# Adaptive Wavelet Solvers for the Unsteady Incompressible Navier-Stokes Equations

Michael Griebel, Frank Koster*

Institut für Angewandte Mathematik, Universität Bonn
Wegelerstr. 6, D-53115 Bonn, Germany
e-mail: griebel@iam.uni-bonn.de, koster@iam.uni-bonn.de

**Abstract.** In this paper we describe adaptive wavelet-based solvers for the Navier-Stokes equations. Our approach employs a Petrov-Galerkin scheme with tensor products of Interpolet wavelets as ansatz functions. We present the fundamental algorithms for the adaptive evaluation of differential operators and non-linear terms. Furthermore, a simple but efficient preconditioning technique for the resulting linear systems is introduced. For the Navier-Stokes equations a Chorin-type projection method with a stabilized pressure discretization is used. Numerical examples demonstrate the efficiency of our approach.

## 1 Introduction

A main problem for the numerical solution of the unsteady Navier-Stokes equations is the large range of scales present in the solution. The most prominent estimate for that range is Kolmogorov's $Re^{3/4}$ law for isotropic, three-dimensional turbulence [28]. This leads to a huge number of degrees of freedom which are required to resolve *all* active scales. This number scales like $Re^{9/4}$ just for the spatial resolution. However, it is widely accepted that even in the turbulent regime the flow is governed by very localized, coherent structures. These structures are quasi-singularities in the flow which are characterized by strong gradients of the velocity, the vorticity and/or the pressure. One typical example are the essentially two-dimensional vortex rolls known for example from cigarette smoke. This suggests that the *intrinsic* complexity of turbulent flows is much smaller than that estimated by Kolmogorov's law. It should be possible to exploit this fact by means of *adaptive* solution methods.

The main advantage of wavelets are their localization in space and scale. Therefore, they are perfect candidates for adaptive methods. More precisely it can be shown that functions which have at most a few (quasi-)singularities can be approximated with a quite small number of degrees of freedom in terms of a wavelet basis. For a recent review on this topic we refer to the excellent paper [23]. The foundation for this property are the following two aspects: First, polynomials up to a certain degree are in the span of a wavelet basis. This explains their approximation power. Second, for a certain range of Sobolev spaces $H^s$ the Sobolev-norms of wavelet expansions and sequence-norms of the expansion coefficients are equivalent.

* Research supported by the Deutsche Forschungsgemeinschaft, GR 1144/7-2

These norm equivalences also play a role for an important practical problem: The efficient preconditioning of the large linear systems that have to be solved during the numerical integration of the Navier-Stokes equations. It is this unification of insight into approximation theory on one side and the efficient preconditioning of linear systems on the other side which makes up the beauty of wavelets and which explains to some extent the interest they have attracted in the last years.

However, wavelets are not for free. The associated algorithms for the evaluation of linear or non-linear terms are in general quite complicated. A main topic of this paper is the description of such basic algorithms for adaptive wavelet methods for the treatment of the Navier-Stokes equations. Of course, these algorithms are dependent on the wavelets at hand. There exist orthogonal Daubechies wavelets [19–21], biorthogonal spline wavelets [16] or Interpolets [22,26]. In this paper we will use Interpolets.

To treat the multivariate case, there are two different types of constructions by means of tensor products of univariate wavelets. The first one employs tensor products of wavelets and so-called scaling functions of the same level of refinement. This leads to a linear order of successively finer approximations spaces as in the univariate case, i.e. to a multi-resolution analysis (MRA) [21]. However, in this paper we will use another approach. Here, simple tensor products of wavelets of arbitrary levels of refinement are used similar to Fourier spectral methods. This technique does not lead to a linear order of approximation spaces, but instead to a $d$-dimensional array of approximation spaces (c.f. [53] sect. 5.4), where $d$ denotes the dimension of the computational domain. Therefore, we will call this approach MRA-$d$. The main advantage of this approach is its excellent approximation property. For example, for the approximation of functions with bounded mixed derivatives a much smaller number of degrees of freedom is required compared to the classical MRA-approach. This property has been extensively studied in e.g. [24,34,62]. Another situation where the MRA-$d$-approach is superior to the MRA-approach is the approximation of strong anisotropic functions like the velocity or vorticity field in a turbulent boundary layer with its strong anisotropic coherent structures (so-called hair-pin vortices) [49].

