-entrant corner and along the three edges starting from there.

Talking about the most important properties of sparse grids, we have at least to look at the number of grid points involved and at the approximation accuracy in the piecewise multilinear case. For a detailed analysis, we once again refer to [8, 30]. For general d, the approach described above and illustrated in Fig. 4 leads to regular sparse grids with $O(N(\log_2(N))^{d-1})$ grid points, if N denotes the maximum number of grid points in one dimension (i. e., $\frac{1}{N}$ is the smallest mesh width occurring). With some modification, sparse grids with $O(N)$ grid points can be defined, too. Concerning the approximation quality, the accuracy of the sparse grid interpolant is only slightly deteriorated from $O(N^{-2})$ to $O(N^{-2}(\log_2(N))^{d-1})$ with respect to the L_2- or L_∞-norm. With regard to the H^1-norm, both the sparse grid interpolant and the finite element approximation to the solution of the given problem are of the order $O(N^{-1})$. Thus, we get a number of grid points that is nearly or even actually independent of d (a behaviour known from Monte Carlo methods, e. g.), but we have to pay for it with only a logarithmic loss in accuracy. Therefore, sparse grids are a very promising approach for many tasks in numerical mathematics [4, 6, 9, 10, 11, 12, 15, 16, 17, 21, 22, 23, 27] and especially attractive for problems with a large parameter d.

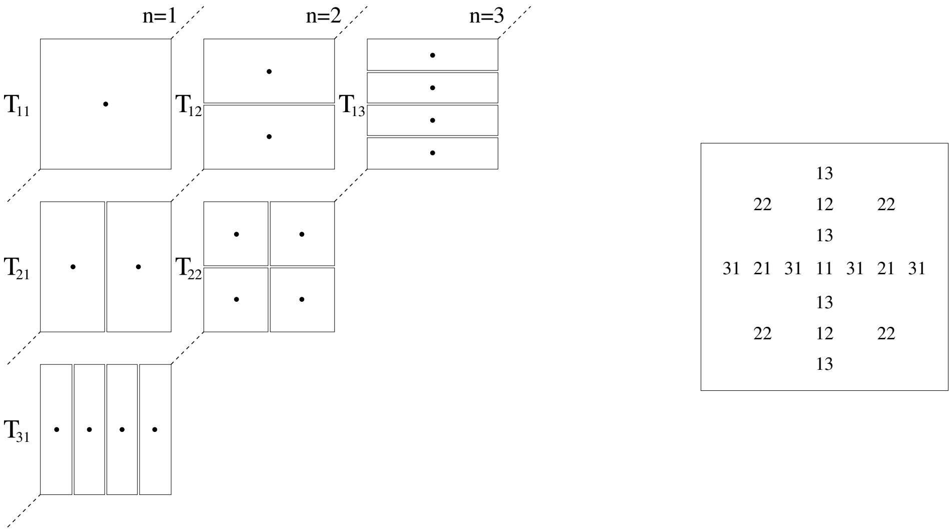


Figure 4: Hierarchical subspace decomposition: subspace scheme for sparse grids $\tilde{G}_n^{(2)}$ (left) and corresponding pattern of grid points for $n = 3$ (right)

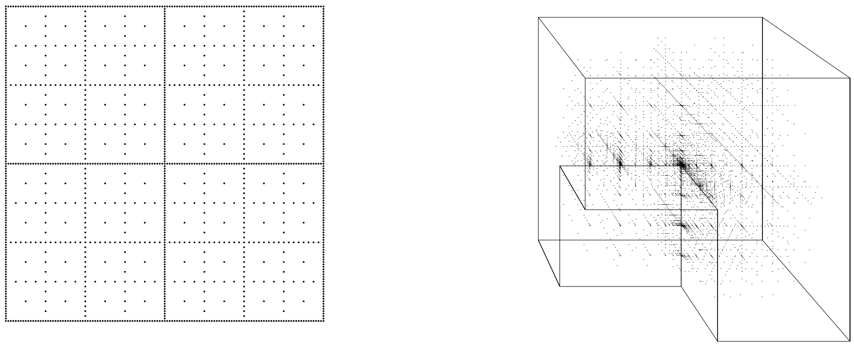


Figure 5: Sparse grids: regular example (left) and adaptive one (right)

2 Algorithmic Aspects

In this paper, we deal with finite element discretizations on sparse grids for the numerical solution of elliptic partial differential equations. The main algorithmic principle of either sparse grid method discussed here is the so-called *unidirectional* approach (cf. [4]), i. e. the fact that a d-dimensional problem is reduced to the simpler 1 D case via recursion. Thus, the parameter d can be handled as an input parameter of the code, and all algorithmic work can be done in just one dimension. The characteristics of this unidirectional concept shall be discussed in the following.

To solve the arising linear systems, we use iterative schemes like the damped Jacobi iteration, a (preconditioned) cg-technique, or a multigrid method. Those schemes are all based on a routine to compute $S \cdot \underline{u}$ for the stiffness matrix S and arbitrary input vectors $\underline{u}$. Since this calculation of $S \cdot \underline{u}$ can be separated from the iteration itself, only the matrix-vector-product has to be implemented in our recursive data structure. To do this, we need to have a closer look at S. In a tensor product approach, d-dimensional hierarchical basis functions $\varphi_i(x_1, \ldots, x_d)$ are defined as products of 1 D hierarchical basis functions $\varphi_{i,l}(x_l)$, $1 \leq l \leq d$:

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

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

$$s_{i,j} \ = \ \sum_{k=1}^{d} \left(I_{i,j;k}^{\text{stiff}} \cdot \prod_{l \neq k} I_{i,j;l}^{\text{mass}} \right) \ , \tag{2}$$

where

$$I_{i,j;k}^{\text{stiff}} \ := \ \int_{\mathbf{\Omega_{i,k}} \cap \mathbf{\Omega_{j,k}}} \frac{\partial \varphi_{i,k}(x_k)}{\partial x_k} \cdot \frac{\partial \varphi_{j,k}(x_k)}{\partial x_k} \, dx_k \ , \tag{3}$$

$$I_{i,j;k}^{\text{mass}} \ := \ \int_{\mathbf{\Omega_{i,k}} \cap \mathbf{\Omega_{j,k}}} \varphi_{i,k}(x_k) \cdot \varphi_{j,k}(x_k) \, dx_k \ , \tag{4}$$

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

Actually, for an efficient calculation of $S \cdot \underline{u}$, we must not compute the $s_{i,j}$ themselves, since we have lost some sparsity of S due to the use of hierarchical bases, but just the sums $\sum_{j=1}^{N} s_{i,j} u_j$ for $1 \leq i \leq N$. This is done in a recursive way, such that we get all of those sums during a few passes through the data structure. In the 1 D case, we start with a vector $\underline{u}$ containing the actual solution u_i in all grid points i and make a copy $\underline{uu}$ of it. Then, with $\underline{u}$, a top-down-pass (called *down* in the following) through the data structure is done in hierarchical order, and with $\underline{uu}$, we make a bottom-up pass (called *up*). Note that, for the recursive extension, it is important to separate the two collection steps in *down* and *up*. After that, u_i contains the sum of all products $s_{i,j} u_j$ originating from grid points j hierarchically higher than i and from i itself, and uu_i contains all $s_{i,j} u_j$ from grid points j hierarchically lower than i. Finally, $\underline{u} := \underline{u} + \underline{uu}$ provides $\sum_{j=1}^{N} s_{i,j} u_j$ for each grid point i, and $\underline{u}$ now contains the product $S \cdot \underline{u}$. Thus, apart from the copy process, $S \cdot \underline{u}$ is calculated in place, and, therefore, we need only two variables per grid point or unknown, resp. The underlying 1 D algorithmic scheme of this process is shown in the upper part of Fig. 6. The recursive extension of the 1 D algorithmic principle to the general case is given by the lower part of Fig. 6. There, for $d = 2$, e. g., the grey boxes entitled *rec_ext* have to be replaced by the 1 D scheme. Note that it is important to do the recursive calls *before* the down, but *after* the up step. Concerning the storage requirement, the influence of the parameter d is very small. Since we can handle the whole process on the stack, there exist only local copies of parts of the data structure which are dominated by the copy of the d-dimensional overall structure.

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

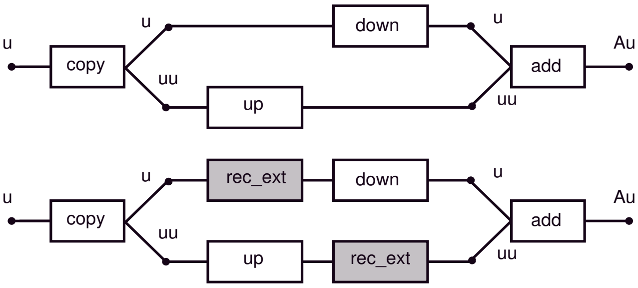


Figure 6: Scheme of the unidirectional algorithm: one-dimensional (top) and general d-dimensional case (bottom)

The following three sections deal with different extensions of the applicability of sparse grids in the p. d. e. context concerning the operators, the domains, and the type of the hierarchical basis. All techniques presented here have been implemented on the basis of the unidirectional approach.

3 Discretization of Linear Elliptic Partial Differential Equations of Second Order

In this section, we want to examine sparse grid discretizations for elliptic Dirichlet boundary value problems on the d-dimensional unit interval $\mathbf{Q} =]0,1[^d$. For a given function $f \in L_2(\mathbf{Q})$ and for the coefficient functions $A \in (L_\infty(\mathbf{Q}))^{d\times d}$, $\underline{b} \in (L_\infty(\mathbf{Q}))^d$, $c \in L_\infty(\mathbf{Q})$ describing a continuous and H_0^1-elliptic bilinear form[1]

$$\begin{aligned} a : H_0^1(\mathbf{Q}) \times H_0^1(\mathbf{Q}) &\rightarrow \mathbb{R}, \\ (u,v) &\mapsto \int_{\mathbf{Q}} \nabla u A (\nabla v)^T + \nabla u \underline{b}\, v + ucv \, d\underline{x}, \end{aligned} \tag{5}$$

we are looking for the solution $u \in H_0^1(\mathbf{Q})$ of the weak equation

$$a(u,v) = \int_{\mathbf{Q}} fv \, d\underline{x} \qquad \forall v \in H_0^1(\mathbf{Q}). \tag{6}$$

3.1 Discretization on Sparse Grids

Using the function space $S_{\tilde{G}_n^{(d)}}$ of piecewise d-linear functions on the sparse grid $\tilde{G}_n^{(d)}$ as an approximation of $H_0^1(\mathbf{Q})$, we arrive at a first finite element discretization of (6): Find the solution $u_h \in S_{\tilde{G}_n^{(d)}}$ of the discretized equation[2]

$$a(u_h, v_h) = \int_{\mathbf{Q}} \mathrm{I}^{\mathrm{lin}}_{\tilde{G}_n^{(d)}}(f) v_h \, d\underline{x} \qquad \forall v_h \in S_{\tilde{G}_n^{(d)}}, \tag{7}$$

[1] The bilinear form a is H_0^1-elliptic, iff the diffusion coefficient $A(\underline{x})$ is positive definite for almost all $\underline{x} \in \mathbf{Q}$.

[2] Since we use sparse grid discretizations only, we write u_h instead of $u_{\tilde{G}_n^{(d)}}$ and, later, a_h instead of $a_{\tilde{G}_n^{(d)}}$.

with the piecewise linear interpolation operator on sparse grids $\mathrm{I}^{\mathrm{lin}}_{\tilde{G}^{(d)}_n}$. If the coefficients of (5) are constant functions, all integrals involved can be calculated exactly using the technique of Sect. 2. But, if they are arbitrary functions, it becomes too costly in general or impossible at all to calculate the resulting stiffness matrix exactly. Thus, a second discretization step is necessary, now for the bilinear form a.

