

Adaptive Wavelet Schemes for Elliptic Problems – Implementation and Numerical Experiments*

Arne Barinka Titus Barsch Philippe Charton Albert Cohen
Stephan Dahlke Wolfgang Dahmen Karsten Urban

Abstract

Recently an adaptive wavelet scheme could be proved to be asymptotically optimal for a wide class of elliptic operator equations in the sense that the error achieved by an adaptive approximate solution stays proportional to the smallest possible error that can be realized by *any* linear combination of the corresponding number of wavelets. On one hand, the results are purely asymptotic. On the other hand, the analysis suggests new algorithmic ingredients for which no prototypes seem to exist yet. It is therefore the objective of this paper to develop suitable data structures for the new algorithmic components and to obtain a quantitative validation of the theoretical results. We briefly review first the main theoretical facts, give a detailed description of the algorithm, highlight the essential data structures and illustrate the results by one and two dimensional numerical examples.

Key Words: Elliptic operator equations, multiscale methods, adaptive methods, wavelets, quasi sparse matrices and vectors, fast matrix vector multiplication, best N -term approximation, thresholding, Besov spaces, C++, STL.

AMS subject classification: 35B65, 41A25, 41A46, 42C15 46E35, 65F99, 65N12, 65N55

1 Introduction

The development of adaptive numerical methods is of enormous current interest. Although such concepts have not entered yet industrial applications at large, current research developments for instance in a finite element context indicate their very promising potential

*This work has been started during the European Summerschool CEMRACS '98 that took place in Marseille. The authors have been supported by the following funds: Volkswagen-Stiftung in the priority project *Modellierung komplexer Systeme in der Verfahrenstechnik*, Deutsche Forschungsgemeinschaft (DFG), Grant Da 117/13-1, project *Adaptive Multiscale Methods*, European Commission in the TMR project *Wavelets and Multiscale Methods in Numerical Analysis and Simulation*, DAAD (Deutscher Akademischer Austauschdienst) within the Procope project.

[2, 3, 4, 5, 9, 10, 11, 12, 48, 49]. Such hopes and numerical experiences are, however, contrasted by negative statements proved in the context of complexity theory. In fact, on a rigorous level not much has been proved about the efficiency of adaptive finite element schemes in comparison with a-priorily fixed meshes. Only recently, in the context of wavelet discretizations it could be shown in [24] that a certain adaptive scheme converges for a wide class of elliptic operator equations without any a-priori assumptions on the unknown solution such as the saturation property. Aside from the guaranteed convergence it is interesting that the scheme works for differential operators as well as for singular integral operators. A comparable result in the finite element context concerns a much smaller scope of problems, namely bivariate piecewise linear finite element discretizations for Poisson's equation [39]. However, in either case nothing can be said about the actual *speed* of convergence so that conclusions on the efficiency compared with a-priorily fixed discretizations remain open. Here speed means to relate the number of degrees of freedom invoked by the adaptive scheme to the achieved accuracy of the solution. Substantial progress could be accomplished then in [18] in the following sense. There an adaptive wavelet scheme has been developed which is shown to be *asymptotically optimal*. This means that it produces the same rate of convergence as a *best N -term approximation* for the same class of elliptic operator equations referred to above. Moreover, the number of floating point operations required to compute the approximate solution stays proportional to the number N of wavelets needed to approximate the solution at that level of accuracy. The proof of the latter fact is constructive in the sense that the algorithm is described to the level of detail that the number of arithmetic operations can be rigorously estimated.

The result is interesting from two points of view. Since the rate of best N -term approximation can be characterized by Besov regularity [25] one can see that, in principle, such an adaptive scheme is asymptotically more efficient than uniform schemes exactly when the solution lacks Sobolev regularity relative to Besov regularity. On the other hand, the results are asymptotic while a more quantitative assessment of the performance is of equal interest in practical applications. Moreover, the analysis of the scheme suggests new algorithmic ingredients centering on an approximate fast matrix/vector multiplication combined with sorting entries of sequences. Therefore the efficient realization of these ingredients and the development of suitable data structures that support best the conceptual strength of the scheme in practical realizations is a challenging task. In fact, the realization of that task seems to be essential for a quantitative validation of the theoretical results which after all are phrased in a necessarily simplified computational model.

This report is to describe the developments of such algorithmic ingredients and corresponding data structures. It is organized as follows. In Section 2, we briefly review the main theoretical facts needed for the understanding of the algorithm. So far, we have assumed that the wavelets have certain properties, namely they are local, they induce *isomorphisms* between certain sequence and function spaces and they have certain *cancellation properties*. We are able to extract from theory the essential requirements on implementation. It is worth stressing that a fairly large part of data structure, namely everything concerned with sorting and organizing arrays, can be kept independently of the particular application. Section 3 is devoted to a brief outline of these structures. The main interface to a special application resides on a proper encoding of the wavelet index sets which implicitly also encodes the topology of the domain. This part of the implemen-

tation is tied to the type of application treated in Section 4. There we describe the type of boundary value problems in one and two spatial dimensions, that we are concerned with. In each case we briefly indicate which type of wavelets can be used to an extent needed to see the requirements on the encoding of indices. The examples are designed to bring out the effects of different sources of singularities whose occurrence, according to the theoretical part, makes adaptive schemes more efficient than nonadaptive ones. While in the 1D cases the singularity is induced by the right hand side data we consider in the 2D case a problem with smooth right hand side where the singularity in the solution comes from the shape of the domain. Moreover, we outline the problem dependent algorithmic ingredients.

2 Theoretical Background

2.1 The Problem

Suppose that H is a Hilbert space with norm $\|\cdot\|_H$ induced by the inner product $\langle \cdot, \cdot \rangle$ and that the selfadjoint operator $A : H \rightarrow H'$, where H' is the normed dual of H , is *H-elliptic*, i.e.,

$$a(v, w) := \langle Av, w \rangle \lesssim \|v\|_H \|w\|_H \quad \text{and} \quad a(v, v) \sim \|v\|_H^2. \quad (2.1)$$

Here $a \lesssim b$ means that a can be uniformly bounded by a constant multiple of b and vice versa independent of any parameters on which a and b may depend. $a \gtrsim b$ is to be understood in the analogous fashion and $a \sim b$ means that $a \lesssim b$ and $a \gtrsim b$. Clearly (2.1) means that A is an isomorphism from H to H' , i.e.,

$$\|Av\|_{H'} \sim \|v\|_H, \quad v \in H. \quad (2.2)$$

Thus the equation

$$Au = f \quad (2.3)$$

has for any $f \in H'$ a unique solution which will always be denoted by u . Typical examples are second order elliptic boundary value problems with Dirichlet boundary conditions on some open domain $\Omega \subset \mathbb{R}^d$. In this case $H = H_0^1(\Omega)$ and $H' = H^{-1}(\Omega)$. Other examples are obtained by turning an exterior boundary value problem into a singular integral equation on the boundary Γ of the domain. For a formulation in terms of the single layer potential operator one obtains for instance $H = H^{-1/2}(\Gamma)$ and $H' = H^{1/2}(\Gamma)$, see [20, 45] for details. Thus H is typically a Sobolev space and

$$H \subset L_2 \subset H' \quad \text{or} \quad H' \subset L_2 \subset H.$$

We sometimes write then $H = H^t$ to indicate the Sobolev regularity although often a closed subspace of the full Sobolev space determined by boundary conditions is meant. H^{-t} is always the dual of this particular subspace. One further property of A will matter unless A is a differential operator with regular coefficients. Whenever A has a global Schwartz kernel K , i.e.,

$$(Av)(x) = \int K(x, y)v(y)dy, \quad (2.4)$$

we will assume in addition that when (2.1) holds for $H = H^t$ then

$$\left| \partial_x^\alpha \partial_y^\beta K(x, y) \right| \lesssim \text{dist}(x, y)^{-(d+2t+|\alpha|+|\beta|)}. \quad (2.5)$$

We hasten to add though that A need not be a scalar equation but could as well represent a system in which case H is typically a product of Sobolev spaces.

We are interested in solving (2.3) approximately with the aid of a Galerkin method, i.e., we pick some finite dimensional space $S \subset H$ and search for $u_S \in S$ such that

$$\langle Au_S, v \rangle = \langle f, v \rangle, \quad v \in S, \quad (2.6)$$

where $\langle \cdot, \cdot \rangle$ denotes the standard L_2 -inner product.

2.2 Wavelet Bases and Isomorphisms

In our context the trial spaces S in (2.6) will be spanned by elements of a *wavelet* basis $\Psi = \{\psi_\lambda : \lambda \in \mathcal{J}\}$ for H . We will postpone at this point any technical description of the basis Ψ (which necessarily depends on the particular setting at hand) but will only list those properties that will be relevant in the following. Later in connection with concrete applications we will describe Ψ in more detail. The indices $\lambda \in \mathcal{J}$ typically encode several types of information, namely the *scale* often denoted by $|\lambda|$, the spatial location and also the type of the wavelet. Recall that in a classical setting a tensor product construction yields $2^d - 1$ types of wavelets [34, 46]. For instance, for wavelets on the real line λ can be identified with (j, k) , where $j = |\lambda|$ denotes the dyadic refinement level and $2^{-j}k$ signifies the location of the wavelet. In fact, we will require the wavelets to be local in the sense that

$$\text{diam}(\text{supp } \psi_\lambda) \sim 2^{-|\lambda|}, \quad \lambda \in \mathcal{J}. \quad (2.7)$$

What matters here is that any $v \in L_2$ has a *unique* expansion

$$v = \sum_{\lambda \in \mathcal{J}} d_\lambda \psi_\lambda =: \mathbf{d}^T \Psi$$

and that these expansions induce an *isomorphism* between H and ℓ_2 in the following sense: There exists a *diagonal matrix* $\mathbf{D} = \text{diag}(\omega_\lambda : \lambda \in \mathcal{J})$ such that

$$\|\mathbf{D}\mathbf{d}\|_{\ell_2(\mathcal{J})} \sim \|\mathbf{d}^T \Psi\|_H. \quad (2.8)$$

Denoting by $\tilde{\Psi}$ the *dual basis* to Ψ , i.e.,

$$\langle \psi_\lambda, \tilde{\psi}_\nu \rangle = \delta_{\lambda, \nu}, \quad \lambda, \nu \in \mathcal{J}, \quad (2.9)$$

(2.8) implies the dual relation

$$\|\mathbf{D}^{-1}\mathbf{d}\|_{\ell_2(\mathcal{J})} \sim \|\mathbf{d}^T \tilde{\Psi}\|_{H'}. \quad (2.10)$$

Similar relations are also known to hold for Sobolev spaces in L_p for $p \neq 2$. Moreover, interpolation between such spaces provides norm equivalences for a whole range of *Besov*

spaces $B_q^\alpha(L_p)$ [26, 38, 40, 46]. In the present context we will have to make use of the following special case

$$\|\mathbf{d}\|_{\ell_\tau(\mathcal{J})} \sim \|\mathbf{d}^T \Psi\|_{B_\tau^\alpha(L_\tau)}, \quad (2.11)$$

where the smoothness index α and the integrability index τ are related by

$$\frac{1}{\tau} = \frac{\alpha}{d} + \frac{1}{2}. \quad (2.12)$$

2.3 Cancellation Property

The second main requirement on the wavelet bases is that integration of a function against a wavelet annihilates the smooth part of the function, i.e.,

$$|\langle v, \psi_\lambda \rangle| \lesssim 2^{-|\lambda|(\tilde{m} + \frac{d}{2})} \|v\|_{W_\infty^{\tilde{m}}(\text{supp } \psi_\lambda)}, \quad (2.13)$$

where the positive integer \tilde{m} is related to the dual basis $\tilde{\Psi}$. In the classical case \tilde{m} is the order of *vanishing polynomial moments*, see [27]. Property (2.13) will ensure later that matrix representations of operators of the type (2.4) are almost sparse.

2.4 An Equivalent ℓ_2 -Problem

Once a basis with the above properties is given, it is natural to transform the operator equation (2.3) over a function space H into a *matrix* equation over the corresponding *sequence space*. The matrix in question is the representation of the operator with respect to the chosen wavelet basis. More precisely, the relation (2.8) suggests a special *scaling* of the basis which leads to

$$\mathbf{A} := \mathbf{D}^{-1} \langle \Psi, A\Psi \rangle \mathbf{D}^{-1} := \left(\omega_\lambda^{-1} \omega_\nu^{-1} \langle \psi_\lambda, A\psi_\nu \rangle \right)_{\lambda, \nu \in \mathcal{J}}. \quad (2.14)$$

Specifically, when $H = H^t$ an admissible choice for the diagonal weights is $\omega_\lambda = 2^{t|\lambda|}$. The crucial point is that the norm equivalence (2.8) in conjunction with ellipticity (2.2) implies that the (infinite) matrix \mathbf{A} defined by (2.14) is now an automorphism on ℓ_2 , [27, 29].

Theorem 2.1 *The function $u = \mathbf{d}^T \Psi \in H$ solves the original operator equation (2.3) if and only if the sequence*

$$\mathbf{u} := \mathbf{D}\mathbf{d} \quad (2.15)$$

solves the matrix equation

$$\mathbf{A}\mathbf{u} = \mathbf{f}, \quad (2.16)$$

where $\mathbf{f} := \mathbf{D}^{-1} \langle \Psi, f \rangle$.

Moreover, denoting by $\|\cdot\|$ the spectral norm on ℓ_2 , the matrix \mathbf{A} defined by (2.14) satisfies

$$\|\mathbf{A}\|, \|\mathbf{A}^{-1}\| < \infty. \quad (2.17)$$

As an immediate consequence there exists a finite number κ such that all finite sections

$$\mathbf{A}_\Lambda := \left(\omega_\lambda^{-1} \omega_\nu^{-1} \langle \psi_\lambda, A\psi_\nu \rangle \right)_{\lambda, \nu \in \Lambda}, \quad \Lambda \subset \mathcal{J},$$

have uniformly bounded condition numbers

$$\text{cond}_2(\mathbf{A}_\Lambda) \leq \kappa, \quad \Lambda \subset \mathcal{J}. \quad (2.18)$$

Hence the original problem has been reduced to an equivalent well-posed problem in ℓ_2 . This fact will be crucial for what follows.

Moreover, an economic treatment of this ℓ_2 -problem is partly due to the following consequence of the cancellation property (2.13). It can be shown that when $H = H^t$ suitable choices of Ψ entail the following decay of the entries of \mathbf{A} , [24, 29].

Theorem 2.2 *Suppose that for operators of the form (2.4) property (2.5) holds. Then for $H = H^t$ one has*

$$2^{-(|\lambda'|+|\lambda|)t} |\langle A\psi_{\lambda'}, \psi_\lambda \rangle| \lesssim \frac{2^{-\|\lambda|-|\lambda'\|\sigma}}{(1+d(\lambda, \lambda'))^{d+2\tilde{m}+2t}}, \quad (2.19)$$

where

$$d(\lambda, \lambda') := 2^{\min(|\lambda|, |\lambda'|)} \text{dist}(\Omega_\lambda, \Omega_{\lambda'}), \quad (2.20)$$

$\Omega_\lambda := \text{supp } \psi_\lambda$ and $\sigma > d/2$ depends on the regularity of the wavelets ψ_λ .

It is important to note, however, that (2.19) is only a *sufficient* condition for the following *compression property* of \mathbf{A} that will be needed later. The following fact has been proved in [18].

Proposition 2.3 *Let*

$$s^* := \min \left\{ \frac{\sigma}{d} - \frac{1}{2}, \frac{2t + 2\tilde{m}}{d} \right\}. \quad (2.21)$$

Then for every $s < s^$ there exists a positive summable sequence $(\alpha_j)_{j \geq 0}$ and for every $j \geq 0$ there exists a matrix \mathbf{A}_j with at most $2^j \alpha_j$ nonzero entries per row and column such that*

$$\|\mathbf{A}_j - \mathbf{A}\| \lesssim \alpha_j 2^{-sj}. \quad (2.22)$$

The class of matrices with the property (2.22) is called \mathcal{A}_s . For matrices with the particular decay properties (2.19) concrete truncation rules can be given [18]. The first step is a truncation in scale: Given j , set

$$\tilde{a}_{\lambda, \nu} := \begin{cases} a_{\lambda, \nu}, & \|\lambda\| - \|\nu\| \leq j/d, \\ 0, & \text{else,} \end{cases} \quad (2.23)$$

followed by a spatial truncation

$$a'_{\lambda, \nu} := \begin{cases} \tilde{a}_{\lambda, \nu}, & d(\lambda, \nu) \leq 2^{j/d - \|\lambda\| - \|\nu\|} \gamma(\|\lambda\| - \|\nu\|), \\ 0, & \text{else,} \end{cases} \quad (2.24)$$

Here $\gamma(n)$ is any summable sequence, e.g., $\gamma(n) := (1+n)^{-2/d}$.

One should note that because of (2.7) the first truncation (2.23) already suffices for local operators A .

2.5 The Basic Paradigm

The practical realization of adaptive approximations to (2.3) in a finite element context is to refine step by step a given mesh according to a posteriori local error indicators. The point of view taken by wavelet schemes is somewhat different. Trial spaces are refined directly by incorporating additional basis functions whose selection depends on the previous step. Specifically, setting for any finite subset $\Lambda \subset \mathcal{J}$

$$S_\Lambda := \text{span} \{ \psi_\lambda : \lambda \in \Lambda \},$$

and denoting by $u_\Lambda \in S_\Lambda$ always the Galerkin solution determined by (2.6), we start with some small index set Λ_0 (possibly the empty set) and proceed as follows:

Given Λ_j and u_{Λ_j} and some fixed $\theta \in (0, 1)$, find $\Lambda_{j+1} \supset \Lambda_j$ as small as possible such that the new error $u - u_{\Lambda_{j+1}}$ in the energy norm is at most θ times the previous error. Obviously, iteration of this step entails convergence of the resulting sequence of approximations in the energy norm.

Successively growing index sets in this way, one hopes to track the *most significant* coefficients in the true wavelet expansion $\mathbf{d}^T \Psi$ of the unknown solution u . The error at each step is naturally measured in the energy norm, see (2.1),

$$\|v\|^2 := a(v, v). \tag{2.25}$$

But then (2.8) suggests to work directly on the *discrete side*, noting that by Theorem 2.1 u_Λ solves (2.6) for $S = S_\Lambda$ if and only if \mathbf{u}_Λ solves

$$\mathbf{A}_\Lambda \mathbf{u}_\Lambda = \mathbf{f}_\Lambda := \mathbf{f}|_\Lambda, \tag{2.26}$$

and u_Λ is related to \mathbf{u}_Λ by

$$u_\Lambda = \sum_{\lambda \in \Lambda} \omega_\lambda^{-1} (\mathbf{u}_\Lambda)_\lambda \psi_\lambda. \tag{2.27}$$

Note that $\mathbf{u}_\Lambda \in \mathbb{R}^{\#\Lambda}$ is a finite vector. It will sometimes be convenient to view \mathbf{u}_Λ as a sequence in ℓ_2 , i.e., all components of \mathbf{u}_Λ outside Λ are understood to be zero. Since it will be always clear from the context which interpretation is meant we will not introduce any notational distinction between the finite vector \mathbf{u}_Λ and its canonical injection in ℓ_2 . Likewise for $\mathbf{v} \in \ell_2$ its restriction to Λ is denoted by $\mathbf{v}|_\Lambda$. Thus an equivalent formulation of (2.26) is

$$(\mathbf{A}\mathbf{u}_\Lambda - \mathbf{f})|_\Lambda = \mathbf{0}.$$

Defining now in analogy to (2.25) the *discrete energy norm*

$$\|\mathbf{v}\|^2 := \mathbf{v}^T \mathbf{A} \mathbf{v} =: \mathbf{a}(\mathbf{v}, \mathbf{v}), \tag{2.28}$$

(2.17) in Theorem 2.1 says that

$$\|\cdot\| \sim \|\cdot\|_{\ell_2}. \tag{2.29}$$

We can now detail the above adaptive paradigm as follows:

Given $\Lambda \subset \mathcal{J}$, find a possibly small index set $\hat{\Lambda} \supset \Lambda$ such that

$$\|\mathbf{u} - \mathbf{u}_{\hat{\Lambda}}\| \leq \theta \|\mathbf{u} - \mathbf{u}_{\Lambda}\|. \quad (2.30)$$

Of course, neither side of (2.30) can be evaluated. In order to still find a suitable expanded set $\hat{\Lambda}$ one tries to find $\hat{\Lambda}$ such that for some $\beta \in (0, 1)$

$$\|\mathbf{u}_{\hat{\Lambda}} - \mathbf{u}_{\Lambda}\| \geq \beta \|\mathbf{u} - \mathbf{u}_{\Lambda}\|, \quad (2.31)$$

i.e., $\hat{\Lambda}$ should be large enough to be sufficiently closer to the true solution. At least the left hand side is, in principle, computable.

Remark 2.4 *In fact, since \mathbf{u}_{Λ} is the orthogonal projection of \mathbf{u} onto $\mathbb{R}^{\#\Lambda}$, it immediately follows from Pythagoras' Theorem that (2.31) implies (2.30) with*

$$\theta := \sqrt{1 - \beta^2}. \quad (2.32)$$

The strategy to accomplish (2.31) has been used already earlier in the finite element context, see e.g. [12, 39]. It was also the starting point in [24]. Nevertheless, the problem remains to actually verify (2.31) since still the unknown sequence \mathbf{u} is involved. Therefore in most treatments it is *assumed* that for some *fixed* refinement of the old trial space a relation like (2.31) holds. This is usually referred to as the *saturation assumption*. However, in [24] and before for a much more specialized situation in [39] an adaptive refinement scheme was designed which guarantees (2.31) without an a-priori assumption like the saturation property.

In neither case though it was possible to derive a concrete *convergence rate*, i.e., to relate the error $\|\mathbf{u} - \mathbf{u}_{\Lambda_j}\|$ to the number of degrees of freedom $N_j = \#\Lambda_j$. But such a relation would eventually be needed for appraising the performance of an adaptive scheme in comparison with any schemes using preassigned discretizations.

2.6 Best N -Term Approximation

Before proceeding a few comments of conceptual nature are in order. To obtain a *benchmark* it is important to clarify first what the *optimal* outcome of an adaptive scheme might be. The answer is readily given by Theorem 2.1 and (2.29). Suppose for a moment that we have complete knowledge of u , i.e., we know *all* components in \mathbf{u} . Then we could choose N coefficients of \mathbf{u} so that the corresponding finite vector approximates \mathbf{u} best in the energy norm (2.28). The function u^N defined in analogy to (2.27) approximates, in view of (2.8), u , up to a uniform constant, best in the continuous energy norm or in H . \mathbf{u}^N is called a *best N -term approximation*. Clearly the selection of such best coefficients is a *nonlinear process*. Best N -term approximation is therefore a special instance of *nonlinear approximation*, see [36].