A further advantage of our MRA-$d$-approach is its algorithmical simplicity. For the discretization of PDEs we use a Petrov-Galerkin scheme with specially chosen test functions. Then, the adaptive evaluation of discrete differential operators completely boils down to the univariate case. We furthermore use univariate Interpolets as basic building blocks in the MRA-$d$-approach. Then also the adaptive evaluation of non-linear terms essentially boils down to the univariate adaptive wavelet transform and its inverse.

Hence, in our opinion the MRA-$d$-approach with Interpolet wavelets and a Petrov-Galerkin discretization is the simplest way to carry the adaptive univariate wavelet setup over to the adaptive multivariate case. This also allows for very efficient data-structures and algorithms. Furthermore, code developed for the univariate case can directly be reused.

The efficient solution of the resulting linear systems is another important issue. To solve this problem we use a simple preconditioning technique based on a diagonal scaling together with the lifting scheme [13,60]. This leads to $O(1)$ condition numbers of the linear systems of equations and, therefore, to efficient iterative solution methods.

For the discretization of time-dependent problems we will use a common time-stepping method. This method is combined with an algorithm for the selection of the adaptive basis in space. Here, the main idea is that locations where adaptive refinement is required evolve smoothly in time. Therefore, adaptive refinement for a new time slab is required in the vicinity of regions which were refined in the previous time slab.

All these ingredients (MRA-$d$, Petrov-Galerkin, Interpolets, time stepping methods and adaptive basis selection) form the building blocks of an adaptive wavelet-based Navier-Stokes solver. According to our 'efficient, but as simple as possible' strategy we here employ a Chorin-type projection method. This method decouples the momentum and continuity equation and leads to a Poisson equation for the pressure in each time step. We present two numerical experiments which show that our wavelet-based Navier-Stokes solver works very well and yields accurate and reliable results.

This paper is organized as follows. In the following section we introduce the setting of univariate biorthogonal wavelets and approximation spaces for the example of Interpolets. Furthermore, we derive a general algebraic form of the wavelet transform algorithms and we briefly sketch how wavelets on the interval are built. In the third section we carry our setting over to the adaptive case. Then, we also present the basic adaptive algorithms for the wavelet transform, for its inverse, for the adaptive evaluation of non-linear terms and for the matrix-vector multiplication with the (generalized) stiffness matrix. In addition we also briefly introduce the wavelet-finite-difference technique as an alternative to the Petrov-Galerkin approach. The multivariate MRA-$d$-approach is described in section 4. There we show how the previous adaptive algorithms for the univariate case can be used in the multivariate case as well. In section 5 we briefly present what is known about consistency and stability of the Petrov-Galerkin method associated to MRA-$d$-Interpolets. The problem of efficient preconditioning of linear systems which arise from elliptic equations is treated in section 6. Here, we introduce two simple preconditioners which exploit the norm-equivalences mentioned earlier. Furthermore, an analysis of these preconditioners is given for the non-adaptive case. After these preparations we describe in section 7 how adaptive wavelet methods can be applied to time-dependent problems such as convection-diffusion equations. Here, we describe how in a time-stepping method the adaptive basis is selected from time step to time step. Section 8 deals with the correct treatment of the continuity equation and with the stabilization of the pressure discretization. The main idea is to use a Chorin type projection method together with a consistent discretization of the pressure-Poisson equation. For stabilization reasons we introduce an upwind-downwind discretization for the pressure gradient and the divergence operator, respectively. Then, we give numerical examples where our adaptive Interpolet Petrov-Galerkin method developed so far is applied to two complicated two-dimensional flow problems. The first is the interaction and merger of three vortices and the second is the evolution of a shear layer.