Therefore, we first introduce the *piecewise constant interpolation operator on sparse grids* (cf. [26]). The 1 D operator is given by

$$\begin{aligned}\mathrm{I}^{\mathrm{con}}_h : \mathcal{C}([0,1]) &\to \mathcal{F}([0,1]) := \{f : [0,1] \to \mathbb{R}\}\,, \\ \mathrm{I}^{\mathrm{con}}_h(f)(ih+s) &\mapsto \tfrac{f(ih)+f((i+1)h)}{2}\,,\ i \in \{0,\dots,2^n-1\}\,,\ 0 < s \le h := 2^{-n}\,, \\ \mathrm{I}^{\mathrm{con}}_h(f)(0) &\mapsto \tfrac{f(0)+f(h)}{2}\,,\end{aligned}$$

and the extension to d dimensions on regular full grids $G^{(d)}_{\underline{n}}$ with the grid size $\underline{h} = (h_1,\dots,h_d) = (2^{-n_1},\dots,2^{-n_d})$ is defined as the composition of the 1 D interpolation operators in the respective coordinate directions[3]

$$\mathrm{I}^{\mathrm{con}}_{G^{(d)}_{\underline{n}}} : \mathcal{C}(\bar{\mathbf{Q}}) \to \mathcal{F}(\bar{\mathbf{Q}})\,, \quad \mathrm{I}^{\mathrm{con}}_{G^{(d)}_{\underline{n}}} := \prod_{i=1}^{d} \mathrm{I}^{\mathrm{con}}_{h_i}\,.$$

The piecewise constant interpolation operator on the sparse grid $\tilde{G}^{(d)}_n$ then is defined via the *combination formula* (cf. [10, 18, 26])

$$\mathrm{I}^{\mathrm{con}}_{\tilde{G}^{(d)}_n} := \sum_{k=1}^{d} (-1)^{k+1} \binom{d-1}{k-1} \sum_{|\underline{m}|=n+d-k} \mathrm{I}^{\mathrm{con}}_{G^{(d)}_{2^{-\underline{m}}}}\,, \tag{8}$$

where $|\underline{m}| = |(m_1,\dots,m_d)| := \sum_{i=1}^{d} m_i$ and $2^{-\underline{m}} = (2^{-m_1},\dots,2^{-m_d})$.

A first approach for the discretization of (5) was given in [26] as

$$a_h(u_h,v_h) := \int_{\mathbf{Q}} \nabla u_h \mathrm{I}^{\mathrm{con}}_{\tilde{G}^{(d)}_n}(A)(\nabla v_h)^T + \nabla u_h \mathrm{I}^{\mathrm{con}}_{\tilde{G}^{(d)}_n}(\underline{b})\, v_h + u_h \mathrm{I}^{\mathrm{con}}_{\tilde{G}^{(d)}_n}(c) v_h \, d\underline{x}\,, \tag{9}$$

where the coefficient functions of a are interpolated with piecewise constant functions on the sparse grid $\tilde{G}^{(d)}_n$.[4] This discretization is easy to handle for theoretical results: In [26], a $O(h|\log h|^{(d-1)})$-convergence in the H^1-norm and a $O(h^{1+1/d}|\log h|^{(d-1)})$-convergence in the L_2-norm have been proved. But till now, there are no implementations for $d > 2$. This is due to the fact that the occurring integrals have to be calculated over products of three functions ($\int ucv\, d\underline{x}$, e. g.), instead of over two functions as for the Laplace operator in Sect. 2. Thus, the algorithmic situation has become much harder. In order to get back to the well-known situation of Sect. 2, we introduce a *mixed interpolation operator on sparse grids*: For

---

[3]Note that the full grid $G^{(d)}_n$ has the same grid size 2^{-n} in each direction, whereas the full grid $G^{(d)}_{\underline{n}}$ has the grid size 2^{-n_i} in the i-th coordinate direction.

[4]The interpolation of vector- and matrix-valued functions is to be read componentwise.

a given coordinate direction x_i, $i \in \{1,\ldots,d\}$, we define the mixed interpolation operator on regular full grids with the grid size $\underline{h} = (h_1,\ldots,h_d) = (2^{-n_1},\ldots,2^{-n_d})$ as

$$\mathrm{I}^{x_i\mathrm{con,lin}}_{G^{(d)}_{\underline{n}}} : \mathcal{C}(\bar{\mathbf{Q}}) \to \mathcal{F}(\bar{\mathbf{Q}}) \,, \quad \mathrm{I}^{x_i\mathrm{con,lin}}_{G^{(d)}_{\underline{n}}} := \prod_{k=1}^{i-1} \mathrm{I}^{\mathrm{lin}}_{h_k} \circ \mathrm{I}^{\mathrm{con}}_{h_i} \circ \prod_{k=i+1}^{d} \mathrm{I}^{\mathrm{lin}}_{h_k}$$

and on the sparse grid $\tilde{G}^{(d)}_n$, again, via the combination formula

$$\mathrm{I}^{x_i\mathrm{con,lin}}_{\tilde{G}^{(d)}_n} := \sum_{k=1}^{d} (-1)^{k+1} \binom{d-1}{k-1} \sum_{|\underline{m}|=n+d-k} \mathrm{I}^{x_i\mathrm{con,lin}}_{G^{(d)}_{\underline{n}}} \,. \tag{10}$$

Now, we can formulate a second discretization:

$$\begin{aligned} a_h(u_h, v_h) \;\; := \int\limits_{\mathbf{Q}} \; \Bigg\{ & \sum_{i,j=1}^{d} \mathrm{I}^{x_i\mathrm{con,lin}}_{\tilde{G}^{(d)}_n}(u_{h,x_i} A_{ij})\, v_{h,x_j} \\ & + \sum_{i=1}^{d} \mathrm{I}^{x_i\mathrm{con,lin}}_{\tilde{G}^{(d)}_n}(u_{h,x_i} b_i)\, v_h + \mathrm{I}^{\mathrm{lin}}_{\tilde{G}^{(d)}_n}(u_h\, c) v_h \Bigg\} d\underline{x} \,. \end{aligned} \tag{11}$$

Here, the products of the ansatz function u_h and the coefficient functions are treated as a single function to be interpolated: the products of $u_{h,x_i} A_{ij}$ and $u_{h,x_i} b_i$ are approximated in the same function space as u_{h,x_i} itself, and the product $u_h\, c$ in the same function space as u_h. Thus, this discretization results in integrals over products of only two functions.

But, it has to be noted that, due to the different treatment of u_h and v_h in $a_h(u_h, v_h)$, for a symmetric bilinear form a discretization (11) produces an unsymmetric stiffness matrix. If all coefficients are constant except the Helmholtz coefficient c, the discretization can be symmetrized (cf. [23]). Also, whereas the joint approximations of the products are advantageous for the algorithmic situation, they are disadvantageous for the theoretical treatment. Up to now, we only proved an estimation of the distance between the exact bilinear form a and the approximated bilinear form (11):

$$|a(u_h, v_h) - a_h(u_h, v_h)| \le h|\log h|^{(d-1)} C ||v_h||_{H^1} \max_{1\le i\le d} \{||u_{h,x_i}||_\infty, ||u_h||_\infty\} \,.$$

Using Strang's first Lemma, this promises a $O(h|\log h|^{(d-1)})$-convergence in the H^1-norm, but the uniform ellipticity of a_h ($\exists \alpha > 0 : \;\; a_h(v_h, v_h) \ge \alpha \|v_h\|^2_{H^1} \;\; \forall v_h \in S_{\tilde{G}^{(d)}_n}$) as an important condition is still an open problem.

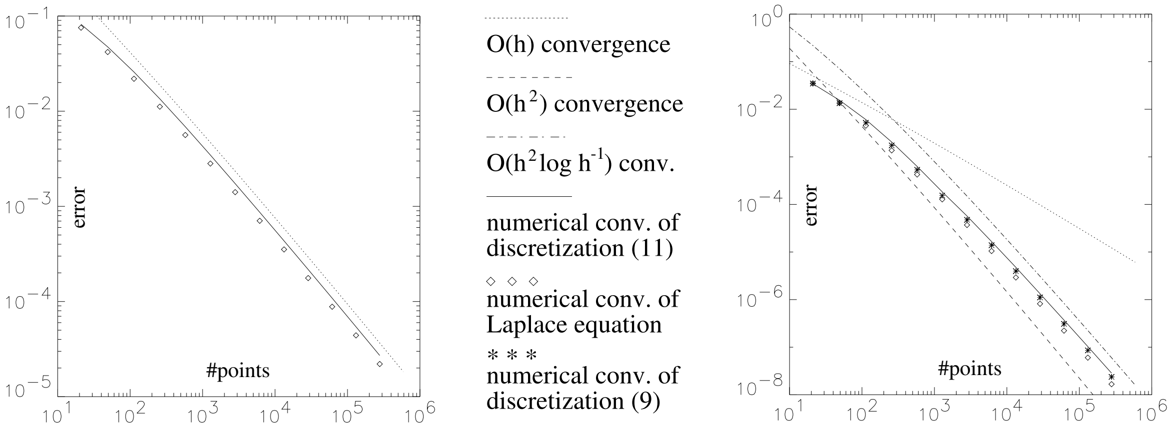
3.2 A Numerical Experiment

We consider the homogeneous diffusion equation

$$\nabla \cdot \big(A(\underline{x})(\nabla u)^T\big) = 0 \quad \text{in } \;]0,1[^2 \tag{12}$$

with the variable diffusion coefficient

$$A(\underline{x}) = \Big(\cos(\pi x_1) \cosh(\pi x_2)/\cosh(\pi) + 2 \Big) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



H^1-error $\qquad\qquad\qquad\qquad$ L_∞-error

Figure 7: Numerical results on regular sparse grids for both the Laplace equation and (12) using the discretizations (9) and (11)

and the solution

$$u(x_1, x_2) = \sin(\pi x_1) \sinh(\pi x_2) / \sinh(\pi) \,. \tag{13}$$

Note that this function u also solves the Laplace equation $\triangle u = 0$.

The numerical results for both the Laplace equation and (12) are shown in Fig. 7, where regular sparse grids of increasing depth are used. On the left-hand side we see the convergence results of the H^1-error for the discretization (11) of (12) and the Laplace equation, on the right-hand side those of the L_∞-error, and, additionally, the convergence results for the discretization (9) of (12). In comparison with the reference curves one can observe an $O(h)$-convergence of the H^1-error and an $O(h^2 \log h^{-1})$-convergence of the L_∞-error. Moreover, in spite of the additional simplification of discretization (11) in comparison with (9), the results are nearly the same.

4 Treatment of Boundary Fitted Coordinates

Now, we want to profit from the discretization of the general elliptic operator (5) of the preceding section in order to deal with more general geometries. This can be done using domain transformation techniques. Let $\mathbf{P} \subset \mathbb{R}^d$ be a simply connected domain[5] – the so-called *physical domain*. Then, we are interested in the solution $u \in H_0^1(\mathbf{P})$ of (6) on $\mathbf{P}$.

4.1 The Governing Equation in Generalized Coordinates

To be able to treat the equation on the curvilinear bounded domain $\mathbf{P}$, we introduce a new coordinate system – the *generalized* or *boundary fitted coordinates*. Thus, first, we have to find a $\mathcal{C}^1$-diffeomorphism ($\mathcal{C}^1$-invertible mapping) $\underline{\psi} : \bar{\mathbf{Q}} \to \bar{\mathbf{P}}$, $\underline{x} \mapsto \underline{\xi} = \underline{\psi}(\underline{x})$, which maps the unit interval $\bar{\mathbf{Q}} = [0,1]^d$ – the so-called *computational domain* – one-to-one onto the physical domain $\bar{\mathbf{P}}$. The coordinate system $\underline{x}$ in $\mathbf{Q}$ is called the *generalized* coordinate system, and a coordinate line in $\bar{\mathbf{Q}}$ is mapped onto a curvilinear coordinate line in $\bar{\mathbf{P}}$ (cf.

[5]We consider a *domain* as an open connected set. As we use in $\mathbf{Q}$ the coordinates $\underline{x}$ (and (x,y), (x,y,z) in 2 D, 3 D, resp.), we use in $\mathbf{P}$ the coordinates $\underline{\xi}$ (and (ξ,η), (ξ,η,ζ) in 2 D, 3 D, resp.).