Even when knowing \mathbf{u} it is not clear what the most significant coefficients are that minimize the error in the energy norm among any possible selection of N terms. At this point the norm equivalence (2.8) in the form (2.29) comes into play. In fact, the best N -term approximation to \mathbf{u} with respect to the energy norm $\|\cdot\|$ produces an error

which is up to uniform constants equivalent to the best N -term approximation of \mathbf{u} in the Euclidean norm. Best N -term approximation in ℓ_2 , in turn, is well understood, a fact that will be heavily exploited later. Nevertheless, since \mathbf{u} is not known even the best N -term approximation of \mathbf{u} in ℓ_2 is not available.

In summary, the best that could be achieved by an adaptive scheme is to produce errors that stay proportional to best N -term approximation in ℓ_2 . Suppose for a moment that an adaptive scheme matches this rate of best N -term approximation. The question remains: what is the potential gain over *linear methods*, i.e., methods for which the trial spaces are a-priorily prescribed? Thus one ultimately has to face the following questions:

- How does the performance of a concrete adaptive scheme compare with best N -term approximation in ℓ_2 ?
- When is the performance of such an adaptive method better than that of *linear schemes*?

The first question essentially concerns approximation in ℓ_2 . The second question will be seen to draw in *regularity theory* of solutions to elliptic problems in certain *non-classical* scales of function spaces. These are function spaces that can be (nearly) characterized by best N -term approximation. The answer to both questions has only recently been given in [18]. We will proceed now with a brief review of these results that lead to concrete adaptive schemes.

2.7 An Adaptive Strategy

Our goal is to realize an estimate of the type (2.31). Again this will rely crucially on (2.29) (and hence on (2.8) and (2.2)) which, in particular means that

$$\|\mathbf{v}\| \sim \|\mathbf{v}\|_{\ell_2} \sim \|\mathbf{A}\mathbf{v}\| \sim \|\mathbf{A}\mathbf{v}\|_{\ell_2}. \quad (2.33)$$

In fact, for any $\hat{\Lambda} \supset \Lambda$ one has

$$\begin{aligned} \|\mathbf{u}_{\hat{\Lambda}} - \mathbf{u}_{\Lambda}\| &\gtrsim \|\mathbf{A}(\mathbf{u}_{\hat{\Lambda}} - \mathbf{u}_{\Lambda})\|_{\ell_2} \geq \|\mathbf{A}(\mathbf{u}_{\hat{\Lambda}} - \mathbf{u}_{\Lambda})|_{\hat{\Lambda}}\|_{\ell_2} \\ &= \|\mathbf{A}(\mathbf{u} - \mathbf{u}_{\Lambda})|_{\hat{\Lambda}}\|_{\ell_2}. \end{aligned}$$

Thus defining

$$\mathbf{r}_{\Lambda} := \mathbf{A}(\mathbf{u} - \mathbf{u}_{\Lambda}) = \mathbf{f} - \mathbf{A}\mathbf{u}_{\Lambda},$$

the above estimate says that for some constant $c_1 \in (0, 1)$ depending only on the constants in (2.33)

$$\|\mathbf{u}_{\hat{\Lambda}} - \mathbf{u}_{\Lambda}\| \geq c_1 \|\mathbf{r}_{\Lambda}|_{\hat{\Lambda}}\|_{\ell_2}. \quad (2.34)$$

Key Idea: If $\hat{\Lambda}$ can be chosen such that

$$\|\mathbf{r}_{\Lambda}|_{\hat{\Lambda}}\|_{\ell_2} \geq a \|\mathbf{r}_{\Lambda}\|_{\ell_2} \quad (2.35)$$

holds for some fixed $a \in (0, 1)$ then again (2.33) combined with (2.34) yields a constant $\beta \in (0, 1)$ such that (2.31) and hence, by Remark 2.4, also (2.30)

$$\|\mathbf{u}_{\hat{\Lambda}} - \mathbf{u}\| \leq \theta \|\mathbf{u}_{\Lambda} - \mathbf{u}\|$$

holds.

Thus the reduction of the error has been reduced to catching the *bulk* of the *residual* \mathbf{r}_Λ . This is a principal improvement since the residual involves only *known* quantities like the right hand side \mathbf{f} and the current solution \mathbf{u}_Λ . A second glance, however, damps optimism since among other things the realization of (2.35), i.e., catching the bulk of the current residual requires knowing *all* coefficients of the infinite sequence \mathbf{r}_Λ . Nevertheless, it will pay to neglect these issues for a moment and adhere to the above idea. Thus, assuming for the moment that $\hat{\Lambda}$ satisfies (2.35), a core ingredient of the refinement strategy can be summarized in the following (idealized) routine:

GROW $(\Lambda, \mathbf{u}_\Lambda) \rightarrow (\hat{\Lambda}, \mathbf{u}_{\hat{\Lambda}})$

Given $(\Lambda, \mathbf{u}_\Lambda)$ find the **smallest** $\hat{\Lambda} \supset \Lambda$ such that $\|\mathbf{r}_\Lambda|_{\hat{\Lambda}}\|_{\ell_2} \geq a\|\mathbf{r}_\Lambda\|_{\ell_2}$.

On the other hand, even if one is able to perform **GROW** it is by no means clear that such an algorithm is asymptotically optimal in the sense of best N -term approximation. Roughly speaking, the inversion of \mathbf{A} that leads from \mathbf{r}_Λ to $\mathbf{u} - \mathbf{u}_\Lambda$ may in quantitative terms smear too much. This will be explained later in more detail. However, it is shown in [18] that optimality can indeed be restored by a *clean up* step. This simply means that after several applications of **GROW** one has to discard all coefficients in the current approximation \mathbf{u}_Λ whose modulus is below a certain threshold. This threshold is chosen so that the current error is at most multiplied by a fixed uniform constant. While thereby the error gets only worse by a little this will turn out to have an essential effect on the behavior of the residual with respect to certain norms that are somewhat stronger than the ℓ_2 -norm. More details will be given later. We summarize this clean up or *thresholding step* again in an idealized form as follows:

THRESH $(\Lambda, \mathbf{u}_\Lambda) \rightarrow (\tilde{\Lambda}, \mathbf{u}_{\tilde{\Lambda}})$

If $\|\mathbf{u} - \mathbf{u}_\Lambda\|_{\ell_2} \leq \epsilon$ find **smallest** $\tilde{\Lambda} \subset \Lambda$ such that $\|\mathbf{u}_\Lambda - \mathbf{u}_\Lambda|_{\tilde{\Lambda}}\|_{\ell_2} \leq 4\epsilon$.
Note that both routines will ultimately require *sorting* coefficients.

2.8 An Optimal (Idealized) Algorithm

We next give a rough idealized version of an adaptive wavelet scheme whose practical counterpart will turn out to be optimal with respect to convergence rates as well as work count.

ALGORITHM

- $\Lambda_0 = \emptyset$, $\mathbf{r}_{\Lambda_0} = \mathbf{f}$, $\varepsilon_0 := \|\mathbf{f}\|_{\ell_2}$
- For $j = 0, 1, 2, \dots$ determine $(\Lambda_{j+1}, \mathbf{u}_{\Lambda_{j+1}})$ from $(\Lambda_j, \mathbf{u}_{\Lambda_j})$ such that

$$\|\mathbf{u} - \mathbf{u}_{\Lambda_{j+1}}\|_{\ell_2} \leq \varepsilon_j/2 := \varepsilon_{j+1}$$

as follows:

Set $\Lambda_{j,0} := \Lambda_j$, $\mathbf{u}_{j,0} := \mathbf{u}_j$;

For $k = 1, 2, \dots, K$ apply

GROW $(\Lambda_{j,k-1}, \mathbf{u}_{\Lambda_{j,k-1}}) \rightarrow (\Lambda_{j,k}, \mathbf{u}_{\Lambda_{j,k}})$
 $(\|\mathbf{r}_{\Lambda_{j,k-1}|\Lambda_{j,k}}\|_{\ell_2} \geq \frac{1}{2}\|\mathbf{r}_{\Lambda_{j,k-1}}\|_{\ell_2})$;

Apply **THRESH** $(\Lambda_{j,K}, \mathbf{u}_{\Lambda_{j,K}}) \rightarrow (\Lambda_{j+1}, \mathbf{u}_{\Lambda_{j+1}})$

The maximal number K of applications of **GROW** can be shown to be uniformly bounded depending only on the constants in (2.33). Moreover, a detailed description of fully computable version **ALGORITHM**^c of the above algorithm is given in [18] to which the following result refers.

Theorem 2.5 *The computable version **ALGORITHM**^c always produces a solution with the desired accuracy after a finite number of steps.*

Moreover, assume that $\mathbf{A} \in \mathcal{A}_s$ for $0 \leq s < s^$, recall Proposition 2.3. If the solution u to the operator equation (2.3) has the property that for some $s < s^*$*

$$\sigma_N(u) := \inf_{d_\lambda, \lambda \in \Lambda, \#\Lambda \leq N} \left\| u - \sum_{\lambda \in \Lambda} d_\lambda \psi_\lambda \right\| \lesssim N^{-s},$$

*then **ALGORITHM**^c generates a sequence u_{Λ_j} of Galerkin solutions to (2.6) satisfying*

$$\|u - u_{\Lambda_j}\| \lesssim (\#\Lambda_j)^{-s}. \quad (2.36)$$

Moreover, # of arithmetic operations needed to compute u_{Λ_j} stays proportional to $\#\Lambda_j$. The number of sorts stays bounded by $(\#\Lambda_j) \log(\#\Lambda_j)$.

It is important to note the above algorithm does *not* require any a-priori knowledge about the rate of N -term approximability of the solution. It is shown to automatically match the rate of best N -term approximation for a certain asymptotic range depending on the operator and the chosen basis.

2.9 When Does Adaptivity Pay?

Before turning to the discussion of realizing the actual ingredients of the above scheme a few comments on principal implications of Theorem 2.5 are in order. In particular, this will guide the selection of test examples.

First of all the theorem says that whenever the best possible rate of convergence in the given framework of wavelet expansions decays in a certain range with a power of the used degrees of freedom then the adaptive scheme matches this convergence rate and up to the number of involved sorts the computational work stays proportional to the number of degrees of freedom. That range of validity depends on the operator and the chosen wavelet basis.

The first question in Section 2.6 has now a positive answer in that the scheme realizes the best possible accuracy at a given allowance of degrees of freedom at nearly minimal cost.

As for the second question, the first remark is that such a polynomial decay N^{-s} of the error is in fact the relevant setting. Recall that *spectral methods* may even exhibit *exponential* decay but only when the solution is arbitrarily smooth throughout the domain. In the present setting we expect to deal with solutions with singularities. For any approximation scheme that is local in the sense that one can arrange only finitely many basis functions to overlap at a given point convergence rates are generally *saturated*, i.e., there is some maximal number α such that, regardless of the smoothness of the approximant, the error of best approximation decays at best like $N^{-\alpha}$ when N is again the number of degrees of freedom in the respective trial space. Multiresolution spaces and classical hierarchies of finite element spaces fall into this category. To make this a bit more concrete let us quickly review the following classical situation. Suppose that we have a nested sequence of (in the above sense) local approximation spaces S_j whose union exhausts $L_2(\Omega)$, $\Omega \subset \mathbb{R}^d$ say. Examples are sequences of finite element spaces obtained by *uniformly* halving the meshsize at each step and likewise any classical multiresolution analysis. Suppose that h_j is the *meshsize* associated with S_j . The approximation errors behave then as

$$\inf_{v_j \in S_j} \|v - v_j\|_{L_2(\Omega)} \lesssim h_j^r \|v\|_{H^r(\Omega)}, \quad v \in H^r(\Omega), \quad (2.37)$$

where r is limited by the maximal order of exactness of the trial spaces. Here S_j is called exact of order m if all polynomials (of total degree) can be represented (locally) by elements in S_j . Any smoothness beyond m does not help decreasing the asymptotic error. Of course, thinking of uniform mesh refinements one has

$$h_j^m \sim N_j^{-m/d},$$

where $N_j := \dim S_j$ and the constants in this relation depend only on the domain. In terms of the above theorem, the best rate that could be achieved by such a *preassigned* sequence of trial spaces S_j is N^{-s} with $s = m/d$ *provided that the solution has enough Sobolev regularity!*

Can one do better by an adaptive scheme based on the same type of multiresolution sequences, i.e., by working with progressively chosen subsets of the full spaces S_j as indicated above? It is important to stress first that the optimal rate $N^{-m/d}$ by itself cannot

be improved! However, the upshot is that such a good rate can be *preserved* by adaptive or best N -term approximation even when the approximated function *lacks* the Sobolev regularity needed to ensure the validity of an estimate like (2.37). The point here is that best N -term approximation can be (nearly) used to characterize spaces from another regularity scale, namely certain Besov spaces, see [37, 38, 36]. The type of result needed here can be formulated as follows [23]. Suppose again that $H = H^t$ for simplicity and define

$$\sigma_{N,t}(g) := \inf \left\{ \|g - \sum_{\lambda \in \Lambda} d_\lambda \psi_\lambda\|_{H^t} : d_\lambda \in \mathbb{R}, \lambda \in \Lambda \subset \mathcal{J}, \#\Lambda = N \right\}.$$

Let $\gamma > 0$ denote the supremum of all α such that $\Psi \subset H^\alpha$. Then the following holds [23].

Proposition 2.6 *Assume that $\alpha - t < \gamma$ and let for $t \leq \alpha$*

$$\frac{1}{\tau^*} := \frac{\alpha - t}{d} + \frac{1}{2}. \quad (2.38)$$

Then one has

$$\sum_{n=1}^{\infty} \left(N^{(\alpha-t)/d} \sigma_{N,t}(g) \right)^{\tau^*} < \infty \quad (2.39)$$

if and only if $g \in B_{\tau^}^\alpha(L_{\tau^*}(\Omega))$.*

Of course, (2.39) implies that the best N -term approximation in H^t (and hence the near best N -term approximation with respect to the energy norm) $\sigma_{N,t}(g)$ decays at least like $N^{-(\alpha-t)/d}$, provided that g is in $B_{\tau^*}^\alpha(L_{\tau^*}(\Omega))$. Note that (2.38) means that $B_{\tau^*}^\alpha(L_{\tau^*}(\Omega))$ is just embedded in H^t but need not have any excess Sobolev regularity beyond the energy space. Thus $B_{\tau^*}^\alpha(L_{\tau^*}(\Omega))$ is significantly larger than the Sobolev space $H^\alpha(\Omega)$. So exactly when the solution of (2.3) has a *higher* Besov regularity in the scale $B_{\tau^*}^\alpha(L_{\tau^*}(\Omega))$ than in the Sobolev scale, the above adaptive scheme produces an asymptotically *better* error decay in terms of the used unknowns than linear methods. It is important to stress here *asymptotic*. The implementation of adaptive schemes will always cause significant overhead and an error reduction by merely a *constant factor* might now pay off when comparing the overall work with the result. For large scale problems a better asymptotics will eventually pay off and justifies efforts for realizing adaptive schemes.

Therefore the next natural question is, does it occur in the context of elliptic problems that the solution has deficient Sobolev regularity compared with the scale $B_{\tau^*}^\alpha(L_{\tau^*}(\Omega))$? The answer is *yes* as shown, e.g., in [21, 25]. So there is a scope of problems where the above adaptive scheme would do better than linear methods. As an example, let us discuss a typical result in this direction which is concerned with Poisson's equation in a Lipschitz domain $\Omega \subset \mathbb{R}^d$,

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ u|_{\partial\Omega} &= 0. \end{aligned} \quad (2.40)$$

In this case, $A = -\Delta$ is an isomorphism from $H_0^1(\Omega)$ onto $H^{-1}(\Omega)$, so that it is natural to consider the best N -term approximation in $H^1(\Omega)$. Based on the investigations in [25], the following theorem was established in [23].

Theorem 2.7 *Let Ω be a bounded Lipschitz domain in \mathbb{R}^d . Let u denote the solution of (2.40) with $f \in B_2^{\mu-1}(L_2(\Omega))$ for some $\mu \geq 1$. Then the following holds:*

$$u \in B_{\tau^*}^\alpha(L_{\tau^*}(\Omega)), \quad \frac{1}{\tau^*} = \left(\frac{\alpha - 1}{d} + \frac{1}{2} \right), \quad 0 < \alpha < \min \left\{ \frac{d}{2(d-1)}, \frac{(\mu+1)}{3} \right\} + 1.$$

Due to singularities near the boundary, the Sobolev regularity of the solution u may not be very high, even for smooth right-hand sides. In fact, it is well-known that in general $u \in H^\alpha(\Omega)$, $\alpha \leq 3/2$, see, e.g., [41, 42] for details. Therefore Theorem 2.7 implies that for $\mu > 1/2$ the Besov regularity of u is in fact much higher than its Sobolev regularity so that adaptive methods should provide better asymptotic accuracy.

On the other hand, it should be kept in mind that the above dividing line depending on the different regularity scales is based on purely *asymptotic* reasoning and therefore may offer only a rather incomplete picture from a practical point of view. To obtain a more quantitative assessment of the error one should note that, in view of (2.37), for uniform refinements the *size* of the respective *Sobolev* norm matters while the error of best N -term approximation involves a *Besov* norm. So in spite of arbitrarily high pointwise smoothness it could well happen that the Besov norm of a function is much smaller than the Sobolev norm. In such a case the gain of efficiency accomplished by adaptive schemes could still be substantial in spite of high pointwise regularity.

2.10 Why GROW and THRESH?

A little more background information about the above **ALGORITHM**^c is helpful for identifying the computational tasks. Let us begin with **GROW**. Obviously, choosing $\hat{\Lambda}$ such that $\|\mathbf{r}_\Lambda - \mathbf{r}_\Lambda|_{\hat{\Lambda}}\|_{\ell_2} \leq \frac{1}{2}\|\mathbf{r}_\Lambda\|_{\ell_2}$ would imply $\|\mathbf{r}_\Lambda|_{\hat{\Lambda}}\|_{\ell_2} \geq \frac{1}{2}\|\mathbf{r}_\Lambda\|_{\ell_2}$. Thus we essentially have to find a sequence in ℓ_2 with possibly small support that approximates the true residual as well as possible in ℓ_2 – again the task of best N -term approximation. Specifically, given $\epsilon := \frac{1}{2}\|\mathbf{r}_\Lambda\|_{\ell_2}$, what is the smallest N such that the error of best N -term approximation stays below ϵ ? To derive quantitative error estimates requires identifying suitable compactly embedded subspaces of ℓ_2 . The subspaces that are characterized by best N -term approximation in ℓ_2 are well-known [36, 18]. They are special cases of *Lorentz sequence spaces* defined as follows. Let for $\mathbf{v} \in \ell_2$ the *nonincreasing rearrangement* of \mathbf{v} be denoted by $\mathbf{v}^* = \{v_n\}_{n \in \mathbb{N}}$, i.e., $v_n^* \geq v_{n+1}^*$ and $v_n^* = |v_\lambda|$ for some $\lambda \in \mathcal{J}$, (which is not unique, but terms with equal modulus can be ordered arbitrarily). Let for $0 < \tau < 2$

$$|\mathbf{v}|_{\ell_\tau^w} := \sup_{n \in \mathbb{N}} n^{1/\tau} |v_n^*|, \quad \|\mathbf{v}\|_{\ell_\tau^w} := \|\mathbf{v}\|_{\ell_2} + |\mathbf{v}|_{\ell_\tau^w}. \quad (2.41)$$

It is easy to see that

$$\|\mathbf{v}\|_{\ell_\tau^w} \leq 2\|\mathbf{v}\|_{\ell_\tau}, \quad (2.42)$$

so that by Jensen's inequality, in particular, $\ell_\tau^w \subset \ell_2$. Moreover, let \mathbf{v}_N denote the restriction of \mathbf{v} to its N largest terms (the first N terms in \mathbf{v}^*). Clearly, $\|\mathbf{v} - \mathbf{v}_N\|_{\ell_2}$ realizes the error of best N -term approximation to \mathbf{v} in ℓ_2 . The following characterization can be shown [18, 36].

Proposition 2.8 *Let*

$$\frac{1}{\tau} = s + \frac{1}{2}, \quad (2.43)$$

then

$$\mathbf{v} \in \ell_\tau^w \iff \|\mathbf{v} - \mathbf{v}_N\|_{\ell_2} \lesssim N^{-s} \|\mathbf{v}\|_{\ell_\tau^w}. \quad (2.44)$$

Remark 2.9 *Note that in these terms the assumption in Theorem 2.5 on u is equivalent to saying $\mathbf{u} \in \ell_\tau^w$. This can be viewed as a regularity assumption. In fact, by (2.42) $\mathbf{u} \in \ell_\tau$ implies $\mathbf{u} \in \ell_\tau^w$. But recall from (2.11) and (2.12) that $\mathbf{u} \in \ell_\tau$ means that $\mathbf{u}^T \Psi \in B_\tau^{sd}(L_\tau)$ for s and τ related through (2.43). Thus when $H = H^t$ this means for $\mathbf{D}_{\lambda,\lambda} = 2^{t|\lambda|}$, on account of (2.15), $u \in B_\tau^{sd+t}(L_\tau)$, recall Proposition 2.6.*

Thus the N needed to ensure $\|\mathbf{v} - \mathbf{v}_N\|_{\ell_2} \leq \varepsilon$ is of the order $N \sim \varepsilon^{-1/s} \|\mathbf{v}\|_{\ell_\tau^w}^{1/s}$. Applying this to the above task of approximating the residual \mathbf{r}_Λ , one finds

$$\#(\hat{\Lambda} \setminus \Lambda) \sim \left(\frac{\|\mathbf{r}_\Lambda\|_{\ell_\tau^w}}{\|\mathbf{r}_\Lambda\|_{\ell_2}} \right)^{1/s}. \quad (2.45)$$

Since, by (2.33) $\|\mathbf{u} - \mathbf{u}_\Lambda\| \sim \|\mathbf{r}_\Lambda\|_{\ell_2}$ the error would exhibit the right relation to $\#\Lambda$ provided that $\|\mathbf{r}_\Lambda\|_{\ell_\tau^w}$ stays uniformly bounded. This is therefore the key requirement to be satisfied.

2.11 How to Bound $\|\mathbf{r}_\Lambda\|_{\ell_\tau^w}$?

We now turn to the routine **THRESH**. Looking at the definition of $\|\cdot\|_{\ell_\tau^w}$ one realizes that this norm will tend to become large when the sequence has many entries of more or less equal but small modulus. Thus, removing these entries would not increase the error in ℓ_2 very much but may reduce the $\|\cdot\|_{\ell_\tau^w}$ significantly. Hence *thresholding* may help. However, since not all entries of \mathbf{r}_Λ are accessible we cannot work directly on \mathbf{r}_Λ . The key observation here is that only *thresholding the current approximate Galerkin solution \mathbf{u}_Λ* will control $\|\mathbf{r}_\Lambda\|_{\ell_\tau^w}$.