In order to simplify the reading and understanding we give a detailed table of symbols and notation at the end of this paper.

## 2 Univariate Interpolets

In this section we introduce the scaling functions, the wavelets and their dual counterparts for the Interpolet family [22,27]. Furthermore, basic notation is set up. To this end, we first consider spaces of functions defined on $\mathbb{R}$. Interpolets of order $N$ ($N \in \mathbb{N}$, $N$ even) are then defined via a single scaling function $\phi$ with the following properties:

• (*Scaling equation:*) The function $\phi$ is the solution of

$$\phi(x) = \sum_{s\in\mathbb{Z}} h_s \phi(2x - s) \ . \tag{1}$$

The mask coefficients $\mathbf{h} := \{h_s\}_{s\in\mathbb{Z}}$ are given by $h_0 = 1$, $h_s = h_{-s}$ ($s \in \mathbb{Z}$) and

$$\begin{pmatrix} 1^0 & 3^0 & \dots & (N-1)^0 \\ 1^2 & 3^2 & \dots & (N-1)^2 \\ \vdots & \vdots & & \vdots \\ 1^{N-2} & 3^{N-2} & \dots & (N-1)^{N-2} \end{pmatrix} \cdot \begin{pmatrix} h_1 \\ h_3 \\ \vdots \\ h_{N-1} \end{pmatrix} = \begin{pmatrix} 1/2 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \ . \tag{2}$$

All other $h_s$ are zero.

• (*Compact support:*)

$$\text{supp } \phi = [-N+1, N-1] \ , \quad \text{supp } \mathbf{h} = \{-N+1, ..., N-1\} \ . \tag{3}$$

• (*Polynomial exactness:*) In a pointwise sense, polynomials of degree *less* than $N$ can be written as linear combination of translates of $\phi$, e.g.

$$\forall\ 0 \le i < N \ : \ \forall x \in \mathbb{R} \quad : \quad x^i = \sum_{s\in\mathbb{Z}} s^i \phi(x - s) \ . \tag{4}$$

• (*Interpolation property:*) Let $\delta(x)$ be the Dirac functional. It should hold

$$\forall s \in \mathbb{Z} \quad : \quad \phi(s) = \delta(s) \ . \tag{5}$$

This is a special property of the Interpolet family. The scaling functions of other wavelet types usually do *not* fulfill this relation.

• (*Dual scaling functions:*) Let $\tilde{\phi}(x) := \delta(x)$. Then (5) can also be written as a biorthogonality relation

$$\langle \tilde{\phi}(x - s) \ , \ \phi(x - t) \rangle = \delta(s - t) \ . \tag{6}$$

This means that translates $\{\tilde\phi(.-s)\}_{s\in\mathbb{Z}}$ of the Dirac functional are the dual scaling functions to $\{\phi(.-s)\}_{s\in\mathbb{Z}}$. Clearly the function $\tilde\phi$ also satisfies a scaling equation since

$$\tilde\phi(x)=\sum_{s\in\mathbb{Z}}\tilde h_s\tilde\phi(2x-s) \quad \text{where} \quad \tilde h_s=\delta(s)\ . \tag{7}$$

• (*Complementary spaces:*) The odd translates $\phi(2x-(2t+1))$ can now be considered as so-called wavelets (see the discussion below and section 2.1). We have

$$\operatorname{span}\{\phi(x-s)\}_{s\in\mathbb{Z}}\oplus\operatorname{span}\{\phi(2x-(2t+1))\}_{t\in\mathbb{Z}}=\operatorname{span}\{\phi(2x-s)\}_{s\in\mathbb{Z}}. \tag{8}$$

This is the main property which distincts Interpolet wavelets from other wavelet families. Since the wavelets are interpolating again, the algorithms for the evaluation of non-linear terms become particularly simple. In the remainder of this paper we will denote wavelets with properties (1) – (8) as Interpolets or Interpolet wavelets [22,26].