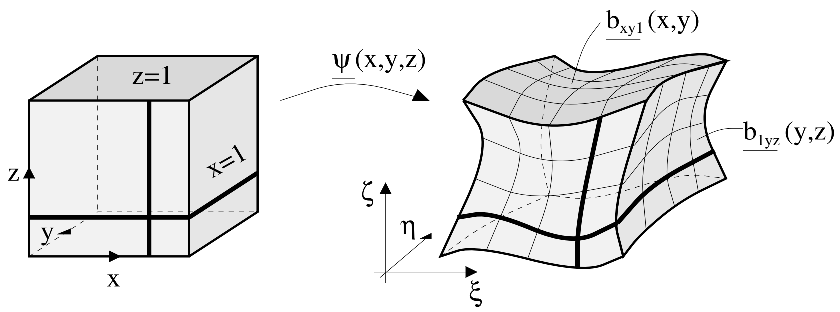


Figure 8: Coordinate transformation in three spatial dimensions

Fig. 8). Thus, we can identify the generalized coordinate system in $\bar{\mathbf{Q}}$ with a curvilinear coordinate system in $\bar{\mathbf{P}}$.

Now, we want to solve the transformed equation

$$a^*(\hat{u}, \hat{v}) = \int_{\mathbf{Q}} f^* \hat{v}\, d\underline{x} \qquad \forall \hat{v} \in H_0^1(\mathbf{Q}) \tag{14}$$

with the new bilinear form

$$\begin{aligned} a^* : H_0^1(\mathbf{Q}) \times H_0^1(\mathbf{Q}) &\rightarrow \mathbb{R}, \\ (\hat{u}, \hat{v}) &\mapsto \int_{\mathbf{Q}} \nabla \hat{u} A^* (\nabla \hat{v})^T + \nabla \hat{u}\, \underline{b}^*\, \hat{v} + \hat{u} c^* \hat{v}\, d\underline{x}\,. \end{aligned}$$

Here, we use the notation and new coefficients

$$\begin{aligned} \hat{g}(\underline{x}) &:= g \circ \underline{\psi}(\underline{x})\,, \quad \text{a composed function}\,, \\ A^*(\underline{x}) &:= J_{\underline{\psi}}^{-1}(\underline{x})\, (A \circ \underline{\psi}(\underline{x}))\, J_{\underline{\psi}}^{-T}(\underline{x})\, |\det J_{\underline{\psi}}(\underline{x})|\,, \\ \underline{b}^*(\underline{x}) &:= J_{\underline{\psi}}^{-1}(\underline{x})\, (\underline{b} \circ \underline{\psi}(\underline{x}))\, |\det J_{\underline{\psi}}(\underline{x})|\,, \\ c^*(\underline{x}) &:= (c \circ \underline{\psi}(\underline{x}))\, |\det J_{\underline{\psi}}(\underline{x})|\,, \quad f^*(\underline{x}) := (f \circ \underline{\psi}(\underline{x}))\, |\det J_{\underline{\psi}}(\underline{x})|\,, \\ J_{\underline{\psi}} &= (\frac{\partial \psi_i}{\partial x_j})_{i,j=1,..,d}\,, \quad \text{the Jacobian matrix of } \underline{\psi}\,, \end{aligned} \tag{15}$$

(cf. [12, 24]). Equation (14) now is formulated on $\mathbf{Q}$, and an approximation $\hat{u}_h$ on sparse grids of its solution $\hat{u} \in H_0^1(\mathbf{Q})$ can be calculated numerically using the discretization (11) of the preceding section. Then, the approximation of the solution of the original problem is provided by $u_h := \hat{u}_h \circ \underline{\psi}^{-1}$.

4.2 Grid Generation on Sparse Grids

The remaining problem is to find a suitable mapping $\underline{\psi} : \bar{\mathbf{Q}} \rightarrow \bar{\mathbf{P}}$ which maps the d-dimensional unit interval $\bar{\mathbf{Q}}$ onto the physical domain $\bar{\mathbf{P}}$. Since we have focused on algebraic mapping techniques for the 2 D case in [12], here, we want to extend our considerations to algebraic mapping techniques for the 3 D (d D) case and elliptic grid generation techniques.

Let the geometric description of the physical domain $\bar{\mathbf{P}}$ be given in terms of its boundary, i. e. for the 2 D case in terms of its four parametrized boundary lines

$$\begin{aligned} \underline{b_{x0}} &: [0,1] \to \partial\mathbf{P}\,,\ x \mapsto \underline{b_{x0}}(x)\,, \quad \underline{b_{x1}} : [0,1] \to \partial\mathbf{P}\,,\ x \mapsto \underline{b_{x1}}(x)\,, \\ \underline{b_{0y}} &: [0,1] \to \partial\mathbf{P}\,,\ y \mapsto \underline{b_{0y}}(y)\,, \quad \underline{b_{1y}} : [0,1] \to \partial\mathbf{P}\,,\ y \mapsto \underline{b_{1y}}(y)\,, \end{aligned} \tag{16}$$

and for the 3 D case analogously in terms of its six parametrized boundary surfaces $\underline{b_{xy0}}(x,y)$, $\underline{b_{xy1}}(x,y)$, $\underline{b_{x0z}}(x,z)$, $\underline{b_{x1z}}(x,z)$, $\underline{b_{0yz}}(y,z)$, and $\underline{b_{1yz}}(y,z)$. Then in 2 D, a mapping $\underline{\psi} : \bar{\mathbf{Q}} \to \bar{\mathbf{P}}$ has to be found with the boundary conditions

$$\begin{aligned} \underline{\psi}(x,0) &= \underline{b_{x0}}(x)\,, \quad \underline{\psi}(x,1) = \underline{b_{x1}}(x) \quad \forall x \in [0,1]\,, \\ \underline{\psi}(0,y) &= \underline{b_{0y}}(y)\,, \quad \underline{\psi}(1,y) = \underline{b_{1y}}(y) \quad \forall y \in [0,1]\,. \end{aligned} \tag{17}$$

The case of three spatial dimensions can again be treated analogously.

4.2.1 Transfinite Interpolation.

One of the most popular algebraic mapping techniques is the *transfinite interpolation* (cf. [13, 14]). In 2 D, this method is defined by

$$\begin{aligned} \underline{\psi}(x,y) := \ & \Big\{(1-y)\,\underline{b_{x0}}(x) + y\,\underline{b_{x1}}(x)\Big\} + \Big\{(1-x)\,\underline{b_{0y}}(y) + x\,\underline{b_{1y}}(y)\Big\} \\ & -\Big\{(1-y)(1-x)\,\underline{b_{0y}}(0) + (1-y)x\,\underline{b_{1y}}(0) \\ & \qquad + y(1-x)\,\underline{b_{0y}}(1) + yx\,\underline{b_{1y}}(1)\Big\}\,, \end{aligned} \tag{18}$$

and in 3 D, the corresponding formula consists of 26 summands.

In our sparse grid context, this interpolation method is intrinsic, since it is equivalent to the d-linear hierarchical interpolation from the boundary. Thus, there is no need for an additional implementation of higher dimensional transfinite interpolation formulas. A further advantage is that grid controlling techniques like the multisurface method in [12] easily can be included. In Fig. 9, the folding grid (a) has been improved by prescribing the location of the interior grid line $\underline{\psi}(x = 1/2, y)$. The same grid as in Fig. 9 (b) can be achieved by using block structuring.

The drawback of such controlling techniques is that the Jacobian matrix $J_{\underline{\psi}}$ becomes discontinuous due to bucklings in the coordinate lines in $\mathbf{P}$. Using interpolation methods, discontinuities in $J_{\underline{\psi}}$ also arise if there are bucklings at the boundary (cf. Fig. 15).

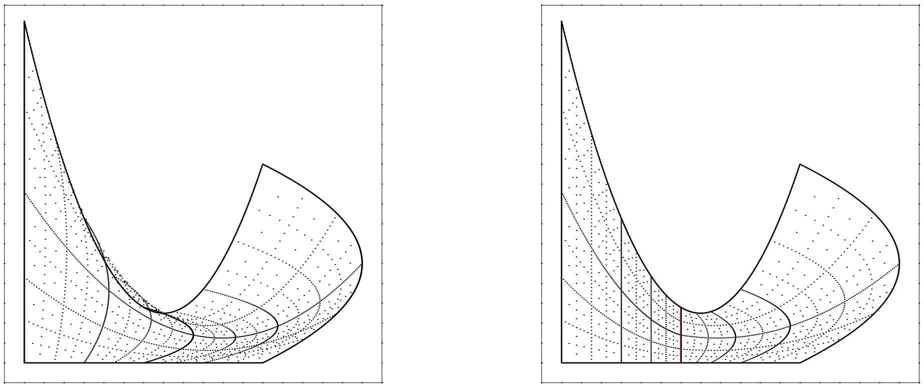
4.2.2 Numerical Experiments.

As a first example, we consider the equation

$$-\nabla \cdot (A(\nabla u)^T) + \nabla u\,\underline{b} + cu = f \tag{19}$$

on the 3 D domain $\mathbf{P}$ shown in Fig. 10. The grid on $\mathbf{P}$ was generated by the transfinite interpolation method. The coefficients are

$$A(\underline{\xi}) = \begin{pmatrix} 4+\sin(2\pi\xi_1)+\sin(2\pi\xi_2) & \sin(\pi\xi_1\xi_2\xi_3)/2 & \xi_1\xi_2 \\ \sin(\pi\xi_1\xi_2\xi_3)/2 & 4+\sin(2\pi\xi_1\xi_2) & 0 \\ \xi_1\xi_2 & 0 & \begin{array}{c} 4+\sin(2\pi\xi_1)+\sin(2\pi\xi_2) \\ +\sin(2\pi\xi_3) \end{array} \end{pmatrix},$$



(a) (b)

Figure 9: A folding grid produced by transfinite interpolation (left) and an improved grid by prescribing the location of the interior grid line $\underline{\psi}(x = 1/2, y)$ (right)

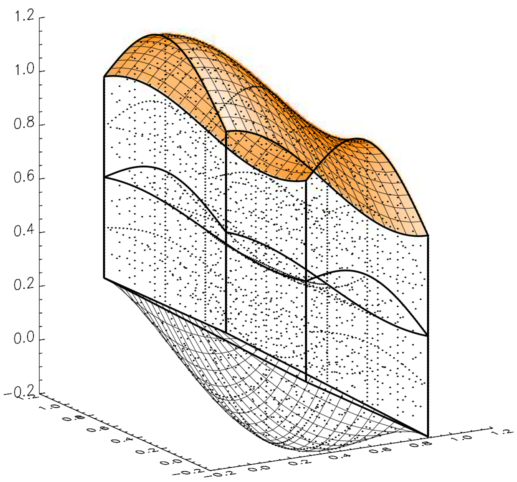
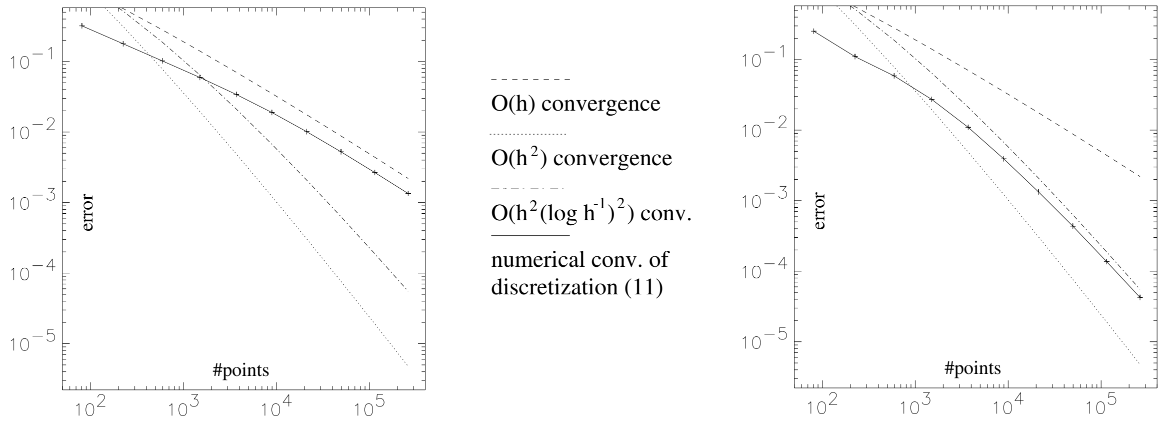


Figure 10: Physical domain $\mathbf{P}$ of problem (19)

$$\underline{b}(\underline{\xi}) = \begin{pmatrix} 1 \\ 2 + 4(1 - 2\xi_2)\sin(2\pi\xi_1) \\ 3 + 4(1 - 2\xi_1)\cos(2\pi\xi_2\xi_3) \end{pmatrix}, \quad c(\underline{\xi}) = 7 + \sqrt{1 + e^{\xi_1+\xi_2+\xi_3}},$$

and the right-hand side f and the Dirichlet boundary conditions are chosen in such a way that $u(\underline{\xi}) = \sin(\pi\xi_1)\sin(\pi\xi_2)\sin(\pi\xi_3)$ solves (19). In Fig. 11, the numerical results for the H^1- and L_∞-error of discretization (11) are shown.