At this point the compressibility of \mathbf{A} , recall Proposition 2.3, comes into play.

Proposition 2.10 [18] *Any \mathbf{A} in \mathcal{A}_s is not only bounded on ℓ_2 but also on ℓ_τ^w , i.e.,*

$$\|\mathbf{A}\mathbf{v}\|_{\ell_\tau^w} \lesssim \|\mathbf{v}\|_{\ell_\tau^w}. \quad (2.46)$$

Thus it suffices to control $\|\mathbf{u}_\Lambda\|_{\ell_\tau^w}$ in order to keep $\|\mathbf{r}_\Lambda\|_{\ell_\tau^w}$ uniformly bounded. This in turn has been shown in [18] to be possible by thresholding the current Galerkin solution \mathbf{u}_Λ . To this end, define the thresholding operator

$$(\mathcal{T}_\eta \mathbf{v})_\lambda := \begin{cases} v_\lambda & \text{if } |v_\lambda| \geq \eta; \\ 0 & \text{if } |v_\lambda| < \eta. \end{cases}$$

The fact that the right amount of thresholding keeps the $\|\cdot\|_{\ell_\tau^w}$ -norm small while preserving the order of the ℓ_2 -error is based on the following observations from [18].

Lemma 2.11 For any $\Lambda \subset \mathcal{J}$ one has

$$\|\mathbf{u}_\Lambda\|_{\ell_\tau^w} \lesssim \|\mathbf{u}\|_{\ell_\tau^w} + (\#\Lambda)^s \|\mathbf{u} - \mathbf{u}_\Lambda\|_{\ell_2}. \quad (2.47)$$

Lemma 2.12 Given $\mathbf{w} \in \ell_w^\tau$ and assume that $\mathbf{v} \in \ell_2$ satisfies

$$\|\mathbf{v} - \mathbf{w}\|_{\ell_2} \leq \epsilon. \quad (2.48)$$

Then, setting $\eta := \epsilon^{1/s\tau}$, one has

$$\|\mathcal{T}_\eta \mathbf{v} - \mathbf{w}\|_{\ell_2} \leq c\epsilon \|\mathbf{w}\|_{\ell_\tau^w}^{\tau/2},$$

and

$$\#\{\lambda \in \mathcal{J} : (\mathcal{T}_\eta \mathbf{v})_\lambda \neq 0\} \leq c\epsilon^{-1/s} \|\mathbf{u}\|_{\ell_\tau^w}^\tau. \quad (2.49)$$

In particular, this will be applied to $\mathbf{w} = \mathbf{u}$ and $\mathbf{v} = \mathbf{u}_\Lambda$ (but also to various other instances caused by inexact computations). In fact, substituting the bound on $\#\Lambda$ from (2.49) in the right hand side of (2.47) shows, in view of (2.48), that $\|\mathbf{u}_\Lambda\|_{\ell_\tau^w}$ and hence, by Proposition 2.10 also $\|\mathbf{r}_\Lambda\|_{\ell_\tau^w}$ stays indeed bounded. This explains the relevance of the routine **THRESH**.

Proposition 2.10 can be proved with the aid of Proposition 2.8. In fact, it suffices to exhibit a sufficiently good approximation to the matrix vector product $\mathbf{A}\mathbf{v}$ involving only N terms. This is a further ingredient of central importance and will be explained below.

2.12 Fast Approximate Matrix/Vector Multiplication

The following fact is proved in [18].

Proposition 2.13 Defining $\mathbf{v}_{[j]} := \mathbf{v}_{2^j}$ (best N -term approximation for $N = 2^j$) and

$$\mathbf{w}_j := \mathbf{A}_j \mathbf{v}_{[0]} + \mathbf{A}_{j-1}(\mathbf{v}_{[1]} - \mathbf{v}_{[0]}) + \cdots + \mathbf{A}_0(\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}), \quad (2.50)$$

then

$$\|\mathbf{A}\mathbf{v} - \mathbf{w}_j\|_{\ell_2} \lesssim 2^{-sj} \|\mathbf{v}\|_{\ell_\tau^w}. \quad (2.51)$$

As a consequence the computational work $\mathbf{CW}(\eta)$ needed to realize an approximation \mathbf{w}_η to $\mathbf{A}\mathbf{v}$ such that $\|\mathbf{A}\mathbf{v} - \mathbf{w}_\eta\|_{\ell_2} \leq \eta$ is of the order

$$\mathbf{CW}(\eta) \sim \#\text{supp } \mathbf{w}_\eta \lesssim \eta^{-1/s} \|\mathbf{v}\|_{\ell_\tau^w}^{1/s}, \quad (2.52)$$

which is again of the right form.

2.13 Computational Tasks

We are now in a position to identify the concrete computational tasks required by a computable version of **ALGORITHM**. A detailed account of these routines can be found in [18] where, in particular, various parameters are identified that steer the refinement process. Here we focus on principle issues arising in the implementation of these routines. Of course, the central issue is to determine the bulk of \mathbf{r}_Λ or, equivalently, to find a good approximation to \mathbf{r}_Λ in ℓ_2 . In this context one faces the following obvious obstructions:

- (i) One has to determine first the Galerkin solution \mathbf{u}_Λ . Even if \mathbf{u}_Λ could be determined *exactly* one cannot compute the infinite vector $\mathbf{A}\mathbf{u}_\Lambda$ to determine the residual.
- (ii) In order stay within the promised bounds of computational complexity the corresponding linear systems *cannot* be solved exactly. Instead one obtains only an approximation $\bar{\mathbf{u}}_\Lambda$ to \mathbf{u}_Λ . Again one cannot compute the infinite vector $\mathbf{A}\bar{\mathbf{u}}_\Lambda$.

Thus at each stage of the **ALGORITHM** one has to be content with an *approximation* to the residual and its bulk. The errors incurred in such approximations are as follows:

$$\mathbf{r}_\Lambda = \mathbf{f}_\eta - \mathbf{w}_\eta + \underbrace{\mathbf{f} - \mathbf{f}_\eta + \mathbf{A}(\bar{\mathbf{u}}_\Lambda - \mathbf{u}_\Lambda) + \mathbf{w}_\eta - \mathbf{A}\bar{\mathbf{u}}_\Lambda}_{\text{error}}. \quad (2.53)$$

At each stage of the **ALGORITHM** this error has to be kept below a certain level η say, specified in [18]. This amounts to the following tasks:

- 1) Determine a sufficiently good approximation \mathbf{f}_η .
- 2) Determine $\bar{\mathbf{u}}_\Lambda$ by an iterative scheme. This requires repeated matrix/vector multiplications.
- 3) Compute an approximation \mathbf{w}_η to $\mathbf{A}\bar{\mathbf{u}}_\Lambda$.
- 4) Find a best N -term approximation to the resulting approximation (2.53) (or keep it as it is).
- 5) Threshold the current approximate Galerkin solution.

Clearly 2) and 3) will heavily rely on the above *matrix/vector multiplication* (2.50). The estimates from previous sections will ultimately ensure that only a finite uniformly bounded number of iterations will be needed at each stage to fulfill the accuracy requirements for the next step. In particular, (2.52) will guarantee that the computational work stays in the desired bounds.

Note next that 1), 4) and 5) involve *thresholding* of a known array, at least conceptually. The way it is needed here is to discard the largest possible number of small entries so that a desired accuracy is preserved by this perturbation. The core task there is to first *sort* the arrays and then sum successively entries in increasing order. This is also used in the error control of the fast matrix/vector multiplication because the algorithm should at no stage use any a-priori assumption about the membership of \mathbf{u} in any of the ℓ_τ^w spaces.

These remarks shed some light on the role of *sorting* and the *fast matrix/vector multiplication* in the whole context. In the following we will discuss some consequences of these facts. In particular, it will be seen that most of the data structures needed here can be designed *independently* of the particular application and even of the particular wavelet basis. The special application enters primarily through calling the significant entries in the columns of \mathbf{A} when performing (2.50). This will be exemplified later in connection with applications.

3 Documentation and Related Issues

One key ingredient for the realization of the adaptive algorithm presented above is the organization of the data, i.e., how to store the active coefficients. The data must be organized in such a way, that the benefits of the adaptive method predicted by the theory is not wasted by a large overhead of data management. Clearly, a certain overhead can not be circumvented: for uniform methods the number of unknowns is a priori known (each level has fixed number of unknowns), they can be organized in static vectors containing all coefficients. Scalar products for example of these vectors are fast on modern computer architectures since the optimizer of the compiler can use the floating point unit of the computer in the most efficient way. Of course, this is no longer the case for adaptive methods: the number of active unknowns is determined during the algorithm.

Therefore suitable data structures providing flexible and efficient storage and allowing fast sorting have to be used. In contrast to uniform methods based on level wise oriented structures, i.e., using vectors, we have to use data structures focusing on individual coefficients. This has to be done ensuring that the overhead produced by the data structures is much smaller than the gain of efficiency by using the adaptive method.

3.1 Key-Based Data Structures

The type of data structures which fits our purpose best are *key based* data structures: the data are divided into two parts, namely the *key*, for example for the wavelets the index λ , and the *value*, i.e., the entry d_λ . So every item forms a *pair* (*key*, *value*). Formally we can view this kind of data structure as a mapping from the set of keys to the set of admissible values:

$$map : key \in Keys \rightarrow value \in Values. \quad (3.1)$$

As long as the key is unique this is sufficient. For our wavelet expansion $\mathbf{u}_\Lambda = \mathbf{d}_\Lambda \Psi_\Lambda$ we assume that every $\lambda \in \Lambda$ is unique, otherwise one would combine two coefficients to one. So this expansion reads like

$$map_\Lambda : \lambda \in \Lambda \rightarrow d_\lambda \in \mathbb{R}. \quad (3.2)$$

The function *map* becomes a *sorted map* whenever we have some ordering on the *Keys*, i.e., there exists a transitive relation

$$less := \{(key_1, key_2) \in Keys \times Keys : key_1 < key_2\} \subset Keys \times Keys. \quad (3.3)$$

The index set Λ allows several possibilities of ordering among which we chose the following: first level wise, then by each wavelet type, then by subdomain and within each subdomain by a lexicographical ordering of the translation index.

So the definition of our data structure *map* relies on having

- a unique encryption for $\lambda \in \Lambda$,
- and a relation *less* for these keys.

The data structure realizing *map* must meet the following requirements with respect to the size $N = |\text{Keys}|$:

- complexity to find/erase an individual element is at most logarithmic,
- logarithmic complexity for insertion of a new element, so the overall time for building/sorting the map is at most $N \log(N)$.

The data structure `map` from the *Standard Template Library*, STL, see [47], matches exactly these requirements. In addition, it is a *generic* class like most of the classes in the STL. This means, that the type of the key and the type of the values are arbitrary, they serve as parameters only. So the data structure is independent of the type of wavelets one uses. Only the key representing the wavelet has to fulfill the requirements above. In C++ generic classes are provided by *templates*, classes with types as parameters. These classes are called *containers*, classes containing some elements of some type. Typical examples of containers are `vectors`, `lists` or `maps`. To define a `map` for our problem reads like

```
coeff_sorted_by_index := map<index, double, index::less_than>
```

i.e., it is a `map`, where the key is given by a class `index`, the values are double precision real numbers and the ordering is given by the `less_than` function of the class `index`.

During the execution of the algorithm for estimating the residual, the coefficients need to be sorted according to their absolute value. So if we think of interchanging the role of the key and the value of our `map`, we loose uniqueness, because several coefficients may have the same value. There is no unique mapping from the values of the coefficients to the indices carrying these values. A slightly more general class `multimap` from the STL is suitable for this case. For a `multimap` the uniqueness of the key is not necessary, so the sorting is also not unique any more. If $N = |\text{Keys}|$ and $n = \text{Number of elements with the same key}$ the complexities of the required operations in `multimap` are

- $\log(N) + n$ time to find or erase all elements with this key,
- insertion of a new element requires $\log(N)$ operations.

In our case the definition reads like

```
coeff_sorted_by_value := multimap<double, index, less_absolute>
```

where the coefficients are sorted with respect to the function `less_absolute` which compares the absolute values.

3.2 Generic Algorithms

To actually work with these classes one has to understand the concept of *iterators*, sometimes also called *generalized pointers*. Next to the templates iterators are a core ingredient making the STL independent of the data used. Every container class has a function `begin()` returning an iterator pointing to the first entry of the container. Addressing each element of the container amounts to incrementing the iterator until the last element of the container is reached.

All routines have to use this concept of iterators for those parts which do not depend on the type of wavelets used. As an example we show a key ingredient of the adaptive algorithm in more detail, namely the fast matrix/vector multiplication defined in Section 2.12:

$$\mathbf{w}_j := \mathbf{B}_j \mathbf{v}_{[0]} + \mathbf{B}_{j-1} (\mathbf{v}_{[1]} - \mathbf{v}_{[0]}) + \cdots + \mathbf{B}_0 (\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}). \quad (3.4)$$

The vectors $\mathbf{v}_{[i]}$ were defined by retaining from the decreasing rearrangement of \mathbf{v}^* only the first 2^i entries. Therefore the sorted vector $\mathbf{v}_{[j]}$ can be written as

$$\mathbf{v}_{[j]} = (\mathbf{v}_{[0]}^T, (\mathbf{v}_{[1]} - \mathbf{v}_{[0]})^T, \dots, (\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]})^T), \quad (3.5)$$

i.e., $\mathbf{v}_{[j]} - \mathbf{v}_{[i-1]}$ can be seen as a section of the vector $\mathbf{v}_{[j]}$ with size 2^{i-1} . The implementation of the fast matrix vector multiplication reads as follows:

```
void FastMatVecMult(map<index, double, index::less_than> &w_j,
                   const multimap<double, index, less_absolute> v_lambda,
                   const stiffnessmatrix &A,
                   int j)
{
    int jj = 0, count = 0;

    multimap<double, index, less_absolute>::iterator nu;

    for (nu = v_lambda.begin(); nu != v_lambda.end() && jj<=j; nu++)
    {
        map<index, double, index::less_than> Column;

        Column = ColumnSet ( (*nu).second, j-jj);

        map<index, double, index::less_than>::iterator mu;

        for (mu = Column.begin(); mu != Column.end(); mu++)
            w_j[( *mu).first] += A(( *mu).first, (*nu).second) * (*nu).first;

        count++;
        jj = int(log(count)/log(2)) + 1;
    }
}
```

The function `FastMatVecMult` consists of two iterations: the outer iteration on `nu` iterates through the elements of the vector `v_lambda`. This vector is of type `multimap<double, index, less_absolute>` and therefore sorted in a decreasing order. For one entry of the vector, `(*nu).second` denotes the wavelet index of this entry, whereas `(*nu).first` the wavelet coefficient. For this index the function `Columnset` computes the set $\Lambda(\nu, j - jj)$ defined in (2.23), (2.24).

Now the inner iteration on `mu` iterates through this set, adding up the products of the vector element ν with the corresponding matrix element. To take the correct \mathbf{B}_i in the sum (3.4) we have to know the section of `lambda_v` we are working with. This amounts to taking the \log_2 of the variable `count`.

For the whole routine only two functions are specific to the problem at hand, namely the routine `ColumnSet` providing the index set $\Lambda(k_\nu, j_\nu, j - jj)$ and the evaluation of the inner product $a(\psi_\mu, \psi_\nu)$ for two wavelets.

3.3 Discussion of Various Specific Routines

In this section we want to briefly discuss other routines needed for our adaptive algorithm. A detailed description of these routines is given in [18]. To this end we start with

Convert

A `coeff_sorted_by_value` can be converted into a `coeff_sorted_by_index` and vice versa. The amount of work is $N \log N$, where N is the size, since reading the elements is of constant time and the building of the new type is $N \log N$.

For the following routines we assume the input to be of `coeff_sorted_by_value`, otherwise use `Convert`.

Best N-term approximation, see Section 2.6

Building the best N -term approximation amounts to erasing all but the first N elements.

Threshold, see Section 2.7

Starts with the first element of \mathbf{u}_Λ , iterate until `value` is less than the prescribed tolerance and erase all following elements.

Bulk

Start with the first entry of \mathbf{u}_Λ , insert it into an empty `coeff_sorted_by_value` and calculate its norm. Proceed analogously until it is larger than the given bound.

APPLYA, see Section 2.12

The application of the fast matrix/vector multiplication `APPLYA` consists of calculating j needed in (3.4) and calling the function `FastMatVecMult`.

NRESIDUAL

`NRESIDUAL` calls `APPLYA` and subtracts an approximation of the righthand side.

NGALERKIN

For the Galerkin solver we used an ordinary *conjugent gradient solver* where we replaced each matrix-vector multiplication by `FastMatVecMult`. Again see [18] for the algorithmic details.

NGROW, see Section 2.7

First call the function **NCOARSE** is called for the righthand side, afterwards in a loop first **NRESIDUAL** to estimate the current residual, use **Bulk** to take a fixed portion and compute for this new index set the approximate solution with **NGALERKIN**.

ColumnSet

In **FastMatVecMult** we used the function **ColumnSet**. This function computes the set $\Lambda(\lambda, J)$ in (2.23), (2.24). For our example of the second order differential operator these are the coefficients of those functions, whose support intersect the support of the given function, i.e., whose entries in the stiffness matrix do not vanish, along with a cut-off criterion of level differences. In fact, for an index $\lambda = (j, k)$ we define

$$\Lambda(\lambda, J) := \left\{ \nu = (j', k') \in \nabla : \text{supp } \psi_\lambda \cap \text{supp } \psi_\nu \neq \emptyset, |j - j'| \leq J \right\}. \quad (3.6)$$

Of course, this function differs significantly for the one dimensional and the two dimensional case. In the one dimensional case this is a tedious but straight forward calculation. The supports of the functions are given by scaling and translation of the support of one generator and one wavelet. This leads to some simple calculation of which translates for which level has to be taken. The most difficult part is to catch the appropriate functions near the boundary, because their support differs from the general formula.

The problem is much more involved for the 2D-case. While simple again in the interior of a subdomain where we can use a tensor product version of the one dimensional routine, it becomes very delicate near the interior boundaries, where functions are supported on more than one subdomain. For our test problem we designed a very simple geometry representation containing information about the connectivity of the subdomains. Up to now it is restricted to parametric mappings consisting of translations of the reference domain.

With this information it is possible to identify for each function 'near' the boundary those functions on other subdomains with intersecting supports. There are some cases which have to be taken care of: a function is supported near an interior boundary but within one subdomain. Nevertheless its support is overlapped by the support of both those functions living on the boundary and those wavelets in the neighboring subdomain overlapping the boundary.

If the function itself is a wavelet overlapping an interior boundary not only those functions living on the other side of the boundary and overlap the boundary must be considered but also those functions which are fully supported inside the other subdomain but whose support is overlapped by the support of the considered wavelet.

This is especially difficult at the corner where the three subdomains meet: here supports of functions overlap, where the subdomains are no neighbors in the sense that they have a common boundary. At this point the function **ColumnSet** is not general but in some sense restricted to the case of the L-shaped domain. Of course it will be another task for the future to overcome this restriction.

To give an impression on the difficulty of identifying all these cases: this part of the code takes about 1700 lines, whereas the complete adaptive algorithm including this part takes about 4000 lines only. These numbers of course do not include the complete construction of the wavelets and the other routines which existed before.

The classes `matrix_column` and `smatrix`

These two classes are designed to contain entries of the stiffness matrix, calculated during the adaptive algorithm. The class `smatrix` representing the stiffness matrix is organized as a `map` of `matrix_columns`. Since our set $\Lambda(\lambda, J)$ is based on level differences the class `matrix_column` consists of a dynamical array of `coeff_sorted_by_index`. For each active coefficient λ the corresponding `matrix_column` contains all entries of the stiffness matrix with respect to $\Lambda(\lambda, J)$, i.e., all $\nu \in \Lambda(\lambda, J), a(\psi_\nu, \psi_\lambda) \neq 0$. If J changes, the needed levels of coefficients are added.

Calculation of the entries in the stiffness matrix

The major obstacle of our current realization is the computation of the entries of the stiffness matrix, i.e., the evaluation of the bilinear form a from (2.1) for two wavelets:

$$a_{\mu,\nu} := a(\psi_\mu, \psi_\nu). \quad (3.7)$$

In our example this amounts to the integration of the product of two wavelets and their derivatives. For scaling functions this is very easy whereas for wavelets this is not so clear how to realize. In the uniform case one can use a simple trick: to set up the stiffness matrix in the single scale basis as a sparse matrix, and use the fact, that one can represent the stiffness matrix in the multiscale basis as the product of the entire stiffness matrix with respect to the single scale basis with the multiscale transformation matrices for the change of basis between single scale and multiscale representation.

Since in a uniform method we don't need individual entries of the matrix, but for the linear solver only the action of the stiffness matrix on a vector, this can be realized efficiently, namely by first using the fast transformation from multiscale to single scale basis, multiply with the (sparse) single scale stiffness matrix and use the fast transformation back to the multiscale representation.

In our context this clearly is no suitable strategy. We plan to compute the entries by numerical integration as it is already done for boundary integral equations, see [44]. Promising efforts to adapt the routines presented there are currently made.

But for the moment we still have to use the following provisional alternative: those columns of the stiffness matrix needed during the adaptive algorithm are computed by using the uniform method for the corresponding unit vector. The result is stored for future use. Of course this leads to quite a limitation, both on the refinement depth we can use and the time and memory consumption, which both are dominated by this alternative. Nevertheless the adaptive algorithm proves to produce good approximations and we believe it will show its full power when combined with an appropriate computation of the bilinear form.

The class `Problem`

The class `Problem` is designed to define the problem we want to solve, i.e., the operator, the domain, the righthand side, the type of wavelets used. It estimates the various constants needed in the algorithm. Functions operating on this class are defined *virtual*, i.e., they can be redefined in an inherited class. This makes it possible to have a common interface, whereas the specific implementation can be changed. Therefore all functions concerning the adaptive algorithm are implemented (with still some exceptions, of course) independently of the specific problem at hand.

4 Numerical Tests

We are now prepared to begin with first realizations of the above adaptive scheme. A few introductory comments in this regard are in order. It is clear from the above developments that a full exploitation of the conceptual power requires new algorithmic ingredients partly designed from scratch. Specifically, for the problem dependent part little can be borrowed from existing software. Consequently not all parts of the whole construction site could be brought to a mature state so that we are still far from having a complete picture. Nevertheless, we think that we have reached a stage where first conclusions are justified even when accepting certain compromises to be detailed later in connection with concrete cases. In fact, such a summary of affairs will provide valuable guidelines for further developments.