### 2.1 Multi-Resolution Analysis

Equipped with functions $\phi(x)$ and $\tilde\phi(x)$ which possess the above properties (1)–(8), we now define a multi-resolution analysis of $L^2(\mathbb{R})$. To this end, we define approximation spaces $V^l$ spanned by translates and dilates of $\phi$, i.e.

$$V^l=\operatorname{span}\{\phi^{(l,s)}\}_{s\in\mathbb{Z}}\ ,\quad l\in\mathbb{N} \quad \text{where} \quad \phi^{(l,s)}(x):=\phi(2^lx-s)\ . \tag{9}$$

Because of (1) the $V^l$ form a ladder of successively finer approximation spaces

$$...\subset V^l\subset V^{l+1}\subset...\subset L^2(\mathbb{R})\ . \tag{10}$$

The properties of polynomial exactness (4) and compact support (3) together with a Bramble-Hilbert argument yield a classical Jackson-type approximation estimate

$$\inf_{u^l\in V^l}\|u-u^l\|_0\le c2^{-lN}\|u^{(N)}\|_0\ ,$$

where $u^{(N)}$ is the $N$th derivative of $u$. Of course the *local* approximation quality on e.g. $\operatorname{supp}\phi^{(l,s)}$ depends only on $\int_{\mathcal{N}}|u^{(N)}|^2\,dx$ for a neighbourhood $\mathcal{N}$ of $\operatorname{supp}\phi^{(l,s)}$. Furthermore it is clear that in regions where $u^{(N)}$ is quite small, there is no significant difference between the best approximation $u^l\in V^l$ and the best approximation $u^{l-1}\in V^{l-1}$. This is the main idea behind any multi-scale representation which attempts to decompose the approximation spaces $V^l$ into the sum of complementary spaces $W_l$. There we have

$$V^L=V^{l_0}\oplus W_{l_0+1}\oplus W_{l_0+2}\oplus..\oplus W_L \quad \text{where} \quad W_l=V^{l-1}\oplus V^l\ .$$

Here, $V^{l_0}$ is a certain coarsest approximation space which is later on related to wavelets on the interval and which can not be further decomposed (see subsection 2.3). To simplify the notation we set

$$W_{l_0} := V^{l_0} \ , \quad \text{i.e. } V^L = \oplus_{l=l_0}^{L} W_l \ .$$

In the above-mentioned situation of local smoothness in the neighbourhood of $\operatorname{supp} \phi^{(l,s)}$ we can assume the following: If $l$ is sufficiently large then the local contributions of $W_l$ are quite small and may be discarded without introducing significant errors. This means that an approximate representation of a function $u \in V^L$ will be much sparser in terms of basis functions of $W_l$ than the single-scale representation in terms of $\phi^{(L,s)}$.

At the present stage neither the spaces $W_l$ nor an appropriate basis of $W_l$ were determined yet. Property (8) suggests a simple definition of $W_l$:

$$W_l := \operatorname{span}\{\psi_{(l,t)}\}_{t\in\mathbb{Z}} \ , \quad \text{where} \quad \psi_{(l,t)} = \begin{cases} \phi^{(l,2t+1)} & \text{if } l > l_0 \\ \phi^{(l_0,t)} & \text{if } l = l_0 \ . \end{cases} \tag{11}$$

First of all, $\{\psi_{(l,t)}\}_{t\in\mathbb{Z}}$ is indeed a basis of $W_l$ and $V^l = W_l \oplus V^{l-1}$ is a direct sum. In the following the basis functions $\psi_{(l,s)}$ will be called wavelets and in case of the Interpolet family alternatively Interpolets or Interpolet wavelets. Figure 1 depicts the different bases for Interpolets in the case $N = 6$.