In the second numerical experiment, we consider the 2 D Laplace equation on $\mathbf{P}$ as in Fig. 12, left, with a sigularity at the boundary point (0,1/2). The boundary conditions are chosen in such a fashion that $u(\xi_1, \xi_2) = \mathrm{Re}(\sqrt{\xi_1 + i\xi_2})$ solves this equation (cf. Fig. 12, middle). The numerical grid on $\mathbf{P}$ was generated by transfinite interpolation, which provides



H^1-error

L_∞-error

Figure 11: Numerical results on regular sparse grids for (19) by discretization (11)

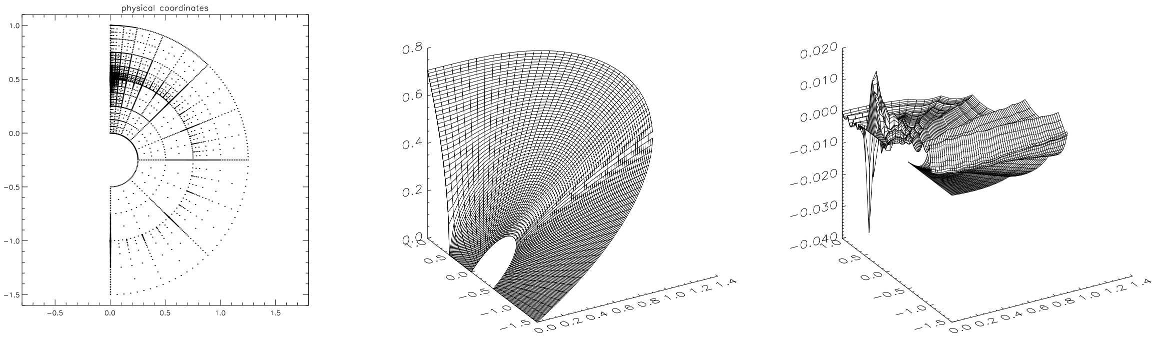


Figure 12: Adaptive grid with 5237 grid points (left), the solution function u (middle), and the error function on a coarser grid using 151 grid points (right) for the Laplace equation with a sigularity on the boundary

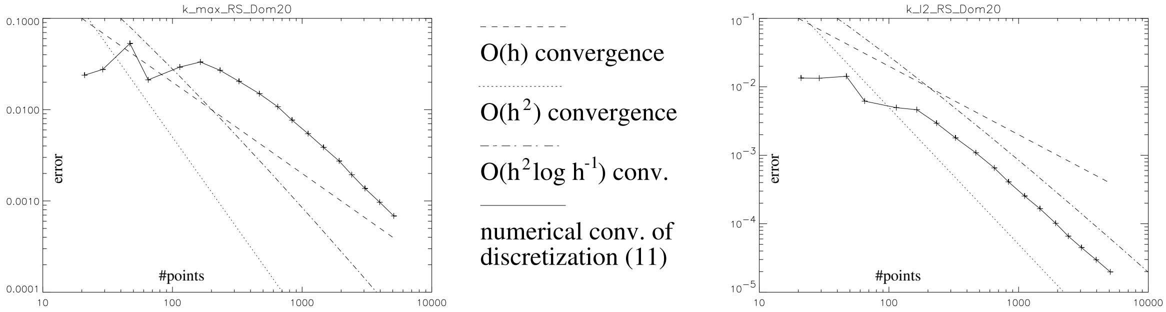
a new diffusion coefficient A^* with nonzero components only in the diagonal:

$$A^*(x_1, x_2) = \begin{pmatrix} 4/\pi(1+4x_2) & 0 \\ 0 & \pi(1+4x_2)/4 \end{pmatrix}.$$

The approximate solution u_h was calculated on adaptive sparse grids with decreasing bounds ε for the hierarchical surplusses of the approximate solution (minimizing the L_2- and L_∞-error). This results in an adaptive grid refinement near the singularity, where the error function $u - u_h$ has its maximum (cf. Fig. 12, right). In Fig. 13, the $O(h^2 \log h^{-1})$ behaviour of both the L_2- and the L_∞-error can be seen.

4.2.3 Elliptic Grid Generation.

To avoid such discontinuities as they were mentioned in the section about transfinite interpolation and to gain smooth interior grids, one often uses elliptic grid generation. The simplest elliptic grid generator is the so-called AH grid generator based on the Dirichlet boundary value problem for the Laplace equation $\triangle\underline{\psi} = \triangle\underline{\xi} = 0$, with boundary conditions like (17) (cf. [1]). But, for nonconvex domains $\bar{\mathbf{P}}$, the generated grids are folded in general. To ensure nonfolding grids (i. e. a bijective mapping $\underline{\psi}$) the Winslow generator can be used (cf. [28]).



L_∞-error $\qquad$ L_2 error

Figure 13: Numerical results on adaptive sparse grids for the Laplace equation with a sigularity on the boundary by discretization (11)

Here, the inverse mapping $\underline{\psi}^{-1}$ is required to be harmonic, i. e. $\triangle\underline{\psi}^{-1} = \triangle\underline{x} = 0$. Since the computational domain $\mathbf{Q}$ is convex, Rado's Theorem from the theory of harmonic mapping guarantees in the 2 D case that the mapping $\underline{\psi}$ itself is a diffeomorphism (cf. [25], e. g.). But, for the numerical calculation, the above equation has to be inverted, which results in a quasi-linear and coupled system of d equations for $\underline{\psi} = (\psi_1, \ldots, \psi_d)^T$ (cf. [28]). This nonlinearity and coupling is the major drawback of this (often used) method, because substantially more work has to be invested in the numerical solution of these equations. Further, a guarantee for nonfolding grids can only be given for 2 D domains, because an extension of Rado's Theorem to domains of higher spatial dimensions is not known. Additionally, grid controlling techniques are based on special control functions on the right-hand side of the elliptic equations above, and they often require user intervention, experience, and skill (cf. [24]).

These problems in generating smooth nonfolding grids in the interior of a domain and the lack of user-friendly grid controlling techniques for elliptic grid generators motivate the introduction of a *hybrid method* which combines the easy control mechanism of interpolation methods with the smooth grids and low costs (in relation to the Winslow generator) of an AH elliptic grid generator.

We start with an arbitrary given $\mathcal{C}^2$-diffeomorphism $\underline{\psi} = (\psi_1, \ldots, \psi_d)^T : \mathbf{Q} \to \mathbf{P}$ which maps the computational domain $\mathbf{Q}$ onto $\mathbf{P}$. Then, the coordinates $\psi_i(\underline{x})$, $i \in \{1, \ldots, d\}$, are solutions of the elliptic equations in divergence form[6]

$$\nabla_{\underline{x}} \cdot (|\det(J_{\underline{\psi}})| J_{\underline{\psi}}^{-1} J_{\underline{\psi}}^{-T} (\nabla_{\underline{x}} \psi_i)^T) = 0 \quad \text{on } \mathbf{Q} =]0,1[^d, \quad i \in \{1, \ldots, d\}\,. \tag{20}$$

Concerning the transformation formula (15), these equations simply are the transformations of the trivial equations $\triangle_{\underline{\xi}}\underline{\psi} = (\triangle_{\underline{\xi}}\xi_i, \ldots, \triangle_{\underline{\xi}}\xi_d) = \underline{0}$ on $\mathbf{P}$ to the according equations on $\mathbf{Q}$.

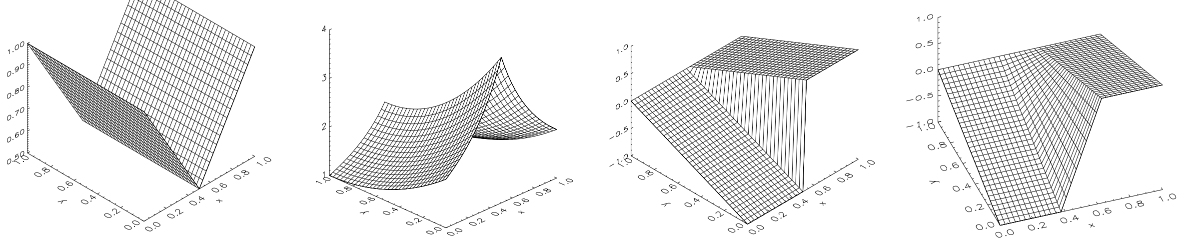
An important implication of (20) is that every grid generation method is equivalent to a certain elliptic method. I. e., for a given mapping $\underline{\psi} : \bar{\mathbf{Q}} \to \bar{\mathbf{P}}$, we get the Dirichlet problem

$$\nabla_{\underline{x}} \cdot (A^*(\nabla_{\underline{x}} \varphi_i)^T) = 0\,, \quad i \in \{1, \ldots, d\}\,, \ \text{ in } \mathbf{Q} =]0,1[^d\,, \tag{21a}$$

with the boundary conditions

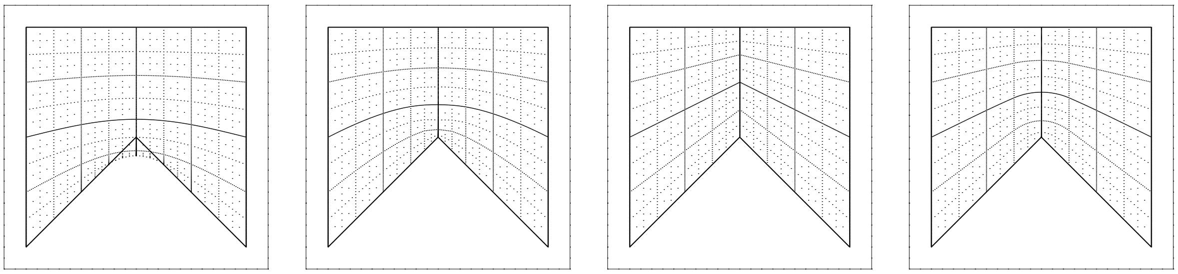
$$\underline{\varphi}(x_1, .., \bar{x}_i, .., x_d) = \underline{\psi}(x_1, .., \bar{x}_i, .., x_d)\,, \ \ i \in \{1, \ldots, d\}\,, \ \ \bar{x}_i \in \{0, 1\} \tag{21b}$$

[6]For clarity, we use the notations $\nabla_{\underline{x}} = (\partial/\partial x_1, \ldots, \partial/\partial x_d)$ and $\triangle_{\underline{\xi}} = \partial^2/\partial\xi_1^2 + \ldots + \partial^2/\partial\xi_d^2$.



$A^*_{11} = \det(J_{\underline{\psi}})$ A^*_{22} $A^*_{12} = A^*_{21}$ smoothed $A^*_{12} = A^*_{21}$

Figure 14: The component functions of A^* in (22)



(a) (b) (c) (d)

Figure 15: Comparison: (a) AH elliptic grid, (b) Winslow elliptic grid, (c) transfinite interpolation grid, (d) hybrid grid

and the diffusion coefficient

$$A^* = |\det(J_{\underline{\psi}})| J_{\underline{\psi}}^{-1} J_{\underline{\psi}}^{-T}. \tag{21c}$$

For the solution $\underline{\varphi}$ of this problem, $\underline{\varphi} \equiv \underline{\psi}$ holds. If we fix the boundary conditions (21b), the new coordinates $\underline{\varphi}$ only depend on A^*, and we get a starting point for grid controlling via the interpolation method, e. g. But, also for nonsmooth gridlines we get a starting point to improve the grid. In (21a), bucklings in the coordinate lines are coupled directly with discontinuities in A^* of (21c). Thus, to reach a smooth coordinate system $\underline{\varphi}$ from (21a), we have to smooth the diffusion coefficient A^*.