Concerning the principal range of applicability, recall that a wide scope of problems is covered including boundary value problems on open domains in Euclidean space as well as boundary integral equations on closed manifolds. However, to keep the demands on sophisticated geometry representations at a minimum we will confine here the discussion to second order elliptic boundary value problems in one and two spatial dimensions. These examples are simple but nevertheless very instructive. Recall that adaptivity is expected to pay off best when the solution exhibits singularities. First we consider one 1D-model problem where the singularities of the solution are only caused by strong gradients of the right-hand side. To our knowledge, most of the earlier studies of adaptive wavelet schemes are concerned with periodic problems in order to avoid the more complicated construction of wavelets for domains. In this setting one can only expect data induced singularities and we wish to confirm that the present scheme lives up to theoretical predictions in this case as well. A detailed documentation will be given in Section 4.2.

The second class of examples is concerned with more sophisticated problems on non-smooth domains in \mathbb{R}^2 . In these cases, there occur also singularities which are not generated by the right-hand sides but by the shape of the domain. Such examples have not been studied before in the wavelet context. Therefore, the quantitative performance of the scheme should be very instructive. In order to validate the results we will exploit the theoretical knowledge about singularity solutions that is available for polygonal domains. The algorithm, of course, does not assume any of such information as input. A detailed description will be presented in Section 4.3.

Finally, the type of test problems also determines the demands on the central tools, namely the wavelets. The prize that has to be paid for a relatively far reaching analysis is to employ bases with sophisticated properties as detailed in Sections 2.2 and 2.3. It is meanwhile understood how to construct wavelet bases with the desired properties for essentially all cases of interest. Roughly speaking the treatment of operators of order less than or equal to $-1/2$ requires a little more sophisticated constructions [32, 30]. For the situation considered here more candidates are available and the constructions in [14, 15, 19, 31] would work equally well. They form all Riesz bases in $L_2(\Omega)$ and induce isomorphisms of the type (2.8) for Sobolev spaces H^s for $-1/2 < s < 3/2$ well covering the relevant case H^1 in the present context. For the particular implementation discussed below we have chosen the construction form [31]. It is conceptually fairly transparent so that the relevant information concerning the data structures is not too hard to extract.

4.1 The Construction Principle

The construction of the wavelets in all subsequent applications is based on a common principle which we will sketch first. It consists of three steps:

- a) Construct *dual pairs* of generator bases

$$\Phi_j = \{\phi_{j,k} : k \in \Delta_j\}, \quad \tilde{\Phi}_j = \{\tilde{\phi}_{j,k} : k \in \Delta_j\},$$

i.e.,

$$\langle \Phi_j, \tilde{\Phi}_j \rangle := \left(\langle \phi_{j,k}, \tilde{\phi}_{j,l} \rangle \right)_{j,l \in \Delta_j} = \mathbf{I}, \quad (4.1)$$

whose elements have local support

$$\text{diam}(\text{supp } \phi_{j,k}) \sim 2^{-j}, \quad \text{diam}(\text{supp } \tilde{\phi}_{j,k}) \sim 2^{-j}, \quad (4.2)$$

such that their spans

$$S(\Phi_j) := \text{span } \Phi_j, \quad S(\tilde{\Phi}_j) := \text{span } \tilde{\Phi}_j,$$

are nested

$$S(\Phi_j) \subset S(\Phi_{j+1}), \quad S(\tilde{\Phi}_j) \subset S(\tilde{\Phi}_{j+1}). \quad (4.3)$$

$S_j(\Phi_j)$, $S_j(\tilde{\Phi}_j)$ are referred to as *primal* and *dual* multiresolution spaces.

- b) Find a stable basis $\check{\Psi}_j$ of *some* complement $S(\check{\Psi}_j)$ of $S(\Phi_j)$ in $S(\Phi_{j+1})$, i.e.,

$$S(\Phi_{j+1}) = S(\Phi_j) \oplus S(\check{\Psi}_j), \quad j = j_0, \dots \quad (4.4)$$

- c) Given such an initial decomposition *project* the initial basis $\check{\Psi}_j$ into a basis Ψ_j which is perpendicular to $S(\tilde{\Phi}_j)$. The union $\Psi := \Phi_{j_0} \cup \bigcup_{j=j_0}^{\infty} \Psi_j$ will be the final primal wavelet basis satisfying the requirements from Sections 2.2 and 2.3.

This program can be carried out for various types of domains. We will use it for $\Omega = (0, 1)$, the unit 2-cube and for domains Ω that are disjoint unions of smooth parametric images of the unit cube (where in this order each stage builds upon the preceding one). The point is that only steps a) and b) depend on the particular situation at hand while c) can be achieved by a general mechanism which we briefly describe now. To this end, note that (4.3) states that each coarse scaling function can be written as a linear combination of fine scale basis functions. Viewing the bases as vectors whose components are the individual scaling functions (4.3) is equivalent to saying that there must be $\#\Delta_{j+1} \times \Delta_j$ matrices $\mathbf{M}_{j,0}$, $\tilde{\mathbf{M}}_{j,0}$ such that

$$\Phi_j^T = \Phi_{j+1}^T \mathbf{M}_{j,0}, \quad \tilde{\Phi}_j^T = \tilde{\Phi}_{j+1}^T \tilde{\mathbf{M}}_{j,0}. \quad (4.5)$$

Likewise there must be $\#\Delta_{j+1} \times (\#\Delta_{j+1} - \#\Delta_j)$ matrices $\check{\mathbf{M}}_{j,1}$ such that $\check{\Psi}_j^T = \Phi_{j+1}^T \check{\mathbf{M}}_{j,1}$ and the stability of $\check{\Psi}_j$ required in step b) is equivalent to saying that the composed matrices $\check{\mathbf{M}}_j := (\mathbf{M}_{j,0}, \check{\mathbf{M}}_{j,1})$ are invertible and both $\|\check{\mathbf{M}}_j\|$ and $\|\check{\mathbf{M}}_j^{-1}\|$ are uniformly bounded. In this latter case $\check{\mathbf{M}}_{j,1}$ is called a *stable completion* of $\mathbf{M}_{j,0}$ [16].

Given a dual pair of generator bases Φ_j , $\tilde{\Phi}_j$ as above and *some initial* stable completions $\check{\mathbf{M}}_{j,1}$, *biorthogonal* wavelet bases can be obtained as follows [16].

Proposition 4.1 For $\Phi_j, \tilde{\Phi}_j$ as above and some stable completion $\check{M}_{j,1}$ of $M_{j,0}$ let $\check{G}_j := \check{M}_j^{-1}$. Then

$$M_{j,1} := (\mathbf{I} - M_{j,0} \check{M}_{j,0}^T) \check{M}_{j,1} \quad (4.6)$$

is also a stable completion and $G_j = M_j^{-1}$ has the form

$$G_j = \begin{pmatrix} \check{M}_{j,0}^T \\ \check{G}_{j,1} \end{pmatrix}. \quad (4.7)$$

Moreover, the collections

$$\Psi_j := M_{j,1}^T \Phi_{j+1}, \quad \tilde{\Psi}_j := \check{M}_{j,1}^T \tilde{\Phi}_{j+1} \quad (4.8)$$

form biorthogonal systems,

$$\langle \Psi_j, \tilde{\Psi}_j \rangle = \mathbf{I}, \quad \langle \Psi_j, \tilde{\Phi}_j \rangle = \langle \Phi_j, \tilde{\Psi}_j \rangle = \mathbf{0}, \quad (4.9)$$

i.e.,

$$\Psi := \Phi_{j_0} \cup \bigcup_{j \geq j_0} \Psi_j, \quad \tilde{\Psi} := \tilde{\Phi}_{j_0} \cup \bigcup_{j \geq j_0} \tilde{\Psi}_j, \quad (4.10)$$

are biorthogonal wavelet bases of the type needed in Sections 2.2, 2.3.

In practice, one needs not to perform the global matrix multiplications reflected by (4.6) to compute the entries of $M_{j,1}$. In fact, one can derive an alternative representation [16]

$$M_{j,1} = \check{M}_{j,1} + M_{j,0} L_j, \quad (4.11)$$

where the entries of L_j can be individually identified as inner products of generator basis functions, see (3.3) and (3.32) in [16].

In all applications step b) of the above road map will turn out to be easy, so that, in summary, the main burden of the construction is shifted to the construction of dual generator bases. In particular, all geometric information can be incorporated in this step. The identification of the corresponding wavelets reduces to an *evaluation* of a relation like (4.6).

4.2 1D-Examples

As a first simple example we consider the second order boundary value problem

$$\begin{aligned} -\frac{d^2 u}{dx^2} &= f \quad \text{on } \Omega = (0, 1), \\ u(0) = u(1) &= 0. \end{aligned} \quad (4.12)$$

We have tested this example for two different right-hand sides. In the first case, f is designed such that the exact solution is

$$u(x) = e^{-100(x-0.5)^2} \quad (4.13)$$

which, up to the numerical precision, indeed satisfies the Dirichlet boundary conditions. In the second case, we choose f corresponding to the solution

$$u(x) = 4 \frac{e^{ax} - 1}{e^a - 1} \left(1 - \frac{e^{ax} - 1}{e^a - 1} \right), \quad (4.14)$$

which also satisfies the boundary conditions. For our tests, we choose $a = 5.0$ (although other values for a are clearly possible). According to (2.6) we will be concerned with the weak formulation

$$\langle u', v' \rangle = \langle f, v \rangle, \quad \text{for all } v \in H_0^1(0, 1). \quad (4.15)$$

In the light of the discussion in Section 2.9 the following points should be kept in mind when interpreting the numerical results to be reported on later. The solution has in this case arbitrarily high pointwise smoothness and therefore has arbitrarily high Sobolev regularity. Therefore one might conclude from the remarks in Section 2.9 at the first glance that adaptive schemes may not work well in this. However, as also explained in Section 2.9, adaptivity may pay off for functions having a very large Sobolev norm while the relevant Besov norm is of moderate size. This present examples are expected to be of this type. Due to the strong gradients of the solutions u , their Sobolev norms $\|u\|_{H^r}$ increase dramatically as r grows. As we shall see later on, at least for the second example the corresponding Besov norms turn out to be indeed significantly smaller. Some quantitative estimates comparing Besov and Sobolev norms will be given in Section 4.2.2.

4.2.1 Wavelets on the Interval

Let us consider the interval $\Omega = (0, 1)$. The construction of wavelets on the interval is meanwhile well understood, see, e.g., [1, 13, 28]. Here we refer to the construction in [28]. According to the above comments we mainly have to explain the construction of suitable dual pairs of generator bases.

The common strategy is to start with a biorthogonal multiresolution analysis on \mathbb{R} . Specifically we choose here a biorthogonal system from the family constructed in [17] where the primal scaling functions consist of cardinal B-splines. For $j \geq j_0$ where j_0 is fixed (sufficiently large to disentangle end point effects) one builds Φ_j by keeping those translates $2^{j/2} \phi(2^j \cdot -k)$, $k \in \mathbb{Z}$, that are fully supported in $[0, 1]$. These will be referred to as *interior basis functions*. For B-splines of order m one adds at each end of the interval m fixed linear combinations of the $2^{j/2} \phi(2^j \cdot -k)$ in such a way that the resulting collection Φ_j spans all polynomials of order m on $(0, 1)$. One proceeds in the same way with the dual scaling functions restoring the original order of polynomial exactness while keeping $\#\Phi_j = \#\tilde{\Phi}_j$. At this point one can verify (4.2), (4.3) and (4.5), i.e., nestedness, locality and refinability. However, only the interior basis functions inherit the biorthogonality from the line whereas the boundary modifications have perturbed biorthogonality. One can show though that in this spline family of dual multiresolution sequences one can *always* biorthogonalize [28] ending up with pairs of generator bases $\Phi_j, \tilde{\Phi}_j$ satisfying all the properties mentioned in Section 4.1. In fact, there are additional noteworthy features which we record now for later use.

- (i) Exploiting symmetry properties of the original scaling functions one can arrange the bases to be invariant under the transformation $x \rightarrow 1 - x$, a fact that will be very useful in the bivariate case.
- (ii) One can arrange that only a single primal and dual basis function differs from zero at the end points of the interval.
- (iii) According to the above comments, the bases $\Phi_j, \tilde{\Phi}_j$ always consist of three parts signified by the index sets $\Delta_j^L, \Delta_j^I, \Delta_j^R$ (and similarly for the dual collections) identifying the left boundary, interior and right boundary basis functions. Only the size of the interior sets Δ_j^I depends on j . The number of boundary functions stays always the same. Moreover, for each end point one has a fixed finite number (namely m for the primal respectively \tilde{m} for the dual side) of scaling relations which can be computed a-priorily and stored. The interior basis functions satisfy, of course, the classical stationary refinement rule from the line case.

The following properties of the corresponding *refinement matrices* $\mathbf{M}_{j,0}$ and $\tilde{\mathbf{M}}_{j,0}$ from (4.5) can be inferred from the above facts.

- 1) It follows from (i) that the matrices $\mathbf{M}_{j,0}, \tilde{\mathbf{M}}_{j,0}$ are invariant under reversing the order of rows and columns.
- 2) The refinement matrices have fixed upper left and lower right blocks with the above symmetry properties. Only the stationary interior block changes its size with growing level j , see Figure 1.

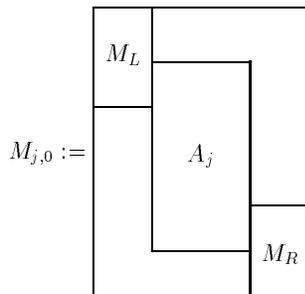


Figure 1: Structure of refinement matrices for spline wavelets on the interval.

A detailed description of the software for constructing wavelets on the interval can be found in [8]. It is based on the *Multilevel Library* presented in [6] and [7].

For the treatment of problem (4.15) the wavelet basis has to satisfy (2.8) at least for $H = H^1(0, 1)$. Therefore we choose the cardinal B-splines and their duals both of order $m, \tilde{m} \geq 2$,

$$\varphi := N_m(x), \quad \tilde{\varphi} := \tilde{N}_{m,\tilde{m}}(x), \quad (4.16)$$

as the starting point of the construction. Recall from (4.15) that the trial functions have to satisfy homogeneous Dirichlet boundary conditions. There are two ways of incorporating such boundary conditions that suggest themselves.

- (I) In view of property (ii) above one can simply remove those basis functions from the generator bases $\Phi_j, \tilde{\Phi}_j$ that do not vanish at the end points of the interval. Obviously, the resulting collections are still biorthogonal and span appropriate subspaces of $H_0^1(0, 1)$, see [19, 15, 31].
- (II) Following [31], remove the two end point boundary functions from Φ_j that do not vanish at 0 and 1 but discard two *interior* basis functions from the dual collection $\tilde{\Phi}_j$. It can be shown that the resulting sets can again be biorthogonalized retaining all the above properties [33]. Note that now only $\Phi_j \subset H_0^1(0, 1)$ while $S(\tilde{\Phi}_j)$ still contains all polynomials of order \tilde{m} on the *whole* interval $(0, 1)$.

To construct now wavelets for either choice one can apply the recipe from Section 4.1. In fact, one easily identifies an *initial stable completion* that works in both cases simultaneously. One can just take as a complement basis the hat functions corresponding to new knots on the next refinement level. This is often referred to as hierarchical basis [51]. The matrices $\check{\mathbf{M}}_{j,1}$ and the inverses $\check{\mathbf{G}}_j$ are very sparse and explicitly given. The point is the following

Property B: The initial complement basis functions $\check{\psi}_{j,k}$ vanish at the end points of $(0, 1)$.

Biorthogonal wavelets are then obtained for (I) and (II) by (4.6) in Proposition 4.1. Again the realization of boundary conditions is completely reduced to the construction of generator bases only.

Remark 4.2 *In both cases (I) and (II) it is easy to determine the adapted refinement matrices which retain the above mentioned properties. The modification of $\mathbf{M}_{j,0}$ is simply discarding the first and last rows and columns. Thus the effect of either option on the corresponding biorthogonal wavelets is the use of the respective versions of $\check{\mathbf{M}}_{j,0}$ in (4.6).*

While (I) is most simple and immediate, the option (II) appears to be conceptually preferable for the following reasons. Quite in line with the structure of the dual $H^{-1}(0, 1)$ of $H_0^1(0, 1)$, the functionals on $H_0^1(0, 1)$ should not be constrained at the end points. This is supported by the following simple observation. Biorthogonality of the wavelets combined with the fact that in (II) the dual system retains full polynomial exactness immediately ensure that the wavelets have vanishing moments of order \tilde{m} on all of $(0, 1)$

$$\langle P, \psi_{j,k} \rangle = 0, \quad P \in \Pi_{\tilde{m}}. \quad (4.17)$$

Hence when the right-hand side is very simple, e.g., $f = 1$ on $(0, 1)$, all right hand-side data $\langle f, \psi_{j,k} \rangle$ arising in the Galerkin scheme except on the scaling function level j_0 vanish in the case (II) while infinitely many wavelet coefficients build up in the case (I) even if f is very smooth near the end points. For a detailed discussion of related effects we refer to [24]. Therefore we used (II) in the present tests.

Remark 4.3 *Since in the present case $H = H_0^1(0,1)$ the diagonal matrix in (2.8) can be chosen as $\omega_\lambda := 2^{|\lambda|}$. We have to verify norm equivalences of the form (2.8) in the presence of boundary conditions, but the results in [33] e.g. imply that the wavelet bases for either choice (I) or (II) satisfy the norm equivalence (2.8) for \mathbf{D} as above. Moreover, for (II) the cancellation properties (2.13) holds for $\tilde{m} = 2$ throughout $(0,1)$. The fact that these cancellation properties deteriorate somewhat near the end points for option (I) should, however, not destroy the overall compressibility property (2.22).*

Moreover, the following fact established in [31] will be useful.

Remark 4.4 *The wavelet bases can be arranged to share the symmetry properties (i) of the generator bases.*

One can see from (4.6) that the two scale matrices $\mathbf{M}_{j,1}$, $\tilde{\mathbf{M}}_{j,1}$ in the definition of the biorthogonal wavelets have the same principal 3-block structure as the corresponding refinement matrices shown in Figure 1.

In Figure 2, we have depicted two members of the resulting set of scaling functions and wavelets, respectively, one in the interior and one which intersects the boundary.

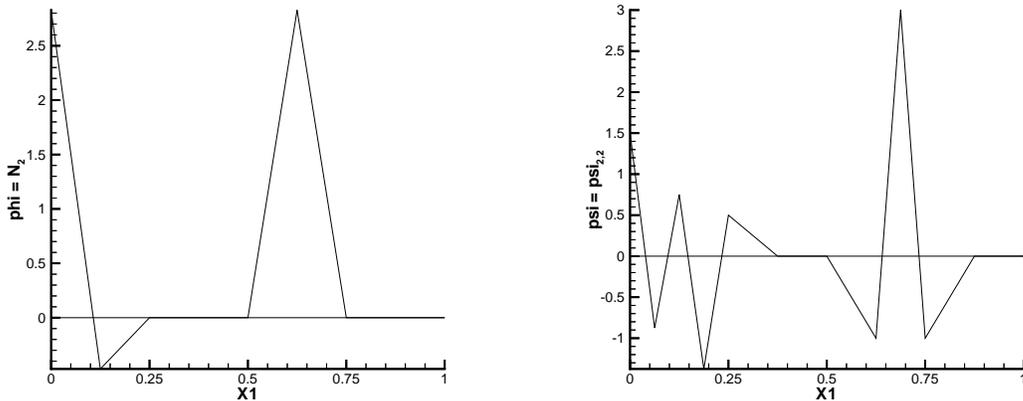


Figure 2: An interior and the at the border non vanishing function, for scaling functions and wavelets.

4.2.2 Discussion of Results

To be able to interpret the results, we have to identify first the *range of asymptotic optimality* permitted by the above choice of bases. Theorem 2.5 implies that the asymptotic behavior depends on the compressibility of the matrix \mathbf{A} . In particular, the parameter s^* defined in (2.21) would provide a range where optimality is guaranteed. However, in the present situation this criterion is too weak. In fact, we know from [24] that the parameter

σ in (2.19) must satisfy $t + \sigma < \gamma$ where γ bounds the Sobolev regularity of the wavelets. In this case $t = 1$, $\gamma = m - 1/2$ which gives $\sigma = m - 3/2$, i.e., $s^* = m - 2$. Therefore, in the case $\varphi = N_2$, we end up with $s^* = 0$ which is clearly useless. Of course, the condition (2.21) is only sufficient and in this case a more detailed analysis of the compression properties is necessary. In fact, for our special case, a sharper result can be shown.

Lemma 4.5 *Let \mathbf{A} denote the stiffness matrix to (4.12) obtained by B -spline wavelets of order m as basis functions. Then for any $\epsilon > 0$ the following compression estimate holds:*

$$\|\mathbf{A} - \mathbf{A}_J\| \lesssim 2^{-J(m-3/2-\epsilon)}, \quad \text{i.e.,} \quad \mathbf{A} \in \mathcal{A}_s \text{ for all } s < m - 3/2. \quad (4.18)$$

Proof: Eq. (4.18) can be established directly by using a version of the *Schur lemma*: if for the matrix $\mathbf{B} = (b_{\lambda,\lambda'})_{\lambda,\lambda' \in \mathcal{J}}$ there is a sequence ω_λ , $\lambda \in \mathcal{J}$ and a positive constant c such that

$$\sum_{\lambda' \in \mathcal{J}} |b_{\lambda,\lambda'}| \omega_{\lambda'} \leq c \omega_\lambda \quad \text{and} \quad \sum_{\lambda \in \mathcal{J}} |b_{\lambda,\lambda'}| \omega_\lambda \leq c \omega_{\lambda'}, \quad \lambda, \lambda' \in \mathcal{J}, \quad (4.19)$$

then $\|\mathbf{B}\| \leq c$. We want to use (4.19) for the sequence $\omega_\lambda = 1$ for all $\lambda \in \mathcal{J}$. Let us briefly sketch the arguments. The first step is to estimate the entries in the stiffness matrix corresponding to (4.15). Ignoring for the moment the boundary effects, recalling that derivatives of wavelets are again wavelets, see e.g. [27], and using the vanishing moment property of wavelets, we obtain for any polynomial $P_{\lambda'}$ on $\Omega_{\lambda'}$ of degree $< m - 1$ and $j' \geq j$

$$\begin{aligned} \left\langle \frac{d\psi_\lambda}{dx}, \frac{d\psi_{\lambda'}}{dx} \right\rangle &= \left\langle \frac{d\psi_\lambda}{dx} - P_{\lambda'}, \frac{d\psi_{\lambda'}}{dx} \right\rangle \\ &\leq \left\| \frac{d\psi_\lambda}{dx} - P_{\lambda'} \right\|_{L_2(\Omega_{\lambda'})} 2^{j'} \left\| \frac{d\psi_{\lambda'}}{dx} \right\|_{L_2} \\ &\lesssim \left\| \frac{d\psi_\lambda}{dx} - P_{\lambda'} \right\|_{L_2(\Omega_{\lambda'})}. \end{aligned}$$

Since $\frac{d}{dx}\psi_\lambda \in H^s$, $s < m - 3/2$, a classical Whitney type estimate yields therefore

$$\begin{aligned} \left\langle \frac{d\psi_\lambda}{dx}, \frac{d\psi_{\lambda'}}{dx} \right\rangle &\lesssim 2^{j'} 2^{-j'(m-3/2-\epsilon)} \left| \frac{d\psi_\lambda}{dx} \right|_{H^{m-3/2-\epsilon}} \\ &\lesssim 2^{j'} 2^{-j'(m-3/2-\epsilon)} |\psi_\lambda|_{H^{m-1/2-\epsilon}} \\ &\lesssim 2^{j'} 2^{-j'(m-3/2-\epsilon)} 2^{j(m-1/2-\epsilon)} \\ &\lesssim 2^{(j-j')(m-3/2-\epsilon)} 2^{j+j'}, \end{aligned}$$

so that, taking the preconditioning matrix \mathbf{D} into account, we get

$$|a_{\lambda,\lambda'}| \lesssim 2^{(j-j')(m-3/2-\epsilon)}, \quad j' \geq j. \quad (4.20)$$

The case $j' < j$ can be treated analogously,

$$|a_{\lambda,\lambda'}| \lesssim 2^{(j'-j)(m-3/2-\epsilon)}, \quad j' < j. \quad (4.21)$$

However, the crude estimates (4.20) and (4.21) do not tell the whole truth. Indeed, if we combine the fact that the generator is a cardinal B-spline with the vanishing moment property of the wavelet basis, we see that for fixed values of $|\lambda|, |\lambda'|$ a lot of entries $|a_{\lambda, \lambda'}|$ are zero. Roughly speaking, the non vanishing entries correspond only to the wavelets $\psi_{\lambda'}$ whose supports intersect the singular support of ψ_{λ} . It can be shown that the number of these entries does not depend on the refinement level. Consequently, we get

$$\sum_{|\lambda'|=j'} |a_{\lambda, \lambda'}| \lesssim 2^{-|j-j'|(m-3/2-\epsilon)}. \quad (4.22)$$

According to (2.23) and (4.19), we have to show that

$$\sum_{|j-j'|>J} \sum_{|\lambda'|=j'} |a_{\lambda, \lambda'}| \lesssim 2^{-J(m-3/2-\epsilon)}. \quad (4.23)$$

Let us again first consider the case $j' > j$. By using (4.22), we obtain

$$\sum_{j'-j>J} \sum_{|\lambda'|=j'} |a_{\lambda, \lambda'}| \lesssim \sum_{j'=j+J}^{\infty} 2^{(j-j')(m-3/2-\epsilon)} \lesssim 2^{j(m-3/2-\epsilon)} 2^{-(J+j)(m-3/2-\epsilon)} \lesssim 2^{-J(m-3/2-\epsilon)}. \quad (4.24)$$

The case $j' \leq j$ can be treated analogously. The second condition in (4.19) can be checked in a similar fashion and (4.18) is established. \blacksquare

In view of the pivotal role of the approximate fast matrix/vector multiplication, we begin with some tests of this ingredient. Of course, accessing efficiently the relevant matrix entries when performing the telescoping expansion (2.50) is one of the central problem dependent interfaces. In order to obtain a quick impression of the quantitative behavior of such a routine we have employed a very provisional strategy, namely to precompute a possibly large section of the infinite matrix \mathbf{A} from (2.14). This can be done by first determining a full scaling function representation of the stiffness matrix followed by a wavelet transform. We can then simply call the entries needed in (2.50). Of course, this is a preliminary step that allows us to quickly check accuracy. The error estimate (2.51) indicates that the approximation power of the fast/matrix multiplication is determined by the parameter s^* which, according to (4.18), is given by $s^* = m - 3/2$. This is confirmed by our numerical tests. In Figure 3, the error $\|\mathbf{A}\mathbf{v} - \mathbf{w}_j\|_{\ell_2}$ is plotted in a logarithmic scale. We see that the slope of the resulting curve is indeed approximately $m - 3/2$.

According to Theorem 2.5 we can expect that the error in approximating the solution u to (4.15) decays at best like $N^{-(m-3/2)}$. This is confirmed by the numerical results displayed in the figures below. The first pictures in Figure 4 and 5 show the current Galerkin approximation u_{Λ} and the corresponding error $\mathbf{u} - \mathbf{u}_{\Lambda}$ for the first example (4.13) for the wavelet basis associated with $\varphi = N_2$, $\tilde{\varphi} = \tilde{N}_{2,2}$. One can see that the error decreases very rapidly, as expected. In the following pictures in Figure 6, the sets of wavelets selected at the respective stage by the adaptive algorithm are plotted. These wavelets are sometimes called the *active* ones. We should mention, that here we have chosen as Λ_0 the set of indices of all scaling functions on the coarsest level, because this case shows some instructive effects in the later choice of indices by the algorithm. Its qualitative performance when choosing $\Lambda_0 = \emptyset$ however, is the same.

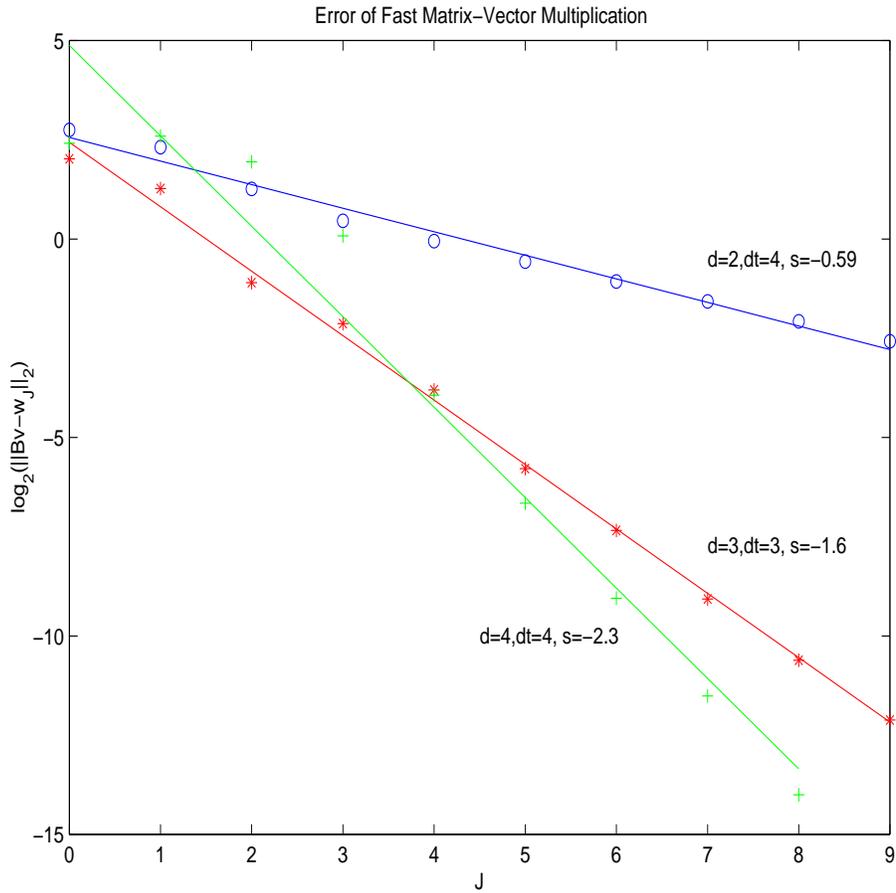


Figure 3: The slope of error reduction.

A few further details shed some light on the way the scheme works. We see that although the problem and the wavelet basis is symmetric with respect to the point $x = 0.5$, sometimes the wavelets on a given refinement level are chosen in a non symmetric way. If for example all coefficients are equal, the N largest coefficients are not uniquely determined and the best N -term approximation is realized by different possible choices, see Figure 6. Also sometimes a dyadic level is skipped when expanding the set of active wavelets. In the course of further refinements, however, symmetrization and filling of ‘gaps’ gradually takes place.

One more comment concerning the behavior of the adaptive scheme is in order. We see that the adaptive algorithm in fact observes the strong gradient of the solution u and adds wavelet coefficients in these regions. Therefore the location of the significant wavelet coefficients of the approximate solution adequately reflects the features of the right-hand side. Consequently, the active wavelets are by no means equally distributed.

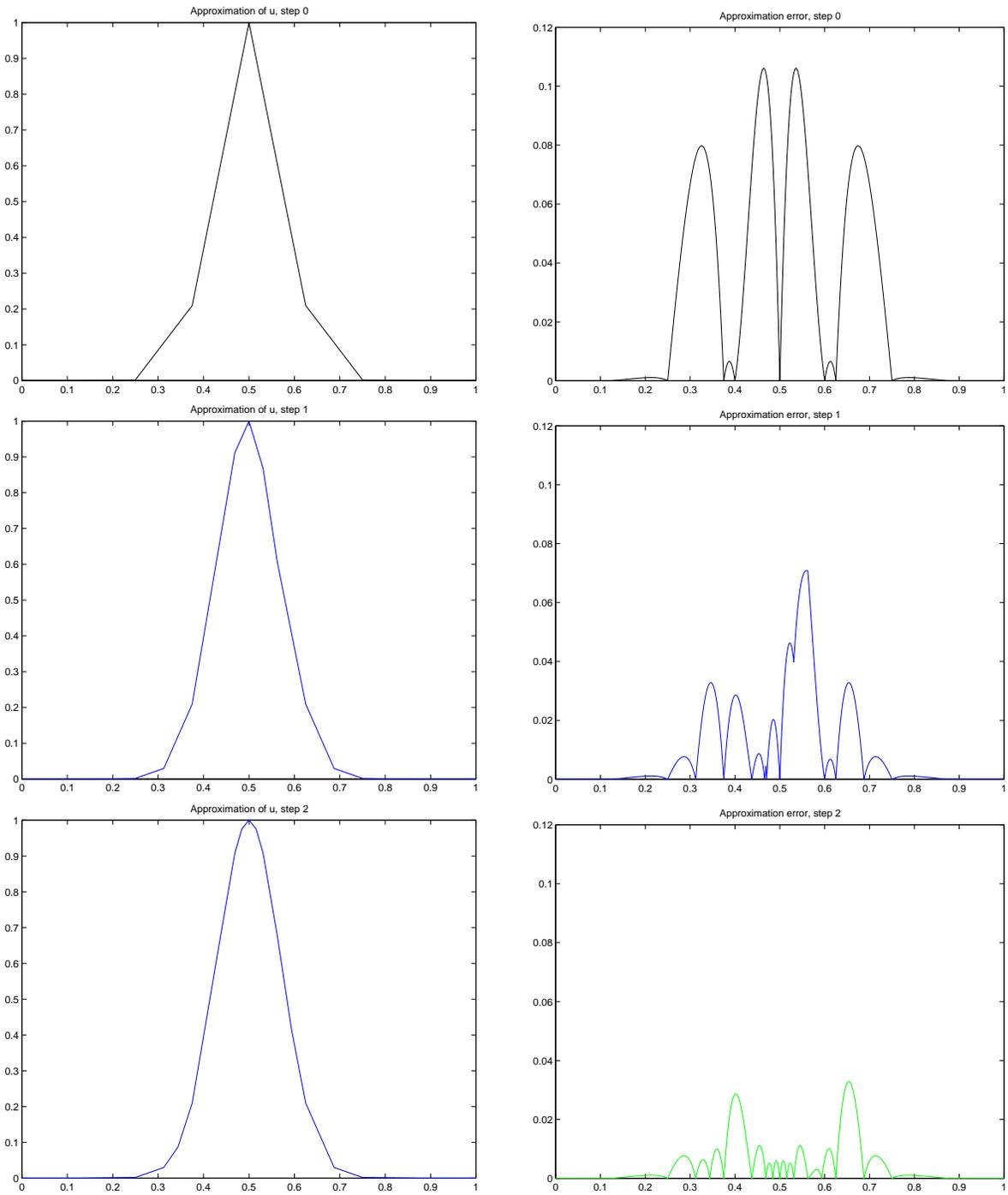


Figure 4: The first three approximate solutions and differences to exact solution.

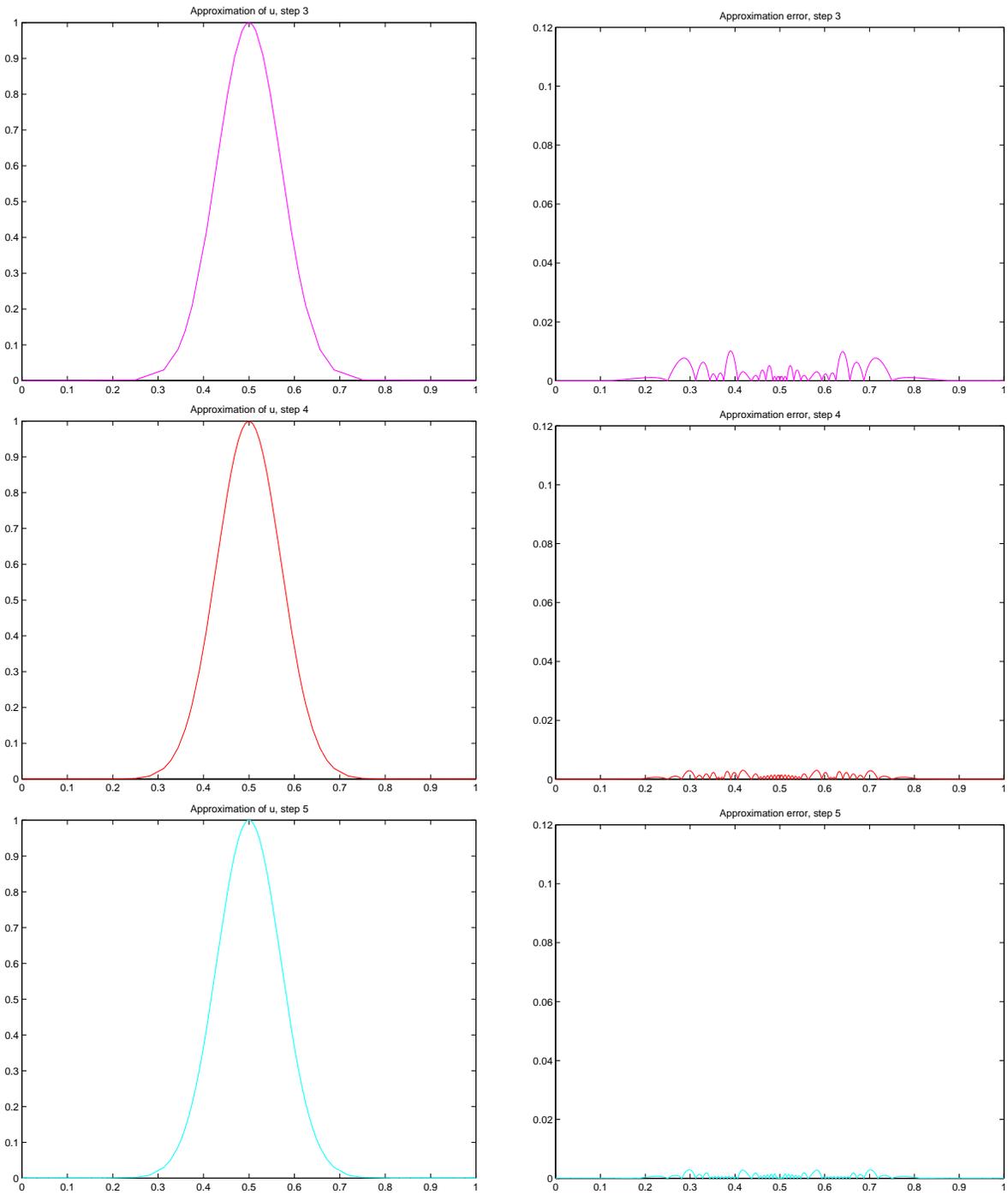


Figure 5: The next three approximate solutions and differences to exact solution.

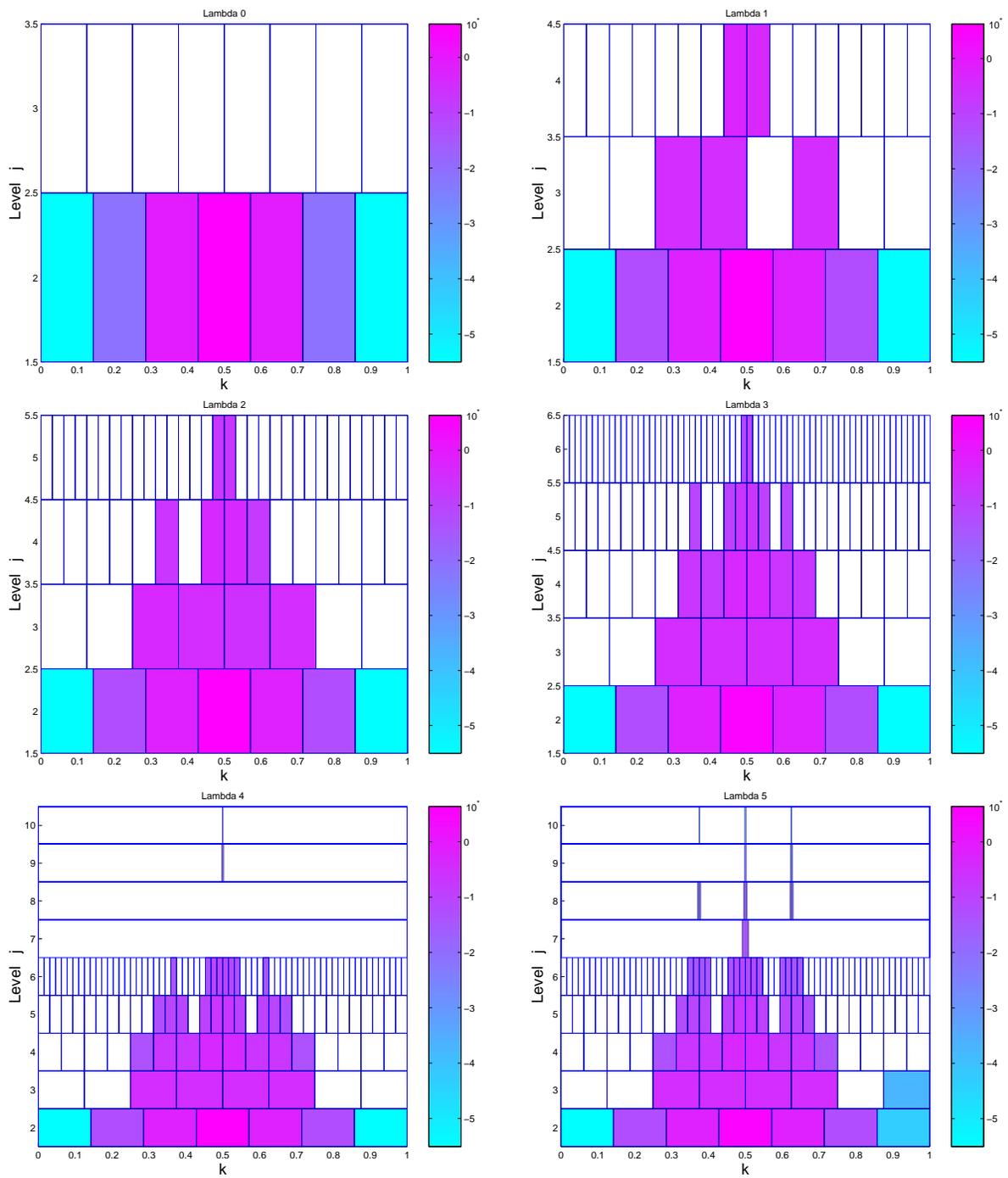


Figure 6: The sets of active indices for the first six iterations.

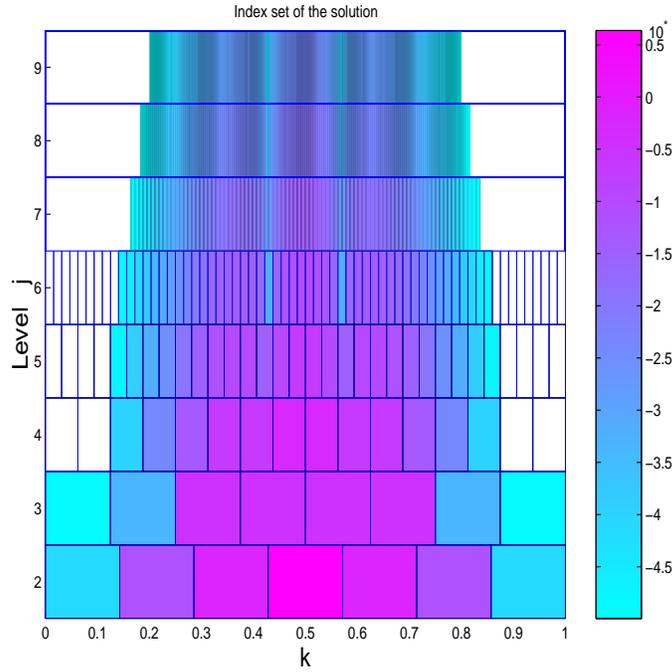


Figure 7: Index set of the exact solution

We expect that the performance of the adaptive scheme can be improved dramatically by increasing the smoothness of the wavelet basis. Indeed, since σ in (2.21) is directly related with the regularity of the basis, the parameter s^* which determines the performance of the algorithm can be made larger in this way, compare again with (4.18). At this point, we want to emphasise that in contrary to linear schemes the approximation order of the adaptive scheme is *not* given by the polynomial exactness of the multiresolution analysis. We have also made some quantitative tests with smoother wavelet bases. From Theorem 2.5 we know that for $s < s^*$ the algorithm should perform with the same order of approximation as the corresponding best N -term approximation. In Figure 8 we have depicted the error for both, the best N -term approximation (continuous line) and the adaptive algorithm (diamond-shaped line), as N increases, in a logarithmic scale. We see that both errors show almost the same behaviour. We also see that the adaptive algorithm indeed performs better for smoother wavelet bases (i.e., for larger values of s^*), as it should.

We performed similar numerical tests also for the second example. The results are depicted in the Figures 9 and 10. As before, the pictures show the current Galerkin approximation u_Λ and the corresponding error $\mathbf{u} - \mathbf{u}_\Lambda$. We used the wavelet basis associated with $\varphi = N_3$, $\tilde{\varphi} = \tilde{N}_{3,3}$. We see that again the error decreases very rapidly as expected. The reader should note that the errors are depicted in different scales!