As an example, we consider the physical domain $\mathbf{P}$ shown in Fig. 15. Then, the transfinite interpolation (18) results in the discontinuous coefficient matrix

$$A^* = \begin{pmatrix} \det(J_{\underline{\psi}}) & (1-y)\operatorname{sign}(\frac{1}{2}-x) \\ (1-y)\operatorname{sign}(\frac{1}{2}-x) & \dfrac{(1-y)^2+1}{\det(J_{\underline{\psi}})} \end{pmatrix}, \ \det(J_{\underline{\psi}}) = \frac{1}{2} + |x - \frac{1}{2}|. \tag{22}$$

The component functions of A^* are shown in Fig. 14, and one can see that the buckling in the coordinate lines $\underline{\psi}(x, y = \text{const})$ results in discontinuous partial derivatives in x-direction of the components A^*_{11} and A^*_{22} and in the component functions $A^*_{12} = A^*_{21}$ themselves. To reach a $\mathcal{C}^1$-coordinate system $\underline{\varphi}$, we smooth the components $A^*_{12} = A^*_{21}$ to continuous (and piecewise linear) functions (cf. Fig. 14, right). This results in the grid shown in Fig. 15 (d).

5 Hierarchical Bases of Higher Order

Up to now, we have only dealt with piecewise multilinear and piecewise constant hierarchical bases. In the following, we introduce hierarchical polynomial bases of piecewise arbitrary degree in each coordinate direction that allow us to combine the efficiency of sparse grids (which, in some sense, can be seen as a method of higher order themselves) and their intrinsic h-adaptivity (cf. Sect. 1) with the improved approximation qualities of higher order basis functions. Thus, in spite of a quite different approach, there are close relations to the p- and h-p-versions of the finite element method [2, 3, 19, 20, 31].

5.1 Construction and Properties of the Hierarchical Bases

A first step towards higher order techniques on sparse grids has been done by Störtkuhl [27]. There, the main emphasis is put on the solution of the 2 D biharmonic equation $\Delta^2 u = 0$. In order to get $\mathcal{C}^1$-elements, a piecewise cubic hierarchical Hermite basis is defined. Since the values of both the function and of its first derivative have to be fixed, this approach leads to 2^d degrees of freedom per grid point for the general d-dimensional case.

In [9], the hierarchical Lagrangian interpolation, an alternative choice of hierarchical polynomials based on $\mathcal{C}^0$-elements with still one degree of freedom per grid point, has been presented and studied in detail for $d = 2$ and $p = 2$. Now, we want to extend the principle of the hierarchical Lagrangian interpolation to arbitrary values of d and p. Since our tensor product approach provides us with d-dimensional functions if 1 D functions are defined, we can restrict ourselves to this simpler case for the following explanations.

Obviously, there is exactly one quadratic polynomial φ_i that fulfils

$$\varphi_i(x_i) = 1, \quad \varphi_i(x_i - h_i) = \varphi_i(x_i + h_i) = 0 \tag{23}$$

and that can be used, consequently, as a quadratic hierarchical basis function in grid point x_i with the support $[x_i - h_i, x_i + h_i]$. But, for some φ_i of a degree $p > 2$, additional degrees of freedom must be fixed. For the construction of a spline interpolant, e. g., those degrees of freedom are invested in more smoothness. However, since we want to work with $\mathcal{C}^0$-elements, we use *that* φ_i with support $[x_i - h_i, x_i + h_i]$ and $\varphi_i(x_i) = 1$ that is part of the polynomial of degree p with zeroes at the support's two boundary points and at the $p-2$ next direct hierarchical ancestors x_k of x_i, with respect to the hierarchical ordering of the grid points or associated subspaces (cf. Figs. 1, 3, and 4). Note that those ancestors x_k are situated *outside* the support; they are only used to construct φ_i. Consequently, we get just one degree of freedom in each grid point, and there is only a subspace of the space of all piecewise polynomials of degree p that can be represented.

Figure 16 illustrates the quartic case. Here, the actual grid point is $x_i = -0.75$, and, due to the hierarchical relations of the points given by the right-hand side of Fig. 16, the four ancestors used for the hierarchical interpolation are $-1, -0.5, 0$, and 1. The left-hand part of Fig. 16, finally, shows the whole resulting interpolant (dotted line) and the part of it that is afterwards used as the hierarchical basis function in x_i (solid line).

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

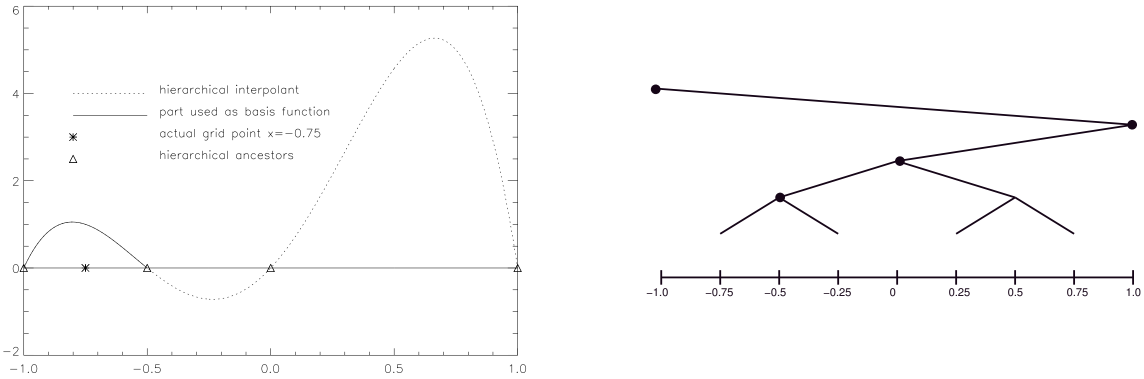


Figure 16: Construction of a hierarchical basis function of degree 4 by hierarchical Lagrangian interpolation (left) and corresponding hierarchical structure of the grid points (right)

general, 2^{p-2} of degree p. Figure 17 shows the basis functions for $p = 2$ and $p = 3$, Fig. 18 illustrates the situation for $p = 4$, where we get two pairs of symmetric polynomials.

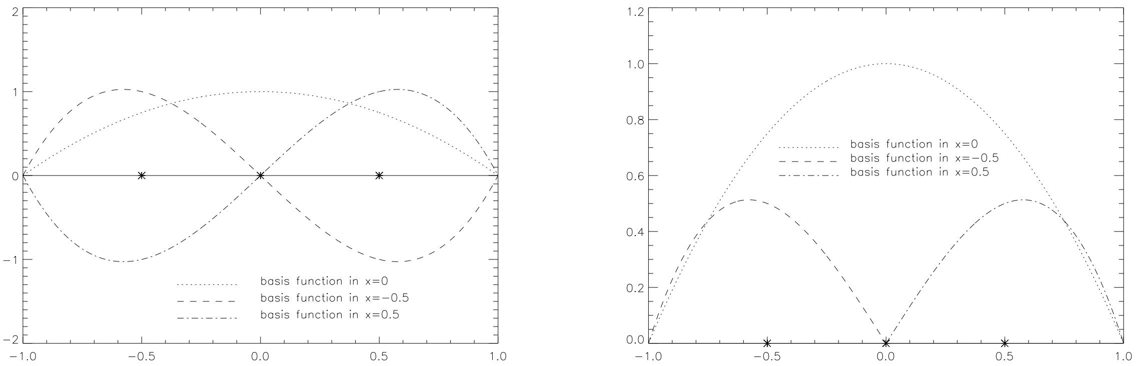
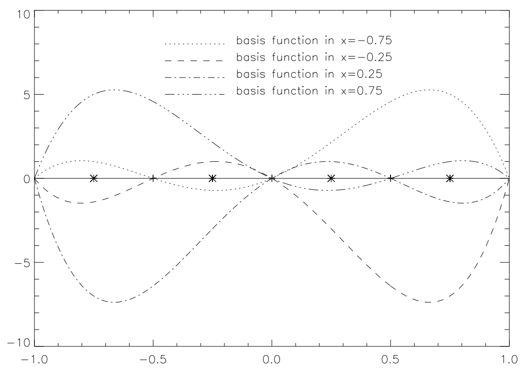


Figure 17: Hierarchical basis functions for $p = 2$ and $p = 3$ (different scaling for reasons of clarity): construction via hierarchical interpolation (left) and used restriction to the respective hierarchical support (right)

Second, though our approach is based on a simple Lagrangian interpolation without any influence on the position of the respective nodal points, we get no numerical problems due to oscillations, since only a small and uncritical part of the resulting interpolant is taken into account. As it can be seen in Fig. 19, the shape of the different basis functions does not change that much even for larger p. I. e., there are only slight changes in the basis functions when we switch from p to $p + 1$, e. g. Therefore, we can expect an only moderate influence of p on the condition of the stiffness matrix (cf. the discussion and Fig. 21 in Sect. 5.2).

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



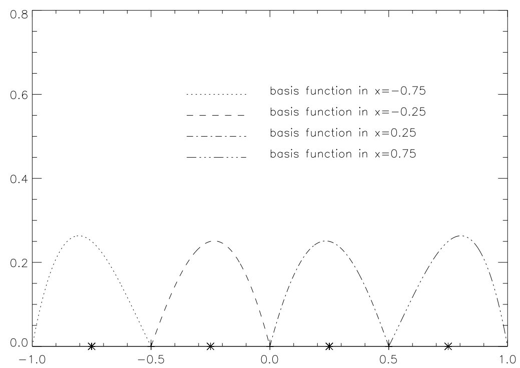


Figure 18: Hierarchical basis functions for $p = 4$ (different scaling for reasons of clarity): construction via hierarchical interpolation (left) and used restriction to the respective hierarchical support (right)

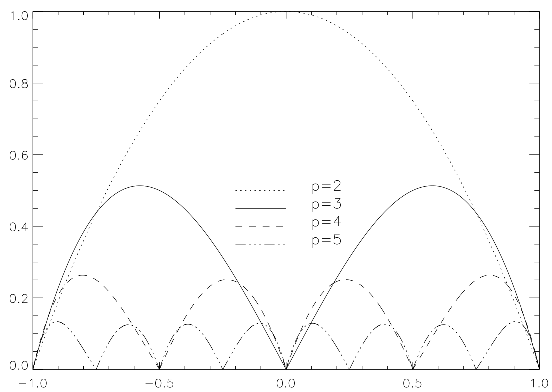


Figure 19: Resulting hierarchical basis for $G_4^{(1)}$ and $p_{\max} = 5$ (different scaling)

between our approach and p-version-type methods.

Now, let us turn to some aspects of the implementation of our method that clearly show its efficiency. First of all, there are no changes in the data structure in comparison with the piecewise multilinear case. We still only store the hierarchical coefficient in each grid point, which, again, leads to an overall storage requirement of about $2 \cdot M$, where M denotes the number of grid points and $M = O(N(\log_2(N))^{d-1})$ (cf. Sects. 1 and 2). The algorithmic handling of the different polynomials is done in 1 D procedures only and is organized on the stack. I. e., there are local vectors of polynomial coefficients or of integrals of the length $p + 1$ which are passed on during the recursive *up* and *down* procedures described in Sect. 2. Thus, for a sparse grid $\tilde{G}_n^{(d)}$ of depth n and a fixed maximum degree $p_{\max}$, there is only an additional storage of $O(n \cdot (p_{\max} + 1)) = O(\log_2(N) \cdot (p_{\max} + 1))$ for the stack. Since we use a classical Taylor representation $\underline{a} \in \mathbb{R}^{p_{\max}+1}$ for all occurring basis functions and local interpolants $f(x)$,

$$f(x) = \sum_{k=0}^{p_{\max}} a_k \cdot \frac{x^k}{k!} \ , \tag{24}$$

an adaptive treatment of p can be easily achieved by omitting the leading coefficient a_p or by taking into account a new a_{p+1}, resp., if suitable criteria for such a process are developed. However, up to now, no experiments with an adaptive handling of p have been done. Finally, in order to avoid repeated calculations with our basis polynomials, we compute the Taylor coefficients of each basis function φ_i and the integrals

$$\int_{\mathrm{supp}(\varphi_i)} \varphi_i(x) \frac{x^k}{k!}\, dx \ , \ \ 0 \leq k \leq p_{\max} \ , \tag{25}$$

during a setup phase before the iteration. The values (25) are important for a cheap calculation of the occurring finite element integrals. Due to the 2^{p-2} different basis polynomials of degree p, we get an additional storage requirement of $O(2^{p_{\max}} \cdot (p_{\max}+1))$. Since $p_{\max} \leq n+1$ for $\tilde{G}_n^{(d)}$ due to the hierarchical Lagrangian approach and since $p_{\max} \ll n$ for reasonable applications with bigger n, both terms $n \cdot (p_{\max}+1)$ and $2^{p_{\max}} \cdot (p_{\max}+1)$ depending on $p_{\max}$ are significantly smaller than the about $2 \cdot M$ variables for the data structure itself. Consequently, there is only a slight influence of the polynomial degree on the overall storage requirement.