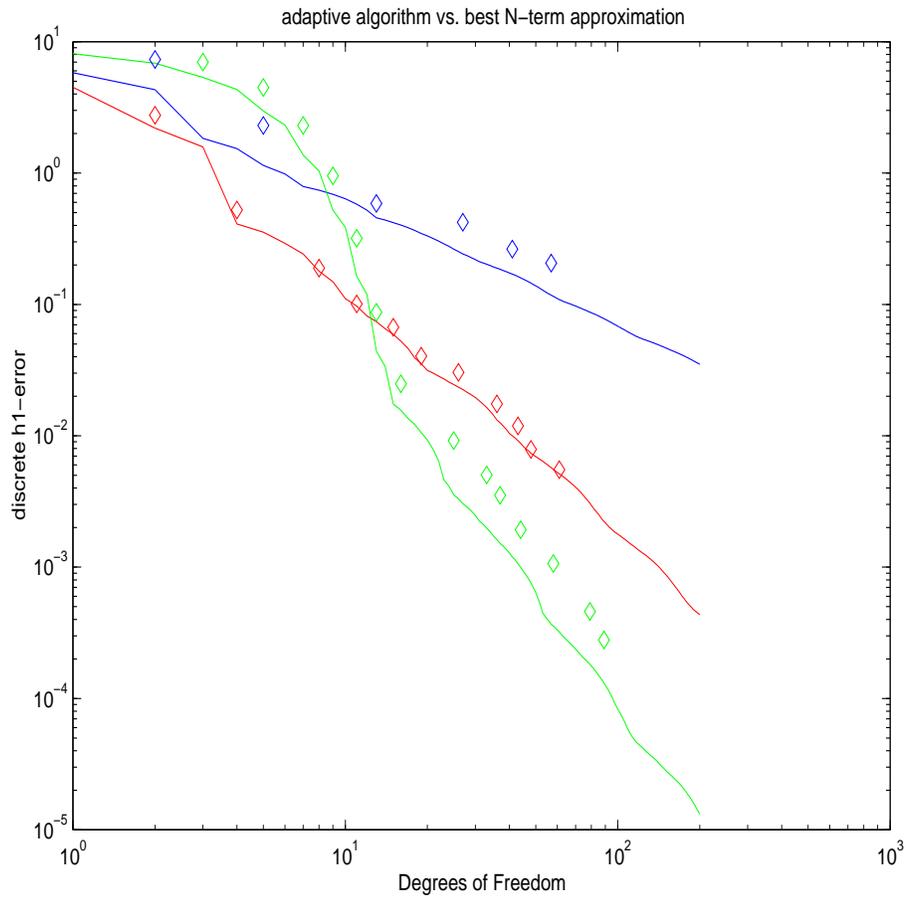


Figure 8: Error of the adaptive algorithm and the best N -term approximation .

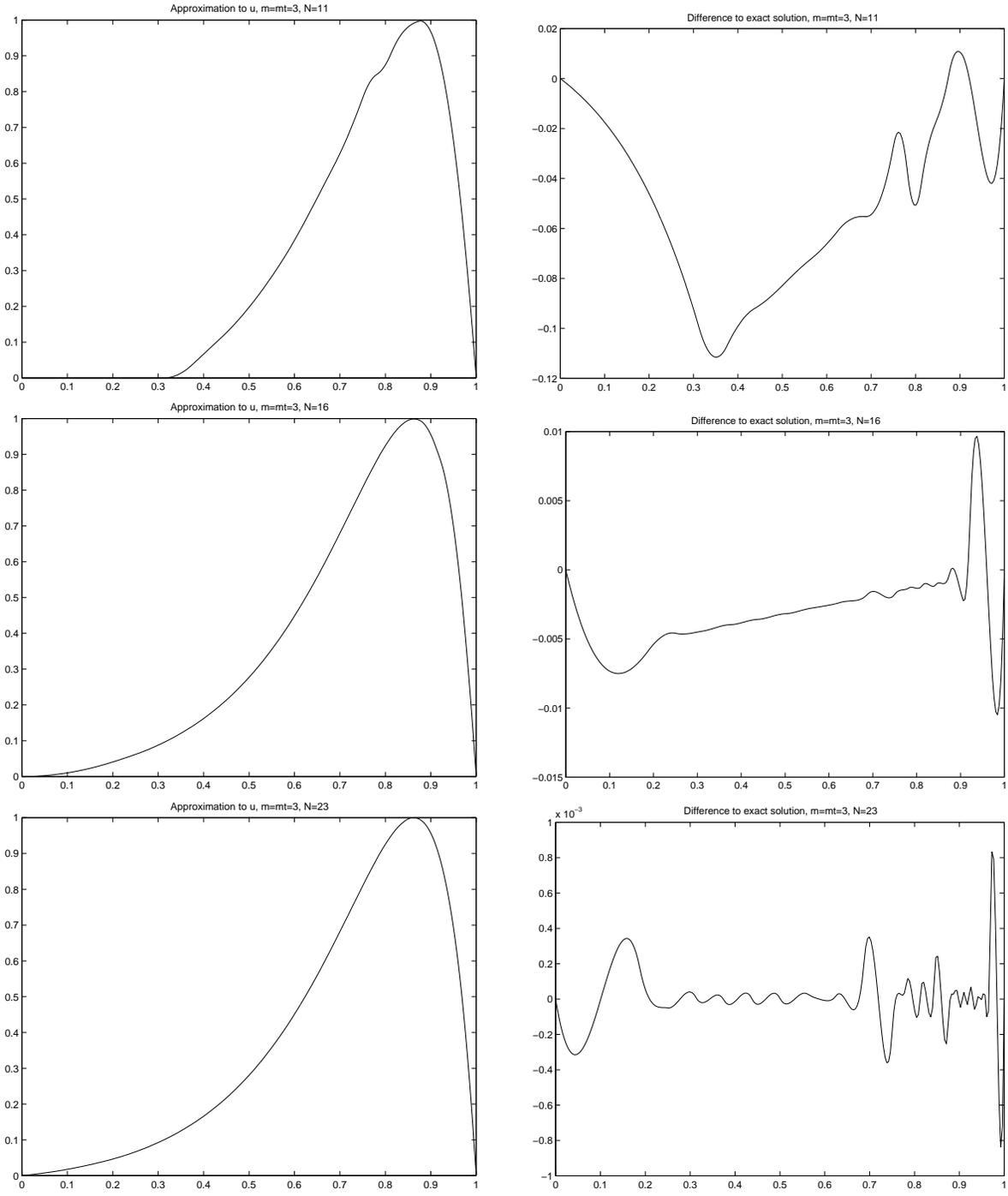


Figure 9: The first three approximate solutions and differences to exact solution, second example

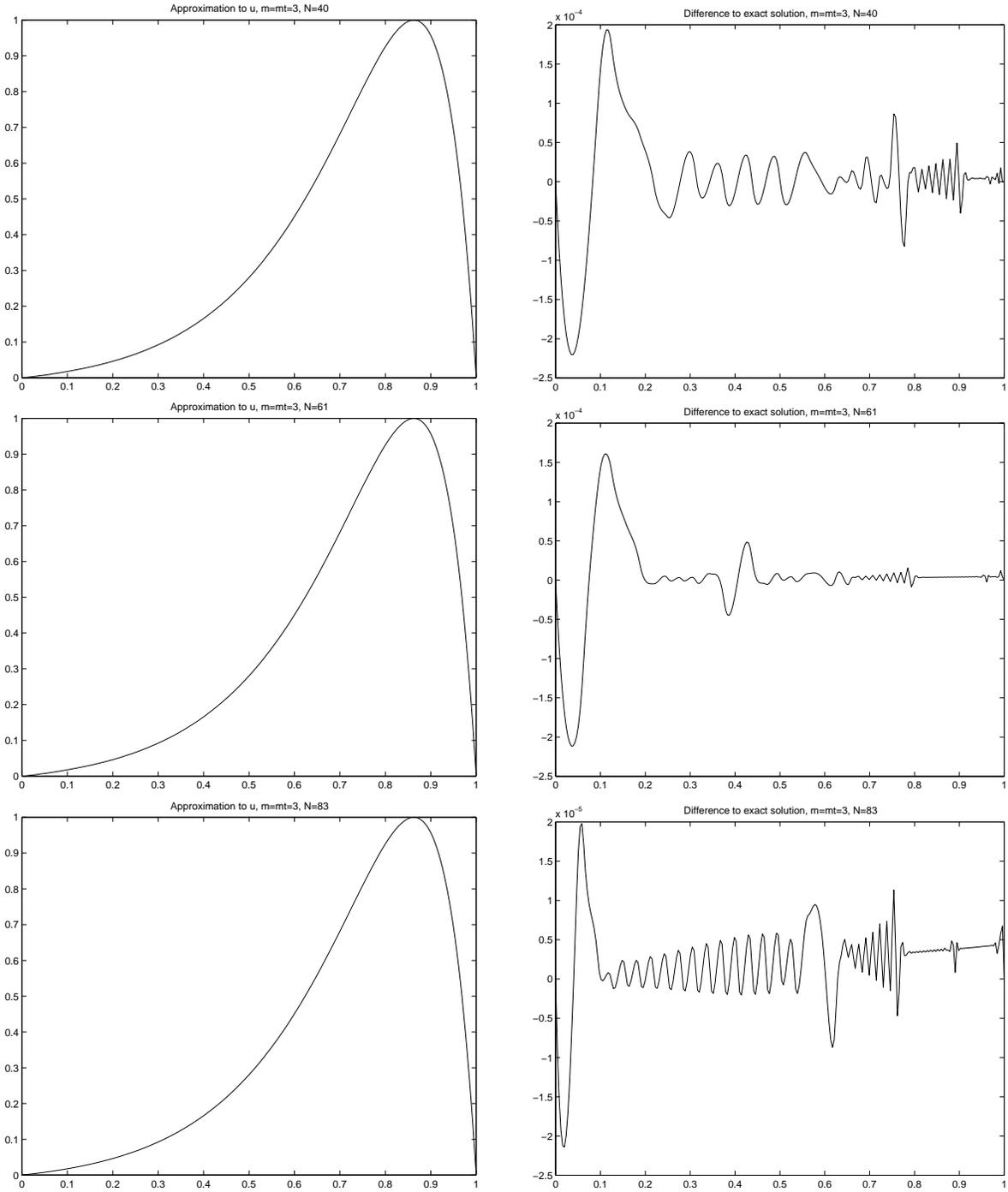


Figure 10: The next three approximate solutions and differences to exact solution, second example

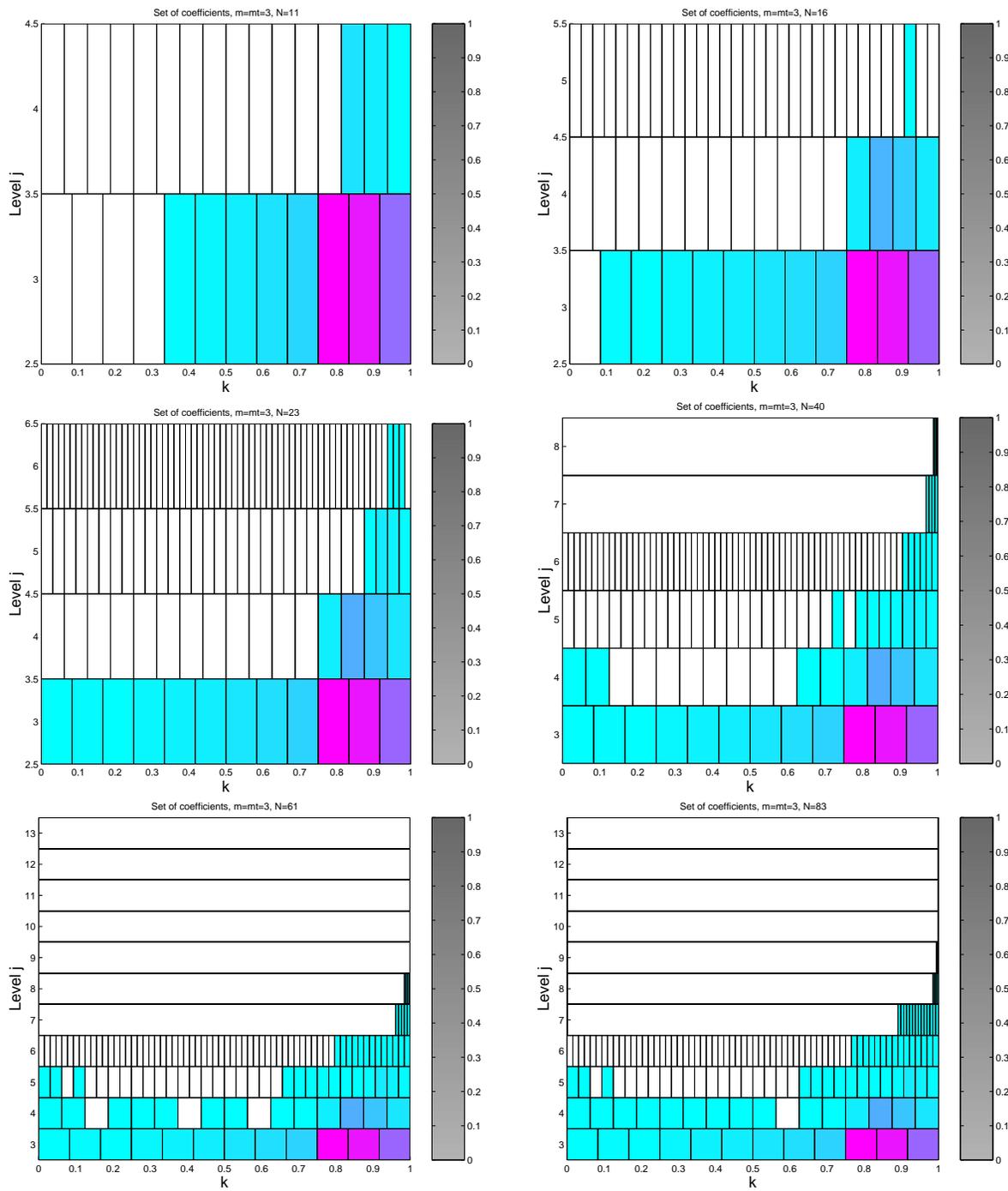


Figure 11: The sets of active indices for the first six iterations.

In Figure 12, the performance of the adaptive algorithm compared with the best N -term approximation is illustrated. Again both algorithms show almost the same behavior. We also observe that the performance of the algorithm gets better as the smoothness of the wavelets increases.

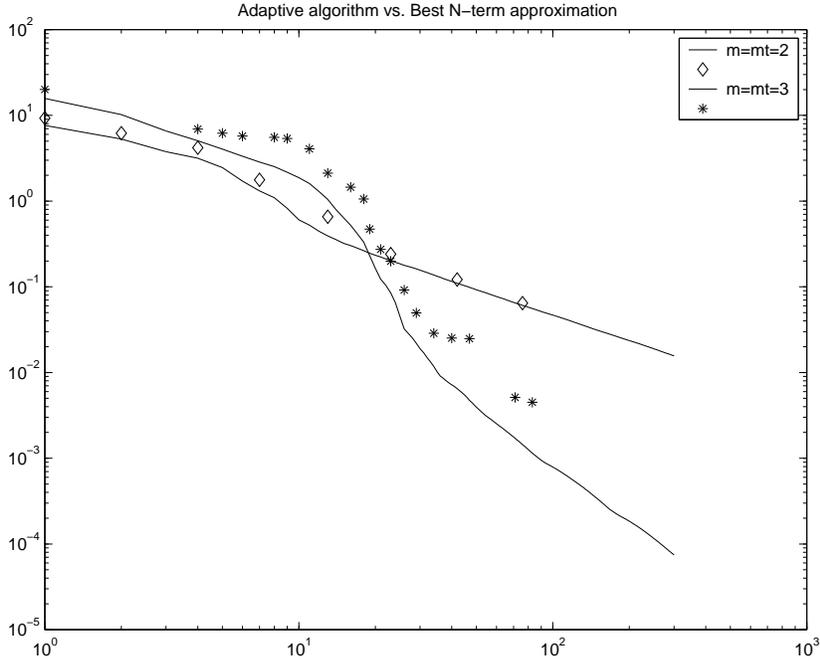


Figure 12: Comparison between best N -term approximation and adaptive algorithm.

Let us end this section with some remarks concerning the comparison of adaptive and nonadaptive schemes. As we have already seen, for both examples studied above, the wavelet coefficients chosen by the adaptive algorithm are by no means equally distributed. This indicates that for these example we indeed gain efficiency by adaptive schemes although the solutions are arbitrary smooth in the Sobolev scale. As already stated above, the order of approximation that can be achieved is limited by the parameter s^* , and the best we can expect in our case is an estimate of the form

$$\|u - u_{\Lambda_j}\| \lesssim (\#\Lambda_j)^{-(m-3/2)}.$$

For a uniform refinement scheme we obtain the same order of approximation for functions in $H^{m-1/2}(0,1)$, i.e.,

$$\inf_{v_j \in S_j} \|v - v_j\|_{H^1(\Omega)} \lesssim N_j^{-(m-3/2)} \|v\|_{H^{m-1/2}(\Omega)}, \quad (4.25)$$

compare with (2.37). We therefore gain efficiency if the $H^{m-1/2}$ -norm of the solution u is large when compared with the corresponding norm in the Besov space $B_{\tau^*}^{m-1/2}(L_{\tau^*}(\Omega))$, $1/\tau^* = m - 3/2 + 1/2 = m - 1$. For the first example, it turns out that these two norms are only slightly different, whereas in the second example, they differ dramatically.

The norms of the latter case are depicted in the following figure:

m	$H^{m-1/2}$	$B_{(m-1)^{-1}}^{(m-1)}(L_{(m-1)^{-1}})$
1.5	6.73	6.73
2	39.4	14.5
2.5	240	47.8
3	1617	275

We have estimated the norms by employing the norm equivalences (2.11), i.e., by computing weighted sequence norms of wavelet expansions. The difference between Sobolev and Besov norms gets plausible if we take a look at the right-hand side and the exact solution for the second example, see Figure 13:

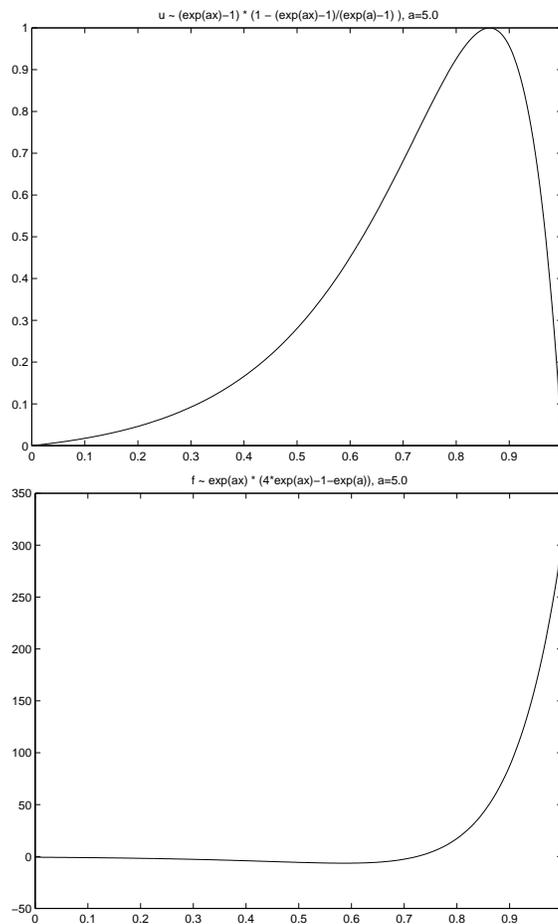


Figure 13: The exact solution and the right-hand side for the second example

In contrary to the first example, the right-hand side of the second example does not satisfy the Dirichlet boundary conditions. This causes the boundary layer of the solution

u . This layer increases the Sobolev norm but does not influence the (weaker) Besov norm too much.

The above comparison of Sobolev and Besov norms indicates that there is indeed some gain of efficiency for adaptive schemes possible for this example. Therefore we made some numerical tests in which we compared our adaptive algorithm with a uniform scheme. The result is depicted in Figure 14.

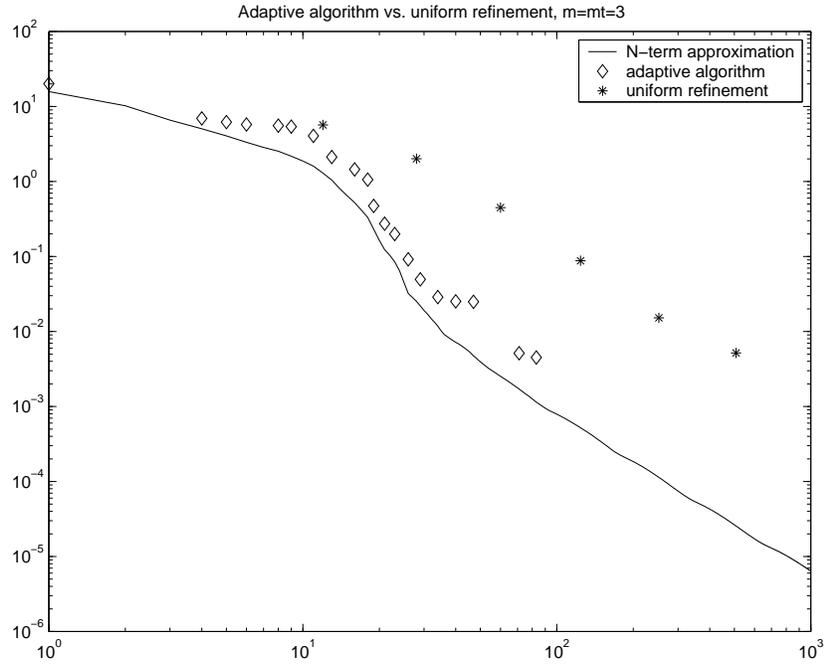


Figure 14: Comparison between adaptive algorithm / uniform refinement.

As one would expect, the slope of the curves is indeed the same. Nevertheless, since the norms differ so much, the line corresponding to the linear scheme lies high above the one corresponding to the adaptive scheme. Therefore we have indeed a spectacular gain of efficiency.

4.3 2D-Examples

Our next test case is the classical Poisson equation on an L -shaped domain Ω in \mathbb{R}^2 ,

$$\begin{aligned} -\Delta u &= f \quad \text{on } \Omega, \\ u|_{\partial\Omega} &= 0, \end{aligned} \tag{4.26}$$

as shown in Figure 15.

This problem is interesting because now the solution may exhibit singularities solely

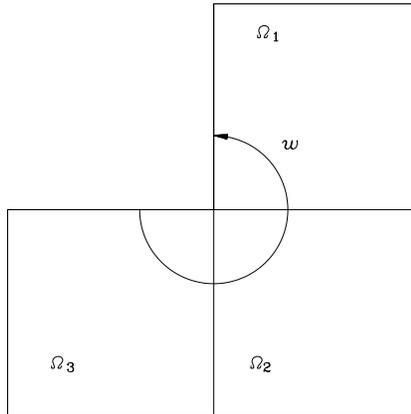


Figure 15: The L -shaped domain.

caused by the shape of the domain even for smooth right hand sides. In order to be able to validate later the numerical results we start by briefly recalling some basic facts on the regularity theory of (4.26).

4.3.1 Regularity Theory for Polygonal Domains

It is well-known that the solution u to (4.26) on a polygonal domain may exhibit singularities in the vicinity of the vertices, especially when the domain is not convex. Therefore the Sobolev regularity of the solution will not be very high, even if the right-hand side f is smooth. According to the discussion in Section 2.9 this is a situation when adaptive schemes are potentially superior to preset discretizations. In fact, since Ω is obviously a Lipschitz domain, Theorem 2.7 tells us that the regularity of u in the specific Besov scale $B_{\tau^*}^\alpha(L_{\tau^*}(\Omega))$, $1/\tau^* = (\alpha - 1)/2 + 1/2$, will in general be higher than in the Sobolev scale. Moreover, for this specific model problem, much more can be said. In fact, for polygonal domains in \mathbb{R}^2 , the singular parts of the solutions which are responsible for the decreasing Sobolev regularity can be classified. The first results in this direction were proved by Kondrat'ev [43], see also [41] and [35].

Let Ω be a simply connected polygonal domain in \mathbb{R}^2 . The segments of $\partial\Omega$ are denoted by $\bar{\Gamma}_l$, Γ_l open, $l = 1, \dots, N$, numbered in positive orientation. Furthermore, \mathcal{V}_l denotes the endpoint of Γ_l and ω_l denotes the measure of the interior angle at \mathcal{V}_l . Moreover, we introduce polar coordinates (r_l, θ_l) in the vicinity of each vertex \mathcal{V}_l . Finally, ζ_l denotes a suitable C^∞ truncation function. The following theorem is quoted from [41].

Theorem 4.6 *Suppose that the right-hand side f in (4.26) is contained in $H^\mu(\Omega)$ for some $\mu \geq -1$. Furthermore, let us assume that $m\pi/\omega_l \neq \mu+1$ for all $l = 1, \dots, N$, $m \geq 1$. Then the solution u to (4.26) has an expansion $u = u_R + u_S$, where $u_R \in H^{\mu+2}(\Omega)$ and*

$$u_S = \sum_{j=1}^N \sum_{0 < \lambda_{l,m} < \mu+1} c_{l,m} \mathcal{S}_{l,m}, \quad \lambda_{l,m} := m\pi/\omega_l, \quad (4.27)$$

where the functions $\mathcal{S}_{l,m}$ are given by

$$\mathcal{S}_{l,m}(r_l, \theta_l) = \zeta_l(r_l) r_l^{\lambda_{l,m}} \sin(m\pi\theta_l/\omega_l), \quad \text{when } \lambda_{l,m} \text{ is not an integer}, \quad (4.28)$$

$$\mathcal{S}_{l,m}(r_l, \theta_l) = \zeta_l(r_l) r_l^{\lambda_{l,m}} [\log r_l \sin(m\pi\theta_l/\omega_l) + \theta_l \cos(m\pi\theta_l/\omega_l)] \quad \text{otherwise.} \quad (4.29)$$

We see that the solution can be decomposed into two parts. The regular part u_R only depends on the right-hand side and can be made arbitrarily smooth by increasing the smoothness of f . By using suitable embeddings, it turns out that $u_R \in B_{\tau^*}^\alpha(L_{\tau^*}(\Omega))$, $1/\tau^* = (\alpha - 1)/2 + 1/2$, $\alpha < \mu + 2$. On the other hand, the singular part u_S does not depend on f but describes the influence of the domain, and we see that the order of convergence that can be achieved by an adaptive scheme essentially depends on the Besov regularity of u_S . From Theorem 2.7 we already know that this regularity is high enough to justify adaptive schemes. Nevertheless, for our special case, a much sharper result is available. Quite surprisingly, it turns out that u_S has arbitrarily high smoothness in the nonlinear approximation scale of Besov spaces [22].

Theorem 4.7 *Any function $\mathcal{S}_{l,m}$ defined by (4.28) satisfies*

$$\mathcal{S}_{l,m} \in B_{\tau^*}^\alpha(L_{\tau^*}(\Omega)), \quad \text{for all } \alpha > 0, \quad \frac{1}{\tau^*} = \left(\frac{\alpha - 1}{2} + \frac{1}{2} \right). \quad (4.30)$$

By combining Theorem 4.6 and Theorem 4.7 we therefore obtain for $f \in H^\mu(\Omega)$

$$u \in B_{\tau^*}^\alpha(L_{\tau^*}(\Omega)), \quad 0 < \alpha < \mu + 2, \quad \frac{1}{\tau^*} = \left(\frac{\alpha - 1}{2} + \frac{1}{2} \right), \quad (4.31)$$

see again [22] for details. The relations in (4.31) imply that for sufficiently smooth right-hand sides adaptive schemes can in principle perform with an arbitrarily high order of convergence – which again calls for high order wavelets!

Motivated by these observations, in our tests the right-hand side f is designed in such a way that the solution u is exactly the ‘worst’ singularity function which, in the case of the L-shaped domain, is obtained by inserting $m = 1$, $\omega = 3\pi/2$ into (4.28), i.e.,

$$u = \mathcal{S}_{1,1}(r, \theta) = \zeta(r) r^{2/3} \sin(2\theta/3). \quad (4.32)$$

It remains to fix the truncation function ζ . We set

$$w(r) := \begin{cases} e^{-1/r^2} & \text{if } r > 0, \\ 0 & \text{else,} \end{cases}$$

and define

$$\zeta(r) := \frac{w(3/4 - r)}{w(r - 1/2) + w(3/4 - r)}.$$

The resulting singularity function is depicted in Figure 16.

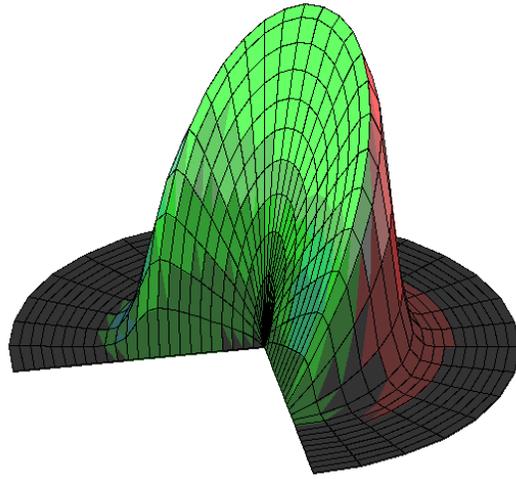


Figure 16: The solution to our model problem.

The corresponding right-hand side is constructed by applying the Laplacian to u . Observe that the function u is harmonic in the vicinity of the critical vertex. Therefore the right-hand side does not ‘see’ the strong gradient of u near this vertex which confirms that the singularities of u are in fact partially generated by the shape of the domain, see Figure 17. To our knowledge so far adaptive wavelet schemes have not been applied yet to situations of this type.

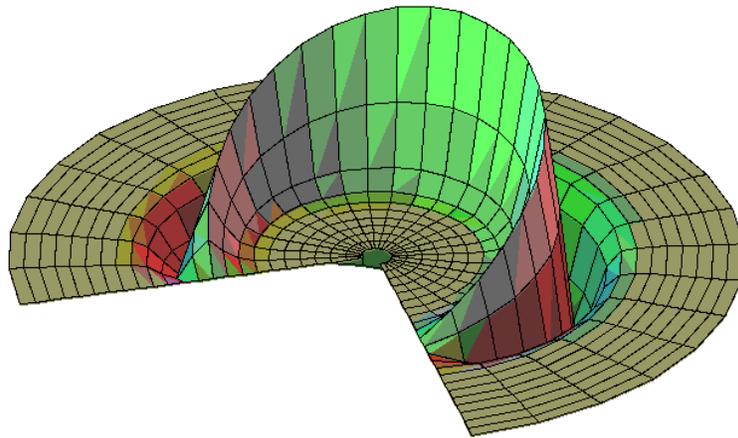


Figure 17: The right-hand side.

4.3.2 The Composite Wavelet Basis

The problem dependent part of the implementation requires again choosing suitable wavelet bases. In principle, several constructions are meanwhile available [19, 14, 31, 32] which qualify for the case at hand. Here we use the so-called *composite wavelet basis* from [31] because of a few technical conveniences.

It is clear that tensor products of wavelets on the interval yield wavelet bases on $\square := (0, 1)^d$. In our situation, the domain is a union of cubes which fits into the following framework.

A typical way to construct wavelets on more complicated domains is to use a domain decomposition technique: the domain Ω of interest is divided into non overlapping subdomains Ω_i

$$\bar{\Omega} = \bigcup_{i=1}^N \bar{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset, i \neq j. \quad (4.33)$$

Here each Ω_i is a smooth parametric image $\Omega_i = \kappa_i(\square)$ of the unit cube where in general for $d \leq d'$

$$\kappa_i : \mathbb{R}^d \rightarrow \mathbb{R}^{d'} : \square \rightarrow \Omega_i.$$

Moreover, suppose that $\Gamma_{i,l} := \bar{\Omega}_i \cap \bar{\Omega}_l$ is the common interface of Ω_i and Ω_l . For the later construction of global wavelet bases on Ω the parametric mappings κ_i have to satisfy the continuity conditions

$$\kappa_i^{-1}(\Gamma_{i,l}) = \rho(\kappa_l^{-1}(\Gamma_{i,l})), \quad (4.34)$$

where ρ is a rotation.

In order to construct wavelet bases on Ω we follow the recipe from Section 4.1, i.e., we have to construct first dual pairs of biorthogonal generator bases on the composite domain Ω . This in turn is fairly easy by stitching together parametric liftings of generator bases on the unit cube. This is essentially due to two facts. Firstly, the boundary properties (ii) of the univariate ingredients in Section 4.2.1 confine the gluing process to very few functions associated with the domain. Secondly, the symmetry properties from Remark 4.4 and (i) in Section 4.2.1 leaves convenient flexibility concerning invariance under rotations in the parametric mappings κ_i . We will detail this a bit by the following remarks.

To avoid confusion we will use the superscript \square to denote functions on \square . As mentioned before, on the reference domain we use *tensor products* of the wavelets and scaling functions constructed in Section 4.2.1.

We use multiindices $k = (k_1, \dots, k_d)$ to describe the scaling functions

$$\varphi_{j,k}^{\square} := \varphi_{j,k_1} \otimes \dots \otimes \varphi_{j,k_d}, \quad k \in \mathbf{I}_j := I_1 \times \dots \times I_d, \quad (4.35)$$

on \square .

The next task is to assemble dual pairs of generator bases on the domain Ω by composing those on the subdomains. Defining

$$\varphi_{j,k}^i(x) := \varphi_{j,k}^{\square}(\kappa_i^{-1}(x)), \quad x \in \Omega_i. \quad (4.36)$$

we have to glue those scaling functions across interfaces of subdomains which do not vanish on these interfaces. To identify these functions it is

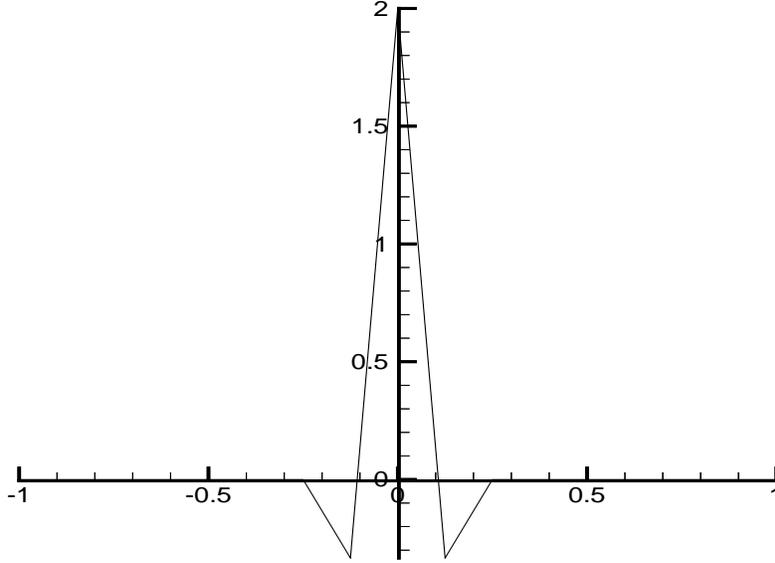


Figure 18: Matched and rescaled scaling functions at the interface.

convenient to associate *grid points* with the indices of our scaling functions. This can be done by defining for $k \in I_j$ and some $0 < x_k < 1$:

$$q(k) := \begin{cases} 0, & \varphi_{j,k}(0) \neq 0, \\ 1, & \varphi_{j,k}(1) \neq 0, \\ x_k, & \text{else,} \end{cases} \quad (4.37)$$

and for $k \in \mathbf{I}_j$

$$q(k) := (q(k_1), \dots, q(k_d)). \quad (4.38)$$

Thus a function $\varphi_{j,k}^i$ is supported inside a single subdomain Ω_i , if and only if $\kappa_i(q(k)) \in \Omega_i \setminus \partial\Omega_i$.

The corresponding grid points on the global domain are now defined using the parametric mappings and the grid points defined before:

$$\xi := \xi(i, k) := \kappa_i(q(k)). \quad (4.39)$$

For all points ξ on a common boundary of more than one subdomain, this ξ has several representations. If $r(\xi)$ is the number of subdomains $\bar{\Omega}_i$ where ξ belongs to, we have

$$\xi = \kappa_{i_1}(q(k^1)) = \dots = \kappa_{r(\xi)}(q(k^{r(\xi)})). \quad (4.40)$$

The idea is best illustrated in the univariate case. In fact, in the example presented in Figure 18, where $\kappa_1(x) := x - 1$ and $\kappa_2(x) = x$ we have $0 = \xi = \kappa_1(1) = \kappa_2(0)$ and $r(\xi) = 2$. At this point essential use is made of the property (ii) in Section 4.2.1 according to which for both ends of the interval only one function does not vanish on the boundary.

Moreover, the symmetry property (i) ensures compatibility of these matchings, recall (4.34).

The same principle works for matching across multivariate cubical subdomains. The scaling functions on the domain Ω are now defined as follows:

$$\varphi_{j,\xi}(x) := r(\xi)^{-1/2} \varphi_{j,k}^i(x), \quad x \in \Omega_i. \quad (4.41)$$

The analogous construction holds for the dual system providing the global generator bases $\Phi_j^\Omega, \tilde{\Phi}_j^\Omega$ on Ω which are *globally continuous* and hence suitable for the Galerkin discretization of the present second order problem.

The resulting system is biorthogonal with respect to the *modified inner product*

$$(v, w) := \sum_{i=1}^N \langle v \circ \kappa_i, w \circ \kappa_i \rangle. \quad (4.42)$$

In fact, biorthogonality in the interior of each subdomain is trivially retained. Using an appropriate scaling of those functions glued across an interface or around a vertex (e.g. by $r(\xi)^{-1/2}$ where $r(\xi)$ is the number of subdomains overlapped by the basis function associated with ξ) biorthogonality is restored for those basis functions as well [31], compare Figure 2 and Figure 18. In Figure 19 we present a typical scaling function overlapping the common boundary of two subdomains. One can see that for this case the function is the tensor product of the usual hat function (in y -direction) with the matched one dimensional function in x -direction shown in Figure 18.

Note that, on account of the assumptions on the parametric mappings κ_i , the modified inner product (4.42) is equivalent to the canonical inner product on Ω in the sense that

$$\| \cdot \|_{L_2(\Omega)}^2 \sim (\cdot, \cdot). \quad (4.43)$$

The Gramians of the mappings κ_i allow us of course to identify an explicit Riesz map $R : L_2(\Omega) \rightarrow L_2(\Omega)$ such that the pairs $\Phi_j^\Omega, R\tilde{\Phi}_j^\Omega$ are now biorthogonal on Ω with respect to the standard inner product $\langle \cdot, \cdot \rangle$. The fact that this Riesz map is a multiplication by a piecewise smooth but in general globally discontinuous function explains why these bases are less suitable for operators of order $\leq -1/2$ where constructions like [32, 30] still work.

So far we have accomplished step a) in Section 4.1. Step b) consists of identifying a suitable stable completion along with the corresponding two-scale matrices. There are several possibilities described in [31] which work for generator bases of any order. For simplicity our first tests will be based again on tensor products of piecewise linear trial functions and corresponding duals of lowest possible order, see Section 4.2.1. In this case the initial stable completion can again be based on hierarchical complement functions. What matters, in view of (4.34), is *Property B* which now facilitates an easy identification of the initial ingredients $\mathbf{M}_{j,1}, \mathbf{G}_j$ for initial stable completions on Ω , retaining the favorable symmetry properties as in Remark 4.4. Corresponding biorthogonal wavelets (with respect to the modified inner product (4.42)) are then again obtained by (4.6) or (4.11).

Of course, again homogeneous Dirichlet boundary conditions have to be incorporated in the trial spaces. As before we have the options (I), (II) described earlier. Recall that the effect on the wavelet bases is merely a different choice of $\tilde{\mathbf{M}}_{j,0}$ in (4.6). As in the

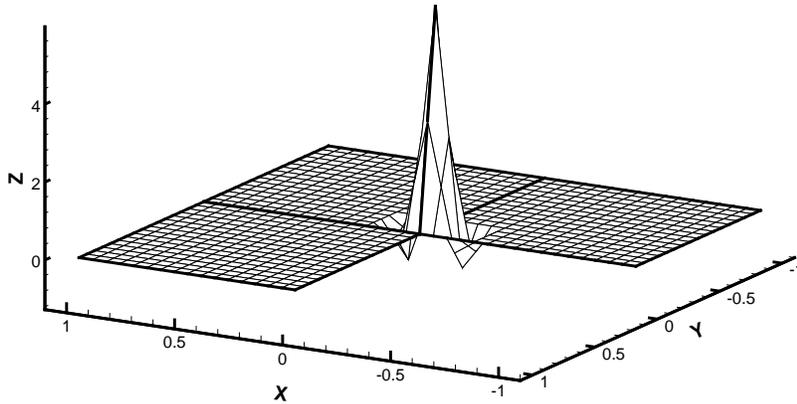


Figure 19: Scaling function supported in two subdomains.

1D–case $\omega_\lambda = 2^{|\lambda|}$ is a suitable choice for the diagonal scaling matrix \mathbf{D} in (2.8). Using the results in [31, 33] ensures that then the norm equivalence (2.8) holds for $H = H_0^1(\Omega)$. Moreover, concerning the cancellation properties and the compressibility of the operator representation \mathbf{A} the same statements as in the 1D–case apply so that the bases satisfy all the requirements from Sections 2.2 and 2.3. One should note though that when the κ_i are not just translations and rotations the cancellation properties (2.13) would deteriorate near the subdomain interfaces due to the modification of inner products. To avoid that one can use the constructions from [32, 30] which realize biorthogonality with respect to the standard inner product. In particular, in [32] conditions like (4.34) are not needed.

Note that all geometric information has been absorbed in the construction of the generator bases $\Phi_j^\Omega, \tilde{\Phi}_j^\Omega$. The wavelets are obtained by an evaluation, in principle, without a need to discuss different types of vertex situations in the partition of Ω . This together with a stronger exploitation of symmetry (to cope with (4.34)) distinguishes the construction in [31] from [14], say. In either case, the principal situation from the univariate case persists, although with significant technical complications. The local index sets entering the encoding of the global index sets (which in turn provide the links to the STL libraries) consist now of more distinct groups depending on the multiplicities $r(\xi)$.

In all applications it is of great importance not to destroy the tensor product structure, because exploiting this structure can significantly reduce computational costs compared to non tensor product structures. Computing the entries of the stiffness matrix for example

would only be possible for one or two wavelet levels because for larger levels the memory consumption would have been too large. Using the tensor product structure we are able to compute up to level nine which corresponds to a uniform grid of 784385 unknowns.

For the L-shaped domain the routines for constructing the wavelets are implemented by Vorloeper and described in detail in [50].

4.3.3 Discussion of Results

The test case described above is interesting for the following reason. Starting with the empty set, the residual in the first step is only influenced by the wavelet coefficients of the right-hand side. These wavelet coefficients are small near the vertex, due to the fact that u is harmonic there. Now in the next steps the adaptive scheme has to ‘recognize’ this deficiency and add wavelet coefficients at the ‘right’ places, namely in the vicinity of the vertex to resolve the there strong gradient of the solution appropriately.

Remark 4.8 *Recall that u has arbitrary high regularity in the Besov scale whereas for the Sobolev scale a direct verification shows that $u \in H^\alpha(\Omega)$, $\alpha < 5/3$. Consequently, uniform grids yield at best a convergence rate $N^{-5/6}$.*

In the Figures 20 and 21 we have depicted both, the approximate solution and the error to the exact solution. It can be seen that the adaptive algorithm indeed behaves like expected. First coefficients are added to reduce the error where strong gradients are induced by the right-hand side whereas in the subsequent iterations the error is reduced near the vertex, so that after five iterations the error is equally distributed. We see that similar to the 1D test problem the error again decreases very rapidly. In Figure 22 we have also depicted the sets of active wavelet coefficients corresponding to the fifth iteration of the adaptive algorithm. The first picture shows the set of coefficients corresponding to the scaling functions whereas in the remaining three we have treated the three different types of wavelets separately. It is shown in detail which coefficients are added on each refinement level. We see that the symmetry of the exact solution is reflected by the similarity of the pictures in the upper right and lower left corner. These two pictures clearly correspond to tensor product functions of wavelet/generator and generator/wavelet type, respectively.

Finally, we have also compared the performance of the adaptive algorithm with the best N -term approximation which can be computed very easily by collecting the N biggest wavelet coefficients. In Figure 23, we have depicted the errors as N increases. The continuous line corresponds to the best N -term approximation. We see that the matching between N -term and adaptive approximation is already pretty good. Indeed, the performance is in some sense better than expected. Observe that the wavelet basis is not very smooth so that no meaningful predictions can be made from the theoretical point of view. Again we expect that the approximation rate of the adaptive scheme can be dramatically increased by using smoother wavelet bases, for then the parameter s^* which determines the order of approximation grows. In fact, since in our case the solution u has arbitrary high Besov regularity, the order of convergence can also be made arbitrary high by increasing the smoothness of the wavelet basis. This topic will be studied in the near future.