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

$$\underline{a}^{(\mathrm{son})} := T \cdot D \cdot \underline{a}^{(\mathrm{father})} \ , \tag{26}$$

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

$$\underline{a}^{(\mathrm{father})} := D \cdot T^T \cdot \underline{a}^{(\mathrm{son})} \ , \tag{27}$$

where $D \in \mathbb{R}^{(p+1)\times(p+1)}$ represents a diagonal scaling, and $T \in \mathbb{R}^{(p+1)\times(p+1)}$ is an upper triangular Toeplitz matrix. Note that both D and T are constant matrices that do neither depend on the actual grid point nor on the hierarchical level. From (26) and (27), it follows that the number of arithmetic operations is of the order $O(p^2)$ in each grid point. Thus, if we work with a fixed maximum degree $p_{\max}$, we get a total of $O(M \cdot p_{\max}^2) = O(N(\log_2(N))^{d-1} \cdot p_{\max}^2)$ operations for the product of the stiffness matrix S with a given solution $\underline{u}$. However, if on each level l degree $p := l+1$ is applied with no limit for p, the result is a total of $O(M \cdot (\log_2(N))^2) = O(N(\log_2(N))^{d+1})$ operations.

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

5.2 A Numerical Example

As a simple example and model problem, we study the Laplace equation in two dimensions on the unit square **Q** with the Dirichlet boundary conditions given in (13). To solve the resulting linear system, a multigrid algorithm based on a damped Jacobi smoothing on $\tilde{G}_n^{(2)}$ and successive coarse grid correction steps on the regular full grids with mesh widths $h_x = 2^{-i_1}$ and $h_y = 2^{-i_2}$ for $i_1 + i_2 = n+1$ and for $i_1 = i_2 \leq n/2$ has been used. Figure 20 shows the behaviour of the L_∞-error $\|\tilde{e}_n^{(2)}\|_\infty$ on the regular sparse grids $\tilde{G}_n^{(2)}$ and the factors ρ_n of error reduction, $\rho_n := \|\tilde{e}_{n+1}^{(2)}\|_\infty / \|\tilde{e}_n^{(2)}\|_\infty$, for increasing n and $2 \leq p \leq 6$. For degree p, the approximation quality turns out to be a little bit worse than the order $O(h^{p+1}) = O(2^{-n(p+1)})$ of the full grid case, which is due to the additional logarithmic factor typical for sparse grids. Nevertheless, the higher order approximation properties can be seen clearly.

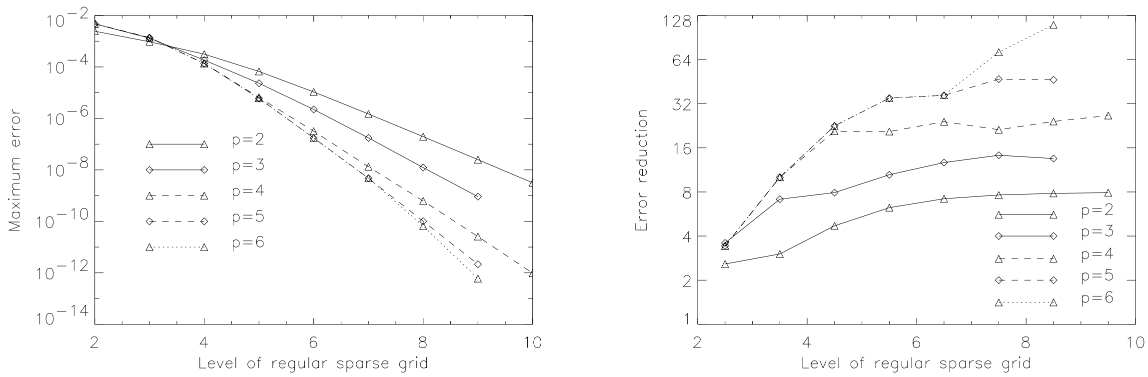


Figure 20: L_∞-error on $\tilde{G}_n^{(2)}$ (left) and factors ρ_n of error reduction (right) for various n and p

Finally, Fig. 21 shows the spectral condition number of the diagonally preconditioned stiffness matrix for different regular sparse grids $\tilde{G}_n^{(2)}$ and $2 \leq p \leq 6$. In comparison with the behaviour known from hierarchical polynomials in a p- or h-p-version context (cf. [31], e. g.), the influence of the polynomial degree on the condition number turns out to be quite moderate. This was to be expected, because, due to the fact that we only use a small part of the respective hierarchical interpolants as our actual basis function, the shape of the basis functions does not differ that much (cf. Fig. 19, e. g.).

6 Concluding Remarks

In this paper, we presented unidirectional algorithms for two important issues in the context of sparse grid finite element discretizations of elliptic partial differential equations. Concerning the class of equations that can be tackled, we extended the discretization to the general linear elliptic differential operator of second order on the d-dimensional unit interval. This extension enables us to deal with more general geometries using domain transformation techniques. Furthermore, we presented two grid generation methods, the transfinite interpolation as the intrinsic method for sparse grids and a hybrid method which produces smooth grids. Concerning the underlying hierarchical basis, we presented a strategy how to define d-dimensional hierarchical tensor product bases of piecewise arbitrary polynomial

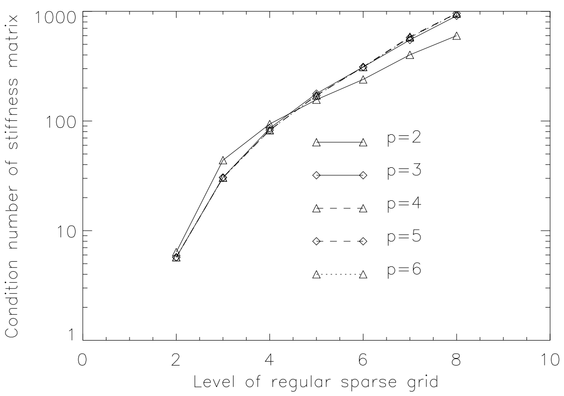


Figure 21: Condition of the (diagonally preconditioned) stiffness matrix for sparse grids $G_n^{(2)}$

degree. This approach opens the way to some kind of p-adaptivity, if suitable criteria for the adaptive handling of p are developed. Since the construction, handling, and storage of our polynomial bases are very simple and cheap, our approach provides a very promising access to finite element methods of higher order on sparse grids. The focus of the future work will be to bring together the different algorithmic concepts that have been developed so far in the sparse grid area in order to get an efficient solver with a broad area of application.

Acknowledgements. We are indebted to Christoph Zenger and to Stefan Zimmer for many fruitful discussions and valuable suggestions.

References

[1] A. Amsden and C. Hirt, *A simple scheme for generating general curvilinear grids*, Journal of Computational Physics, 11 (1973), pp. 348–359.

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

[3] I. Babuška, B. A. Szabó, and I. N. Katz, *The p-version of the finite element method*, SIAM J. Numer. Anal., 18 (1981), pp. 515–545.

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

[5] R. E. Bank, T. Dupont, and H. Yserentant, *The hierarchical basis multigrid method*, Numerische Mathematik, 52 (1988), pp. 427–458.

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

[7] H.-J. Bungartz, *An adaptive Poisson solver using hierarchical bases and sparse grids*, in Iterative Methods in Linear Algebra: Proceedings of the IMACS International Symposium, Brussels, 2.-4. 4. 1991, P. de Groen and R. Beauwens, eds., Elsevier, Amsterdam, 1992, pp. 293–310.

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

[9] ——, *Concepts for higher order finite elements on sparse grids*, in Houston Journal of Mathematics: Proceedings of the 3^{rd} Int. Conf. on Spectral and High Order Methods, 5.–9. 6. 1995, Houston, A. V. Ilin and L. R. Scott, eds., 1996, pp. 159–170.

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

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

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

[13] W. GORDON AND C. HALL, *Construction of curvilinear coordinate systems and application to mesh generation*, Int. J. Numer. Methods Eng., 7 (1973), pp. 461–477.

[14] W. GORDON AND L. THIEL, *Transfinite mapping and their application to grid generation*, in Numerical Grid Generation, J. Thompson, ed., Amsterdam, 1982, North–Holland, pp. 171–192.

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

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

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

[18] M. GRIEBEL, M. SCHNEIDER, AND C. ZENGER, *A combination technique for the solution of sparse grid problems*, in Iterative Methods in Linear Algebra: Proceedings of the IMACS International Symposium, Brussels, 2.-4. 4. 1991, P. de Groen and R. Beauwens, eds., Elsevier, Amsterdam, 1992, pp. 263–281.

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

[20] ——, *The h-p-version of the finite element method (Part 2: General results and applications)*, Computational Mechanics, 1 (1986), pp. 203–220.

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

[22] P. W. HEMKER, *Sparse grid finite-volume multigrid for 3D problems*, Advances in Computational Mathematics, Special issue on multiscale techniques (1994), pp. 215–232.

[23] S. HILGENFELDT, *Numerische Lösung der stationären Schrödingergleichung mit Finite-Element-Methoden auf dünnen Gittern*, Diplomarbeit, Technische Universität München, 1994.

[24] P. KNUPP AND S. STEINBERG, *Fundamentals of Grid Generation*, CRC Press, Inc., London, 1993.

[25] G. LIAO, *On harmonic maps*, in Mathematical Aspects of Numerical Grid Generation, J. Castillo, ed., Philadelphia, 1991, SIAM.

[26] C. PFLAUM, *Diskretisierung elliptischer Differentialgleichungen mit d"unnen Gittern*, phd thesis, Technische Universität München, Februar 1996.

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

[28] J. THOMPSON, F. THAMES, AND C. MASTIN, *Automatic numerical generation of boundary-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies*, Journal of Computational Physics, 15 (1974), pp. 299–319.

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

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

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

SFB 342: Methoden und Werkzeuge für die Nutz ung paralleler Rechnerarchitekturen

bisher erschienen :

Reihe A

342/1/90 A	Robert Gold, Walter Vogler: Quality Criteria for Partial Order Semantics of Place/Transition-Nets, Januar 1990
342/2/90 A	Reinhard Fößmeier: Die Rolle der Lastverteilung bei der numerischen Parallelprogrammierung, Februar 1990
342/3/90 A	Klaus-Jörn Lange, Peter Rossmanith: Two Results on Unambiguous Circuits, Februar 1990
342/4/90 A	Michael Griebel: Zur Lösung von Finite-Differenzen- und Finite-Element-Gleichungen mittels der Hierarchischen Transformations-Mehrgitter-Methode
342/5/90 A	Reinhold Letz, Johann Schumann, Stephan Bayerl, Wolfgang Bibel: SETHEO: A High-Performance Theorem Prover
342/6/90 A	Johann Schumann, Reinhold Letz: PARTHEO: A High Performance Parallel Theorem Prover
342/7/90 A	Johann Schumann, Norbert Trapp, Martin van der Koelen: SETHEO/PARTHEO Users Manual
342/8/90 A	Christian Suttner, Wolfgang Ertel: Using Connectionist Networks for Guiding the Search of a Theorem Prover
342/9/90 A	Hans-Jörg Beier, Thomas Bemmerl, Arndt Bode, Hubert Ertl, Olav Hansen, Josef Haunerdinger, Paul Hofstetter, Jaroslav Kremenek, Robert Lindhof, Thomas Ludwig, Peter Luksch, Thomas Treml: TOPSYS, Tools for Parallel Systems (Artikelsammlung)
342/10/90 A	Walter Vogler: Bisimulation and Action Refinement
342/11/90 A	Jörg Desel, Javier Esparza: Reachability in Reversible Free- Choice Systems
342/12/90 A	Rob van Glabbeek, Ursula Goltz: Equivalences and Refinement
342/13/90 A	Rob van Glabbeek: The Linear Time - Branching Time Spectrum
342/14/90 A	Johannes Bauer, Thomas Bemmerl, Thomas Treml: Leistungsanalyse von verteilten Beobachtungs- und Bewertungswerkzeugen
342/15/90 A	Peter Rossmanith: The Owner Concept for PRAMs
342/16/90 A	G. Böckle, S. Trosch: A Simulator for VLIW-Architectures
342/17/90 A	P. Slavkovsky, U. Rüde: Schnellere Berechnung klassischer Matrix-Multiplikationen

Reihe A

342/18/90 A	Christoph Zenger: SPARSE GRIDS
342/19/90 A	Michael Griebel, Michael Schneider, Christoph Zenger: A combination technique for the solution of sparse grid problems
342/20/90 A	Michael Griebel: A Parallelizable and Vectorizable Multi- Level-Algorithm on Sparse Grids
342/21/90 A	V. Diekert, E. Ochmanski, K. Reinhardt: On confluent semi-commutations-decidability and complexity results
342/22/90 A	Manfred Broy, Claus Dendorfer: Functional Modelling of Operating System Structures by Timed Higher Order Stream Processing Functions
342/23/90 A	Rob van Glabbeek, Ursula Goltz: A Deadlock-sensitive Congruence for Action Refinement
342/24/90 A	Manfred Broy: On the Design and Verification of a Simple Distributed Spanning Tree Algorithm
342/25/90 A	Thomas Bemmerl, Arndt Bode, Peter Braun, Olav Hansen, Peter Luksch, Roland Wismüller: TOPSYS - Tools for Parallel Systems (User's Overview and User's Manuals)
342/26/90 A	Thomas Bemmerl, Arndt Bode, Thomas Ludwig, Stefan Tritscher: MMK - Multiprocessor Multitasking Kernel (User's Guide and User's Reference Manual)
342/27/90 A	Wolfgang Ertel: Random Competition: A Simple, but Efficient Method for Parallelizing Inference Systems
342/28/90 A	Rob van Glabbeek, Frits Vaandrager: Modular Specification of Process Algebras
342/29/90 A	Rob van Glabbeek, Peter Weijland: Branching Time and Abstraction in Bisimulation Semantics
342/30/90 A	Michael Griebel: Parallel Multigrid Methods on Sparse Grids
342/31/90 A	Rolf Niedermeier, Peter Rossmanith: Unambiguous Simulations of Auxiliary Pushdown Automata and Circuits
342/32/90 A	Inga Niepel, Peter Rossmanith: Uniform Circuits and Exclusive Read PRAMs
342/33/90 A	Dr. Hermann Hellwagner: A Survey of Virtually Shared Memory Schemes
342/1/91 A	Walter Vogler: Is Partial Order Semantics Necessary for Action Refinement?
342/2/91 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Rainer Weber: Characterizing the Behaviour of Reactive Systems by Trace Sets
342/3/91 A	Ulrich Furbach, Christian Suttner, Bertram Fronhöfer: Massively Parallel Inference Systems
342/4/91 A	Rudolf Bayer: Non-deterministic Computing, Transactions and Recursive Atomicity

Reihe A

342/5/91 A	Robert Gold: Dataflow semantics for Petri nets
342/6/91 A	A. Heise; C. Dimitrovici: Transformation und Komposition von P/T-Netzen unter Erhaltung wesentlicher Eigenschaften
342/7/91 A	Walter Vogler: Asynchronous Communication of Petri Nets and the Refinement of Transitions
342/8/91 A	Walter Vogler: Generalized OM-Bisimulation
342/9/91 A	Christoph Zenger, Klaus Hallatschek: Fouriertransformation auf dünnen Gittern mit hierarchischen Basen
342/10/91 A	Erwin Loibl, Hans Obermaier, Markus Pawlowski: Towards Parallelism in a Relational Database System
342/11/91 A	Michael Werner: Implementierung von Algorithmen zur Kompaktifizierung von Programmen für VLIW-Architekturen
342/12/91 A	Reiner Müller: Implementierung von Algorithmen zur Optimierung von Schleifen mit Hilfe von Software-Pipelining Techniken
342/13/91 A	Sally Baker, Hans-Jörg Beier, Thomas Bemmerl, Arndt Bode, Hubert Ertl, Udo Graf, Olav Hansen, Josef Haunerdinger, Paul Hofstetter, Rainer Knödlseder, Jaroslav Kremenek, Siegfried Langenbuch, Robert Lindhof, Thomas Ludwig, Peter Luksch, Roy Milner, Bernhard Ries, Thomas Treml: TOPSYS - Tools for Parallel Systems (Artikelsammlung); 2., erweiterte Auflage
342/14/91 A	Michael Griebel: The combination technique for the sparse grid solution of PDE's on multiprocessor machines
342/15/91 A	Thomas F. Gritzner, Manfred Broy: A Link Between Process Algebras and Abstract Relation Algebras?
342/16/91 A	Thomas Bemmerl, Arndt Bode, Peter Braun, Olav Hansen, Thomas Treml, Roland Wismüller: The Design and Implementation of TOPSYS
342/17/91 A	Ulrich Furbach: Answers for disjunctive logic programs
342/18/91 A	Ulrich Furbach: Splitting as a source of parallelism in disjunctive logic programs
342/19/91 A	Gerhard W. Zumbusch: Adaptive parallele Multilevel-Methoden zur Lösung elliptischer Randwertprobleme
342/20/91 A	M. Jobmann, J. Schumann: Modelling and Performance Analysis of a Parallel Theorem Prover
342/21/91 A	Hans-Joachim Bungartz: An Adaptive Poisson Solver Using Hierarchical Bases and Sparse Grids
342/22/91 A	Wolfgang Ertel, Theodor Gemenis, Johann M. Ph. Schumann, Christian B. Suttner, Rainer Weber, Zongyan Qiu: Formalisms and Languages for Specifying Parallel Inference Systems
342/23/91 A	Astrid Kiehn: Local and Global Causes
342/24/91 A	Johann M.Ph. Schumann: Parallelization of Inference Systems by using an Abstract Machine

Reihe A

342/25/91 A	Eike Jessen: Speedup Analysis by Hierarchical Load Decomposition
342/26/91 A	Thomas F. Gritzner: A Simple Toy Example of a Distributed System: On the Design of a Connecting Switch
342/27/91 A	Thomas Schnekenburger, Andreas Weininger, Michael Friedrich: Introduction to the Parallel and Distributed Programming Language ParMod-C
342/28/91 A	Claus Dendorfer: Funktionale Modellierung eines Postsystems
342/29/91 A	Michael Griebel: Multilevel algorithms considered as iterative methods on indefinite systems
342/30/91 A	W. Reisig: Parallel Composition of Liveness
342/31/91 A	Thomas Bemmerl, Christian Kasperbauer, Martin Mairandres, Bernhard Ries: Programming Tools for Distributed Multiprocessor Computing Environments
342/32/91 A	Frank Leßke: On constructive specifications of abstract data types using temporal logic
342/1/92 A	L. Kanal, C.B. Suttner (Editors): Informal Proceedings of the Workshop on Parallel Processing for AI
342/2/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: The Design of Distributed Systems - An Introduction to FOCUS
342/2-2/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: The Design of Distributed Systems - An Introduction to FOCUS - Revised Version (erschienen im Januar 1993)
342/3/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: Summary of Case Studies in FOCUS - a Design Method for Distributed Systems
342/4/92 A	Claus Dendorfer, Rainer Weber: Development and Implementation of a Communication Protocol - An Exercise in FOCUS
342/5/92 A	Michael Friedrich: Sprachmittel und Werkzeuge zur Unterstüt- zung paralleler und verteilter Programmierung
342/6/92 A	Thomas F. Gritzner: The Action Graph Model as a Link between Abstract Relation Algebras and Process-Algebraic Specifications
342/7/92 A	Sergei Gorlatch: Parallel Program Development for a Recursive Numerical Algorithm: a Case Study
342/8/92 A	Henning Spruth, Georg Sigl, Frank Johannes: Parallel Algorithms for Slicing Based Final Placement
342/9/92 A	Herbert Bauer, Christian Sporrer, Thomas Krodel: On Distributed Logic Simulation Using Time Warp
342/10/92 A	H. Bungartz, M. Griebel, U. Rüde: Extrapolation, Combination and Sparse Grid Techniques for Elliptic Boundary Value Problems

Reihe A

342/11/92 A M. Griebel, W. Huber, U. Rüde, T. Störtkuhl: The Combination Technique for Parallel Sparse-Grid-Preconditioning and -Solution of PDEs on Multiprocessor Machines and Workstation Networks

342/12/92 A Rolf Niedermeier, Peter Rossmanith: Optimal Parallel Algorithms for Computing Recursively Defined Functions

342/13/92 A Rainer Weber: Eine Methodik für die formale Anforderungsspezifkation verteilter Systeme

342/14/92 A Michael Griebel: Grid– and point–oriented multilevel algorithms

342/15/92 A M. Griebel, C. Zenger, S. Zimmer: Improved multilevel algorithms for full and sparse grid problems

342/16/92 A J. Desel, D. Gomm, E. Kindler, B. Paech, R. Walter: Bausteine eines kompositionalen Beweiskalküls für netzmodellierte Systeme

342/17/92 A Frank Dederichs: Transformation verteilter Systeme: Von applikativen zu prozeduralen Darstellungen

342/18/92 A Andreas Listl, Markus Pawlowski: Parallel Cache Management of a RDBMS

342/19/92 A Erwin Loibl, Markus Pawlowski, Christian Roth: PART: A Parallel Relational Toolbox as Basis for the Optimization and Interpretation of Parallel Queries

342/20/92 A Jörg Desel, Wolfgang Reisig: The Synthesis Problem of Petri Nets

342/21/92 A Robert Balder, Christoph Zenger: The d-dimensional Helmholtz equation on sparse Grids

342/22/92 A Ilko Michler: Neuronale Netzwerk-Paradigmen zum Erlernen von Heuristiken

342/23/92 A Wolfgang Reisig: Elements of a Temporal Logic. Coping with Concurrency

342/24/92 A T. Störtkuhl, Chr. Zenger, S. Zimmer: An asymptotic solution for the singularity at the angular point of the lid driven cavity

342/25/92 A Ekkart Kindler: Invariants, Compositionality and Substitution

342/26/92 A Thomas Bonk, Ulrich Rüde: Performance Analysis and Optimization of Numerically Intensive Programs

342/1/93 A M. Griebel, V. Thurner: The Efficient Solution of Fluid Dynamics Problems by the Combination Technique

342/2/93 A Ketil Stølen, Frank Dederichs, Rainer Weber: Assumption / Commitment Rules for Networks of Asynchronously Communicating Agents

342/3/93 A Thomas Schnekenburger: A Definition of Efficiency of Parallel Programs in Multi-Tasking Environments

342/4/93 A Hans-Joachim Bungartz, Michael Griebel, Dierk Röschke, Christoph Zenger: A Proof of Convergence for the Combination Technique for the Laplace Equation Using Tools of Symbolic Computation