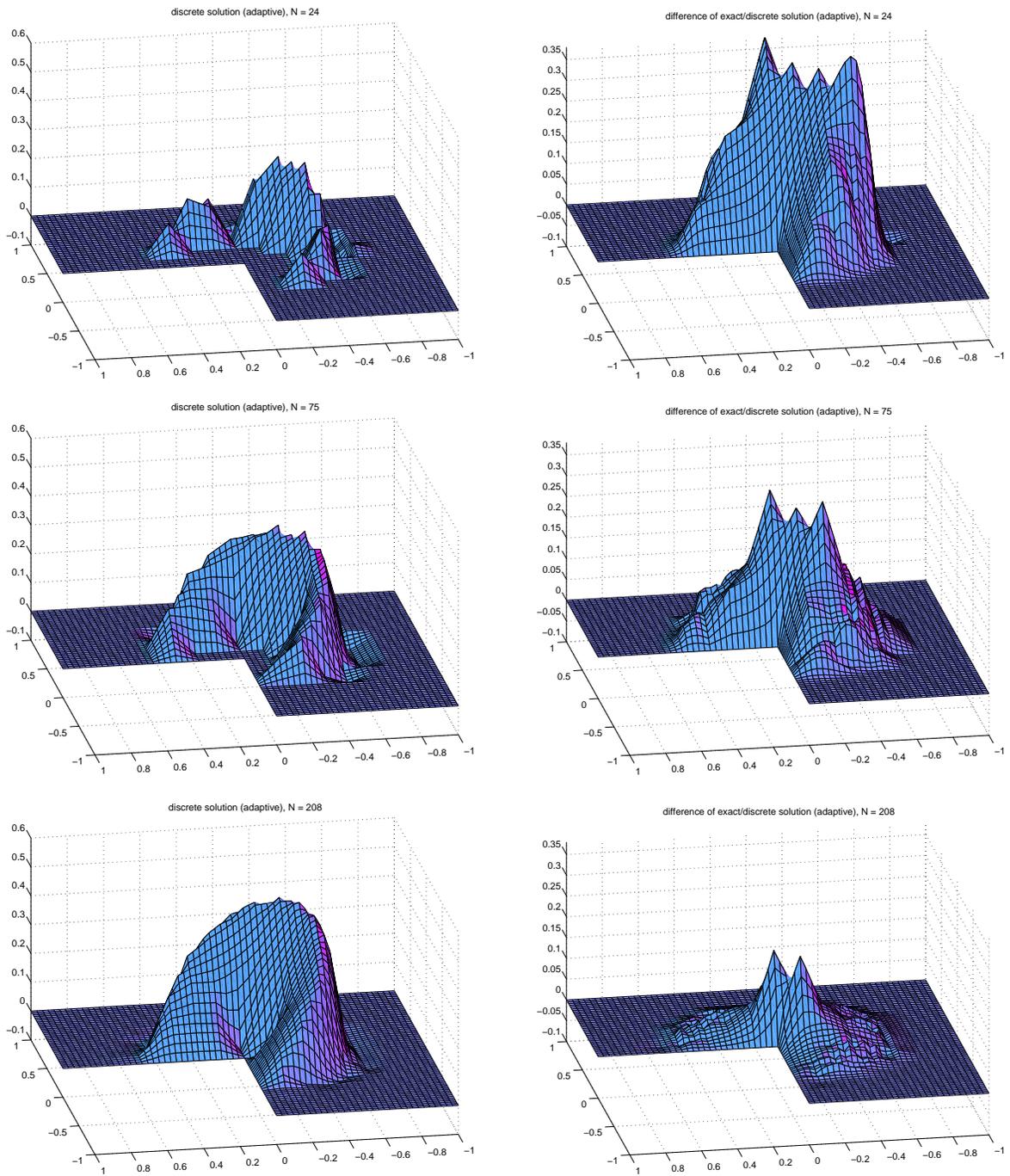


Figure 20: The first three approximate solutions and differences to exact solution.

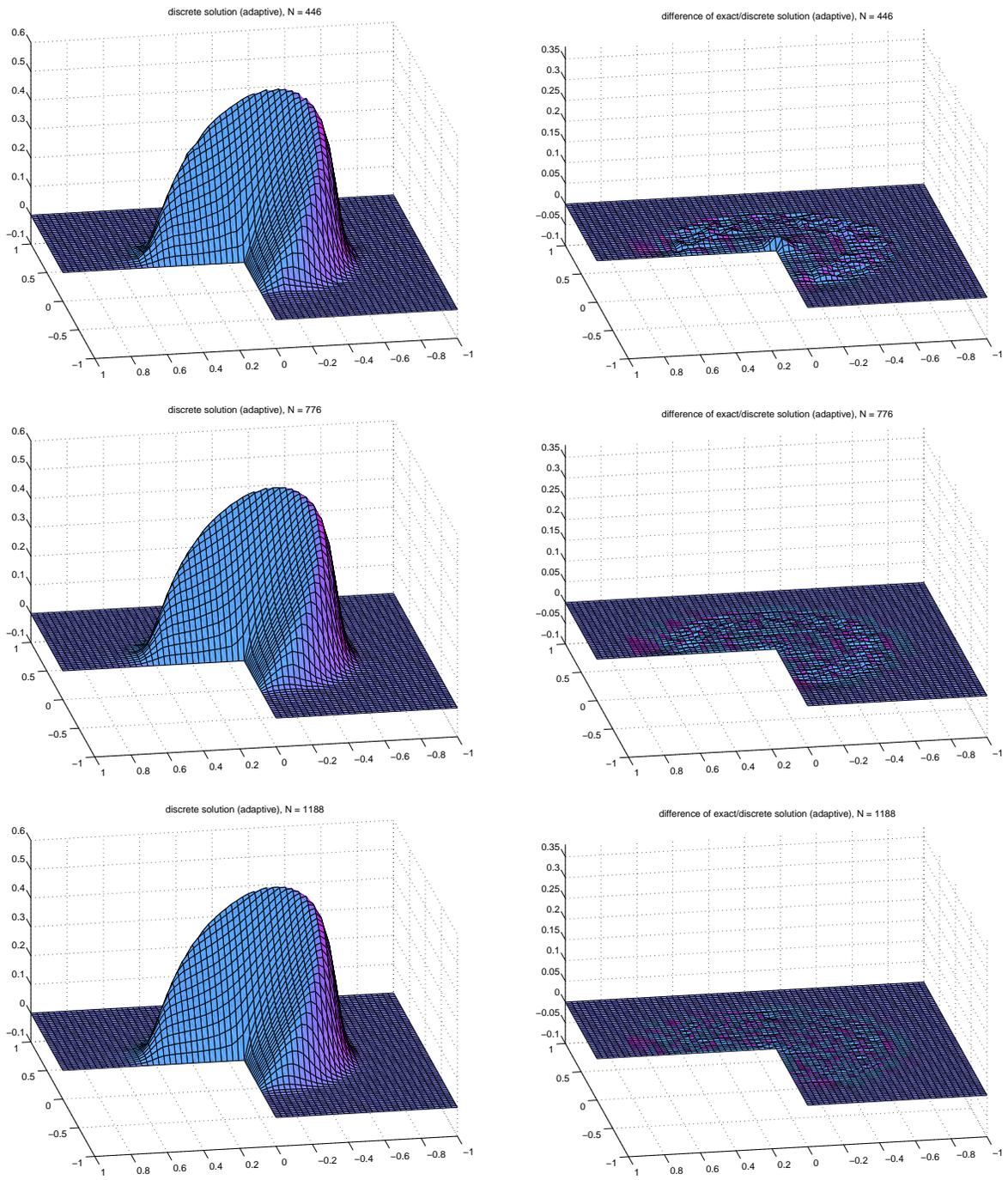


Figure 21: The next three approximate solutions and differences to exact solution.

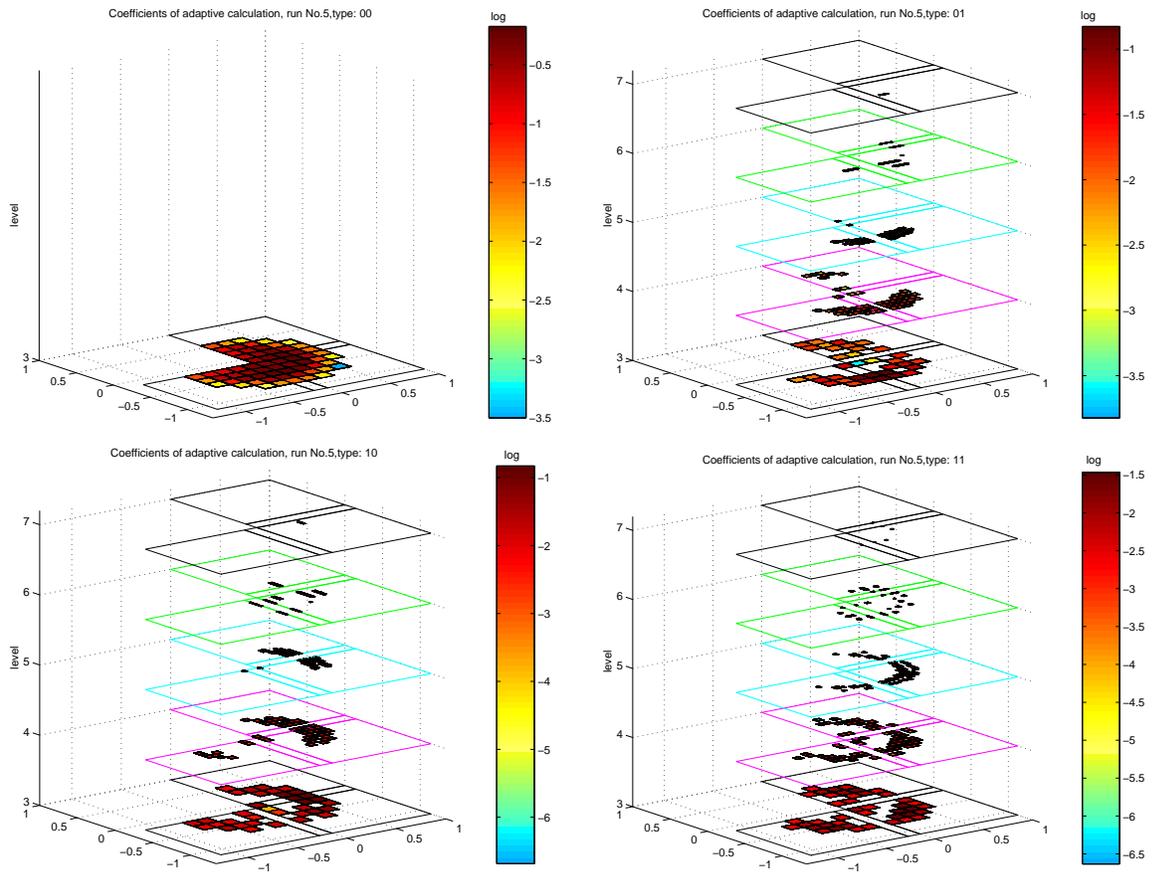


Figure 22: Index sets for the fifth iteration.

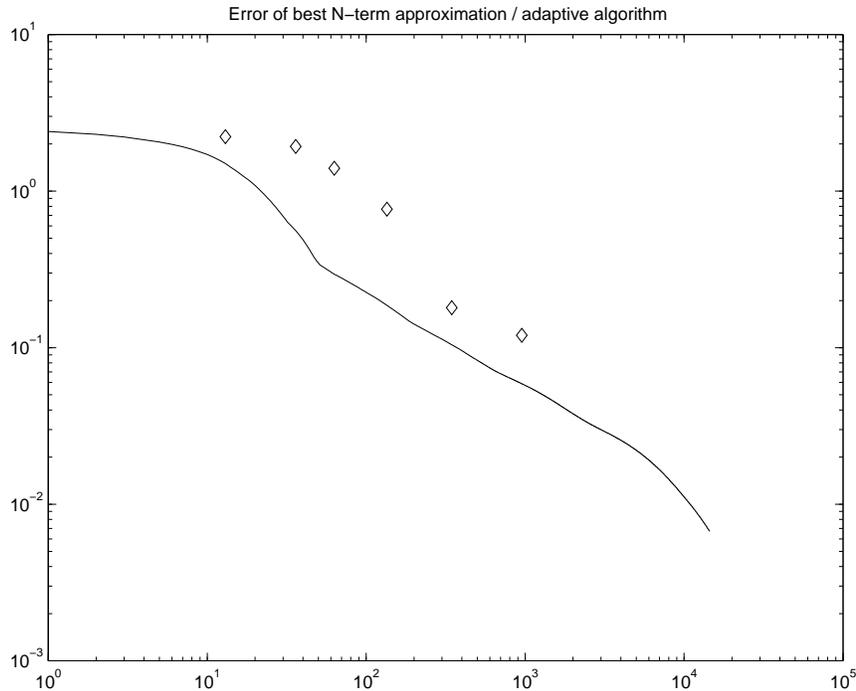


Figure 23: Comparison of best N -term approximation and adaptive algorithm.

Acknowledgements. The authors feel grateful to Yvon Maday for giving them the opportunity to participate in the European Summer School. They also want to thank Jürgen Vorloeper for putting his software concerning the composite wavelet basis at their disposal.

References

- [1] L. Anderson, N. Hall, B. Jawerth, and G. Peters, Wavelets on closed subsets of the real line, in: “Topics in the Theory and Applications of Wavelets”, (L.L. Schumaker and G. Webb, eds.), Academic Press, Boston, 1994, 1-61.
- [2] I. Babuška and W.C. Rheinboldt, A posteriori error estimates for finite element methods, *Int. J. Numer. Math. Engrg.* **12** (1978), 1597–1615.
- [3] I. Babuška and R. Rodriguez, The problem of the selection of an a-posteriori error indicator based on smoothing techniques, *Int. J. Numer. Math. Engrg.* **36** (1993), 539–567.
- [4] R.E. Bank and A. Weiser, Some a posteriori error estimators for elliptic partial differential equations, *Math. Comp.* **44** (1985), 283–301.

- [5] R.E. Bank and B.D. Welfert, A posteriori error estimates for the Stokes equation: a comparison, *Comp. Math. Appl. Mech. Engrg.* **87** (1990), 323–340.
- [6] A. Barinka, T. Barsch, K. Urban, and J. Vorloeper, The Multilevel Library: Software Tools for Multiscale Methods and Wavelets, Version 1.0, Documentation, RWTH Aachen, IGPM Preprint No. 156, 1998.
- [7] T. Barsch, A. Kunoth, and K. Urban, Towards Object Oriented Software Tools for Numerical Multiscale Methods for P.D.E.s using Wavelets, in: “Multiscale Methods for Partial Differential Equations”, (W. Dahmen, A. Kurdila, and P. Oswald, eds.), Academic Press, 1997, 383–412.
- [8] T. Barsch and K. Urban, Software Tools for using wavelets on the interval for the numerical solution of operator equations, RWTH Aachen, IGPM Preprint No. 158, 1998.
- [9] R. Becker, C. Johnson, and R. Rannacher, Adaptive error control for multigrid finite element methods, *Computing* **55** (1995), 271–288.
- [10] R. Becker and R. Rannacher, Weighted a posteriori error control in FE Methods, to appear in: “Proc. ENUMATH-95”, Paris, 1995.
- [11] R. Becker and R. Rannacher, A feed-back approach to error control in finite element methods: basic analysis and examples, *East-West Numer. Math.* **4** (1996), 237–264.
- [12] F. Bornemann, B. Erdmann, and R. Kornhuber, A posteriori error estimates for elliptic problems in two and three space dimensions, *SIAM J. Numer. Anal* **33** (1996), 1188–1204.
- [13] A. Cohen, I. Daubechies, and P. Vial, Wavelets on the interval and fast algorithms on an interval, *C.R. Acad. Sci. Paris* **316** (1993), 417-421.
- [14] C. Canuto, A. Tabacco, and K. Urban, The Wavelet Element Method, Part I: Construction and Analysis, *Appl. Comput. Harmonic Anal.* **6** (1999), 1–52.
- [15] C. Canuto, A. Tabacco, and K. Urban, The Wavelet Element Method, Part II: Realization and additional features in 2D and 3D, Istituto del Analisi Numerica del C.N.R. Pavia, Preprint No. 1052, 1997.
- [16] J.M. Carnicer, W. Dahmen, and J.M. Peña, Local decomposition of refinable spaces and wavelets, *Appl. Comput. Harmonic Anal.* **3** (1996), 127–153.
- [17] A. Cohen, I. Daubechies, and J.-C. Feauveau, Biorthogonal bases of compactly supported wavelets, *Comm. Pure and Appl. Math.* **45** (1992), 485-560.
- [18] A. Cohen, W. Dahmen, and R. DeVore, Adaptive Wavelet Methods for Elliptic Operator Equations — Convergence Rates, RWTH Aachen, IGPM Preprint No. 165, 1998.

- [19] A. Cohen and R. Masson, Wavelet adaptive method for second order elliptic problems, boundary conditions and domain decomposition, Université Pierre et Marie Curie, Paris, Preprint, 1997.
- [20] M. Costabel, Boundary integral operators on Lipschitz domains: elementary results, *SIAM J. Math. Anal.* **19** (1988), 613–626.
- [21] S. Dahlke: *Wavelets: Construction Principles and Applications to the Numerical Treatment of Operator Equations*, Habilitation thesis, Shaker Verlag, Aachen, 1997.
- [22] S. Dahlke: Besov regularity for elliptic boundary value problems on polygonal domains, *Appl. Math. Lett.* **12(6)** (1999), 31–36.
- [23] S. Dahlke, W. Dahmen, and R. DeVore, Nonlinear approximation and adaptive techniques for solving elliptic operator equations, in: “Multiscale Wavelet Methods for PDEs”, (W. Dahmen, A. Kurdila, and P. Oswald, eds.), Academic Press, San Diego, 1997, 237–284.
- [24] S. Dahlke, W. Dahmen, R. Hochmuth, and R. Schneider, Stable multiscale bases and local error estimation for elliptic problems, *Appl. Numer. Math.* **23** (1997), 21–47.
- [25] S. Dahlke and R. DeVore, Besov regularity for elliptic boundary value problems, *Comm. Partial Differential Equations* **22(1&2)** (1997), 1–16.
- [26] W. Dahmen, Stability of multiscale transformations, *J. Fourier Anal. Appl.* **2** (1996), 341–361.
- [27] W. Dahmen, Wavelet and multiscale methods for operator equations, *Acta Numerica* **6** (1997), Cambridge University Press, 55–228.
- [28] W. Dahmen, A. Kunoth, and K. Urban, Biorthogonal spline–wavelets on the interval — Stability and moment conditions, RWTH Aachen, IGPM Preprint No. 129, 1996, to appear in: *Appl. Comp. Harm. Anal.*
- [29] W. Dahmen, S. Prössdorf, and R. Schneider, Multiscale methods for pseudo-differential equations on smooth manifolds, in: “Wavelets: Theory, Algorithms, and Applications”, (C.K. Chui, L. Montefusco, and L. Puccio, eds.), Academic Press, 1994, 385–424.
- [30] W. Dahmen and R. Stevenson, Element-by-element construction of wavelets – stability and moment conditions, RWTH Aachen, IGPM Preprint No. 145, 1997.
- [31] W. Dahmen and R. Schneider, Composite Wavelet Bases for Operator Equations, TU Chemnitz, Preprint, revised version, Dec. 1997. To appear in: *Math. Comp.*
- [32] W. Dahmen and R. Schneider, Wavelets on Manifolds I: Construction and Domain Decomposition, RWTH Aachen, IGPM Preprint No. 149, 1998.
- [33] W. Dahmen and R. Schneider, Wavelets with Complementary Boundary Conditions - Function Spaces on the Cube, *Results in Mathematics* **34** (1998), 255–293.

- [34] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Math. **61**, SIAM, Philadelphia, 1992.
- [35] M. Dauge, Elliptic boundary value problems on corner domains, *Lecture Notes in Mathematics* **1341**, Springer, Berlin, 1988.
- [36] R. DeVore, Nonlinear Approximation, *Acta Numerica* **7** (1998), Cambridge University Press, 51–150.
- [37] R. DeVore, B. Jawerth, and V. Popov, Compression of wavelet decompositions, *Amer. J. Math.* **114** (1992), 737–785.
- [38] R. DeVore and V. Popov, Interpolation spaces and nonlinear approximation, in: “Function Spaces and Approximation”, (M. Cwikel, J. Peetre, Y. Sagher, and H. Wallin, eds.), Springer Lecture Notes in Math., 1988, 191–205.
- [39] W. Dörfler, A convergent adaptive algorithm for Poisson’s equation, *SIAM J. Numer. Anal.* **33** (1996), 737–785.
- [40] M. Frazier and B. Jawerth, Decomposition of Besov spaces, *Indiana Univ. Math. J.* **34** (1985), 777–799.
- [41] P. Grisvard, *Singularities in Boundary Value Problems*, Research Notes in Applied Mathematics **22**, Springer, 1992.
- [42] D. Jerison and C.E. Kenig, The inhomogeneous Dirichlet problem in Lipschitz domains, *J. of Funct. Anal.* **130** (1995), 161–219.
- [43] V.A. Kondrat’ev, Boundary–value problems for elliptic equations in domains with conical or angular points, *Trans. Moscow. Math. Soc.* **16** (1967), 227–313. Translated from: *Tr. Mosk. Mat. Obshch.* **16** (1967), 209–292.
- [44] M. König and R. Schneider, Object–oriented implementation of multiscale methods for boundary integral equations, SFB–Preprint No. 98–14, TU Chemnitz, 1998.
- [45] V.G. Mazya, Boundary integral equations, in: “Encyclopaedia of Math. Sciences **27**, Analysis IV”, (V.G. Mazya and S.M Nikol’skii, eds.), Springer, Berlin, Heidelberg, 1991.
- [46] Y. Meyer, *Wavelets and Operators*, Cambridge Studies in Advanced Mathematics **37**, Cambridge, 1992.
- [47] Standard Template Library Programmer’s Guide, SILICON GRAPHICS INC., 1996, “<http://www.sgi.com/Technology/STL/>”.
- [48] R. Verfürth, A posteriori error estimators for the Stokes equations, *Numer. Math.* **55** (1989), 309–325.
- [49] R. Verfürth, A posteriori error estimation and adaptive mesh-refinement techniques, *Jour. Comp. Appl. Math.* **50** (1994), 67–83.

- [50] J. Vorloeper, *Multiskalenverfahren und Gebietszerlegungsmethoden*, Master Thesis, in Preparation (in German).
- [51] H. Yserentant, Old and new convergence proofs for multigrid methods, *Acta Numerica* (1993), Cambridge University Press, 285–326.

Arne Barinka, Titus Barsch, Stephan Dahlke, Wolfgang Dahmen
RWTH Aachen
Institut für Geometrie und Praktische Mathematik
Templergraben 55
52056 Aachen
Germany
{barinka,barsch,dahlke,dahmen}@igpm.rwth-aachen.de

Philippe Charton
IREMIA – Université de la Réunion
15, avenue René Cassin - BP 7151
97715 Saint Denis Messag. Cedex 9
La Réunion
France
charton@univ-reunion.fr

Albert Cohen
Laboratoire d'Analyse Numerique
Université Pierre et Marie Curie
4 Place Jussieu, 75252 Paris cedex 05
France
cohen@ann.jussieu.fr

Karsten Urban
Istituto di Analisi Numerica del C.N.R.
Via Abbiategrosso, 209
27100 Pavia
Italy
urban@dragon.ian.pv.cnr.it