Reihe A

342/5/93 A	Manfred Kunde, Rolf Niedermeier, Peter Rossmanith: Faster Sorting and Routing on Grids with Diagonals
342/6/93 A	Michael Griebel, Peter Oswald: Remarks on the Abstract Theory of Additive and Multiplicative Schwarz Algorithms
342/7/93 A	Christian Sporrer, Herbert Bauer: Corolla Partitioning for Distributed Logic Simulation of VLSI Circuits
342/8/93 A	Herbert Bauer, Christian Sporrer: Reducing Rollback Overhead in Time-Warp Based Distributed Simulation with Optimized Incremental State Saving
342/9/93 A	Peter Slavkovsky: The Visibility Problem for Single-Valued Surface (z = f(x,y)): The Analysis and the Parallelization of Algorithms
342/10/93 A	Ulrich Rüde: Multilevel, Extrapolation, and Sparse Grid Methods
342/11/93 A	Hans Regler, Ulrich Rüde: Layout Optimization with Algebraic Multigrid Methods
342/12/93 A	Dieter Barnard, Angelika Mader: Model Checking for the Modal Mu-Calculus using Gauß Elimination
342/13/93 A	Christoph Pflaum, Ulrich Rüde: Gauß' Adaptive Relaxation for the Multilevel Solution of Partial Differential Equations on Sparse Grids
342/14/93 A	Christoph Pflaum: Convergence of the Combination Technique for the Finite Element Solution of Poisson's Equation
342/15/93 A	Michael Luby, Wolfgang Ertel: Optimal Parallelization of Las Vegas Algorithms
342/16/93 A	Hans-Joachim Bungartz, Michael Griebel, Dierk Röschke, Christoph Zenger: Pointwise Convergence of the Combination Technique for Laplace's Equation
342/17/93 A	Georg Stellner, Matthias Schumann, Stefan Lamberts, Thomas Ludwig, Arndt Bode, Martin Kiehl und Rainer Mehlhorn: Developing Multicomputer Applications on Networks of Workstations Using NXLib
342/18/93 A	Max Fuchs, Ketil Stølen: Development of a Distributed Min/Max Component
342/19/93 A	Johann K. Obermaier: Recovery and Transaction Management in Write-optimized Database Systems
342/20/93 A	Sergej Gorlatch: Deriving Efficient Parallel Programs by Systemating Coarsing Specification Parallelism
342/01/94 A	Reiner Hüttl, Michael Schneider: Parallel Adaptive Numerical Simulation
342/02/94 A	Henning Spruth, Frank Johannes: Parallel Routing of VLSI Circuits Based on Net Independency
342/03/94 A	Henning Spruth, Frank Johannes, Kurt Antreich: PHIroute: A Parallel Hierarchical Sea-of-Gates Router

Reihe A

342/04/94 A Martin Kiehl, Rainer Mehlhorn, Matthias Schumann: Parallel Multiple Shooting for Optimal Control Problems Under NX/2
342/05/94 A Christian Suttner, Christoph Goller, Peter Krauss, Klaus-Jörn Lange, Ludwig Thomas, Thomas Schnekenburger: Heuristic Optimization of Parallel Computations
342/06/94 A Andreas Listl: Using Subpages for Cache Coherency Control in Parallel Database Systems
342/07/94 A Manfred Broy, Ketil Stølen: Specification and Refinement of Finite Dataflow Networks - a Relational Approach
342/08/94 A Katharina Spies: Funktionale Spezifikation eines Kommunikationsprotokolls
342/09/94 A Peter A. Krauss: Applying a New Search Space Partitioning Method to Parallel Test Generation for Sequential Circuits
342/10/94 A Manfred Broy: A Functional Rephrasing of the Assumption/Commitment Specification Style
342/11/94 A Eckhardt Holz, Ketil Stølen: An Attempt to Embed a Restricted Version of SDL as a Target Language in Focus
342/12/94 A Christoph Pflaum: A Multi-Level-Algorithm for the Finite-Element-Solution of General Second Order Elliptic Differential Equations on Adaptive Sparse Grids
342/13/94 A Manfred Broy, Max Fuchs, Thomas F. Gritzner, Bernhard Schätz, Katharina Spies, Ketil Stølen: Summary of Case Studies in FOCUS - a Design Method for Distributed Systems
342/14/94 A Maximilian Fuchs: Technologieabhängigkeit von Spezifikationen digitaler Hardware
342/15/94 A M. Griebel, P. Oswald: Tensor Product Type Subspace Splittings And Multilevel Iterative Methods For Anisotropic Problems
342/16/94 A Gheorghe Ştefănescu: Algebra of Flownomials
342/17/94 A Ketil Stølen: A Refinement Relation Supporting the Transition from Unbounded to Bounded Communication Buffers
342/18/94 A Michael Griebel, Tilman Neuhoeffer: A Domain-Oriented Multilevel Algorithm-Implementation and Parallelization
342/19/94 A Michael Griebel, Walter Huber: Turbulence Simulation on Sparse Grids Using the Combination Method
342/20/94 A Johann Schumann: Using the Theorem Prover SETHEO for verifying the development of a Communication Protocol in FOCUS - A Case Study -
342/01/95 A Hans-Joachim Bungartz: Higher Order Finite Elements on Sparse Grids
342/02/95 A Tao Zhang, Seonglim Kang, Lester R. Lipsky: The Performance of Parallel Computers: Order Statistics and Amdahl's Law

Reihe A

342/03/95 A Lester R. Lipsky, Appie van de Liefvoort: Transformation of the Kronecker Product of Identical Servers to a Reduced Product Space

342/04/95 A Pierre Fiorini, Lester R. Lipsky, Wen-Jung Hsin, Appie van de Liefvoort: Auto-Correlation of Lag-k For Customers Departing From Semi-Markov Processes

342/05/95 A Sascha Hilgenfeldt, Robert Balder, Christoph Zenger: Sparse Grids: Applications to Multi-dimensional Schrödinger Problems

342/06/95 A Maximilian Fuchs: Formal Design of a Model-N Counter

342/07/95 A Hans-Joachim Bungartz, Stefan Schulte: Coupled Problems in Microsystem Technology

342/08/95 A Alexander Pfaffinger: Parallel Communication on Workstation Networks with Complex Topologies

342/09/95 A Ketil Stølen: Assumption/Commitment Rules for Data-flow Networks - with an Emphasis on Completeness

342/10/95 A Ketil Stølen, Max Fuchs: A Formal Method for Hardware/Software Co-Design

342/11/95 A Thomas Schnekenburger: The ALDY Load Distribution System

342/12/95 A Javier Esparza, Stefan Römer, Walter Vogler: An Improvement of McMillan's Unfolding Algorithm

342/13/95 A Stephan Melzer, Javier Esparza: Checking System Properties via Integer Programming

342/14/95 A Radu Grosu, Ketil Stølen: A Denotational Model for Mobile Point-to-Point Dataflow Networks

342/15/95 A Andrei Kovalyov, Javier Esparza: A Polynomial Algorithm to Compute the Concurrency Relation of Free-Choice Signal Transition Graphs

342/16/95 A Bernhard Schätz, Katharina Spies: Formale Syntax zur logischen Kernsprache der Focus-Entwicklungsmethodik

342/17/95 A Georg Stellner: Using CoCheck on a Network of Workstations

342/18/95 A Arndt Bode, Thomas Ludwig, Vaidy Sunderam, Roland Wismüller: Workshop on PVM, MPI, Tools and Applications

342/19/95 A Thomas Schnekenburger: Integration of Load Distribution into ParMod-C

342/20/95 A Ketil Stølen: Refinement Principles Supporting the Transition from Asynchronous to Synchronous Communication

342/21/95 A Andreas Listl, Giannis Bozas: Performance Gains Using Subpages for Cache Coherency Control

342/22/95 A Volker Heun, Ernst W. Mayr: Embedding Graphs with Bounded Treewidth into Optimal Hypercubes

342/23/95 A Petr Jančar, Javier Esparza: Deciding Finiteness of Petri Nets up to Bisimulation

Reihe A

342/24/95 A	M. Jung, U. Rüde: Implicit Extrapolation Methods for Variable Coefficient Problems
342/01/96 A	Michael Griebel, Tilman Neunhoeffer, Hans Regler: Algebraic Multigrid Methods for the Solution of the Navier-Stokes Equations in Complicated Geometries
342/02/96 A	Thomas Grauschopf, Michael Griebel, Hans Regler: Additive Multilevel-Preconditioners based on Bilinear Interpolation, Matrix Dependent Geometric Coarsening and Algebraic-Multigrid Coarsening for Second Order Elliptic PDEs
342/03/96 A	Volker Heun, Ernst W. Mayr: Optimal Dynamic Edge-Disjoint Embeddings of Complete Binary Trees into Hypercubes
342/04/96 A	Thomas Huckle: Efficient Computation of Sparse Approximate Inverses
342/05/96 A	Thomas Ludwig, Roland Wismüller, Vaidy Sunderam, Arndt Bode: OMIS — On-line Monitoring Interface Specification
342/06/96 A	Ekkart Kindler: A Compositional Partial Order Semantics for Petri Net Components
342/07/96 A	Richard Mayr: Some Results on Basic Parallel Processes
342/08/96 A	Ralph Radermacher, Frank Weimer: INSEL Syntax-Bericht
342/09/96 A	P.P. Spies, C. Eckert, M. Lange, D. Marek, R. Radermacher, F. Weimer, H.-M. Windisch: Sprachkonzepte zur Konstruktion verteilter Systeme
342/10/96 A	Stefan Lamberts, Thomas Ludwig, Christian Röder, Arndt Bode: PFS-Lib – A File System for Parallel Programming Environments
342/11/96 A	Manfred Broy, Gheorghe Ştefănescu: The Algebra of Stream Processing Functions
342/12/96 A	Javier Esparza: Reachability in Live and Safe Free-Choice Petri Nets is NP-complete
342/13/96 A	Radu Grosu, Ketil Stølen: A Denotational Model for Mobile Many-to-Many Data-flow Networks
342/14/96 A	Giannis Bozas, Michael Jaedicke, Andreas Listl, Bernhard Mitschang, Angelika Reiser, Stephan Zimmermann: On Transforming a Sequential SQL-DBMS into a Parallel One: First Results and Experiences of the MIDAS Project
342/15/96 A	Richard Mayr: A Tableau System for Model Checking Petri Nets with a Fragment of the Linear Time μ -Calculus
342/16/96 A	Ursula Hinkel, Katharina Spies: Anleitung zur Spezifikation von mobilen, dynamischen Focus-Netzen
342/17/96 A	Richard Mayr: Model Checking PA-Processes
342/18/96 A	Michaela Huhn: Put your Model Checker on Diet: Verification on Local States

Reihe A

SFB 342 : Methoden und Werkzeuge für die Nutz ung paralleler Rechnerarchitekturen

Reihe B

342/1/90 B	Wolfgang Reisig: Petri Nets and Algebraic Specifications
342/2/90 B	Jörg Desel: On Abstraction of Nets
342/3/90 B	Jörg Desel: Reduction and Design of Well-behaved Free-choice Systems
342/4/90 B	Franz Abstreiter, Michael Friedrich, Hans-Jürgen Plewan: Das Werkzeug runtime zur Beobachtung verteilter und paralleler Programme
342/1/91 B	Barbara Paech1: Concurrency as a Modality
342/2/91 B	Birgit Kandler, Markus Pawlowski: SAM: Eine Sortier- Toolbox - Anwenderbeschreibung
342/3/91 B	Erwin Loibl, Hans Obermaier, Markus Pawlowski: 2. Workshop über Parallelisierung von Datenbanksystemen
342/4/91 B	Werner Pohlmann: A Limitation of Distributed Simulation Methods
342/5/91 B	Dominik Gomm, Ekkart Kindler: A Weakly Coherent Virtually Shared Memory Scheme: Formal Specification and Analysis
342/6/91 B	Dominik Gomm, Ekkart Kindler: Causality Based Specification and Correctness Proof of a Virtually Shared Memory Scheme
342/7/91 B	W. Reisig: Concurrent Temporal Logic
342/1/92 B	Malte Grosse, Christian B. Suttner: A Parallel Algorithm for Set-of-Support Christian B. Suttner: Parallel Computation of Multiple Sets-of-Support
342/2/92 B	Arndt Bode, Hartmut Wedekind: Parallelrechner: Theorie, Hardware, Software, Anwendungen
342/1/93 B	Max Fuchs: Funktionale Spezifikation einer Geschwindigkeitsregelung
342/2/93 B	Ekkart Kindler: Sicherheits- und Lebendigkeitseigenschaften: Ein Literaturüberblick
342/1/94 B	Andreas Listl; Thomas Schnekenburger; Michael Friedrich: Zum Entwurf eines Prototypen für MIDAS