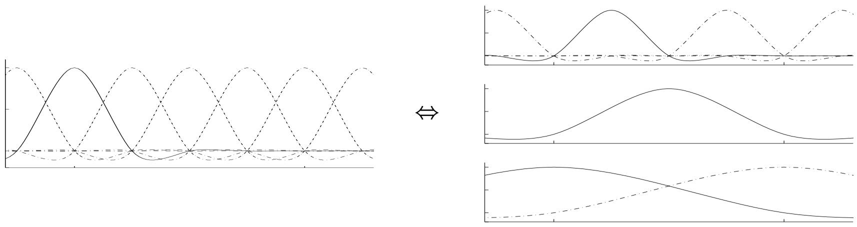

**Fig. 1.** Left: single-scale basis. Right: multi-scale basis.

## 2.2 Wavelet Transform

With a specific choice of scaling functions, wavelets and their duals the complete multi-resolution analysis is determined. It then remains to give algorithms which calculate the multi-scale representation from a single-scale representation. To simplify the explanation we will use the following notation in the remainder of this paper,

– *row* vectors of functions

$$\boldsymbol{\Phi}^l := \{\phi^{(l,s)}\}_{s\in\mathbb{Z}} \ , \quad \boldsymbol{\Psi}_l := \{\psi_{(l,t)}\}_{t\in\mathbb{Z}} \ , \tag{12}$$

– *column* vectors of coefficient sequences

$$\mathbf{u}^l := \{u^{(l,s)}\}_{s\in\mathbb{Z}} \ , \quad \mathbf{u}_l := \{u_{(l,t)}\}_{t\in\mathbb{Z}} \ . \tag{13}$$

With this notation we get the following representations of $u \in V^L$

$$u = \sum_{s\in\mathbb{Z}} u^{(L,s)}\phi^{(L,s)} = \boldsymbol{\Phi}^L \cdot \mathbf{u}^L$$
$$= \sum_{l=l_0}^{L} \sum_{t\in\mathbb{Z}} u_{(l,t)}\psi_{(l,t)} = \sum_{l=l_0}^{L} \boldsymbol{\Psi}_l \cdot \mathbf{u}_l \ .$$

Note that here and in the following superscript indices are used for coefficients and functions in a scaling function representation and subscript indices are applied for coefficients and functions in a wavelet representation.

Now, the wavelet transform has to compute the coefficients $\{\mathbf{u}_l\}_l$ with respect to the multi-scale basis $\{\boldsymbol{\Psi}_l\}_l$ from the coefficients $\mathbf{u}^L$ with respect to the single-scale basis $\boldsymbol{\Phi}^L$. In case of Interpolets there is a very simple interpretation of the scheme: First of all, note that $\mathbf{u}^L$ are the nodal values of $u$ in the grid points $\{2^{-L}s\}_{s\in\mathbb{Z}}$ because of (5). Now consider one step of the wavelet transform, i.e.

$$\boldsymbol{\Phi}^L \cdot \mathbf{u}^L \stackrel{!}{=} \boldsymbol{\Phi}^{L-1} \cdot \mathbf{u}^{L-1} \ + \ \boldsymbol{\Psi}_L \cdot \mathbf{u}_L \tag{14}$$

which is associated to the splitting $V^L = V^{L-1} \oplus W_L$. To calculate for example $u_{(L,t)}$ we pick the odd node $2^{-L}(2t+1)$ and his neighbouring evenly numbered nodes $\mathcal{N} := 2^{-L}\{2t-N+2,\ 2t-N+4,\ \ldots,\ 2t+N-2\}$. We calculate the polynomial $p_\mathcal{N}$ of degree $N-1$ which interpolates at $\{\mathcal{N}, u(\mathcal{N})\}$. Then, the coefficient $u_{(L,t)}$ is given by

$$u_{(L,t)} := (u - p_\mathcal{N})(2^{-L}(2t+1)) \ , \tag{15}$$

see Figure 2 for an example. This is done for all $t \in \mathbb{Z}$. Because of (5) the auxiliary coefficients $\{u^{(L-1,s)}\}_{s\in\mathbb{Z}}$ are given by $u^{(L-1,s)} = u^{(L,2s)}$.

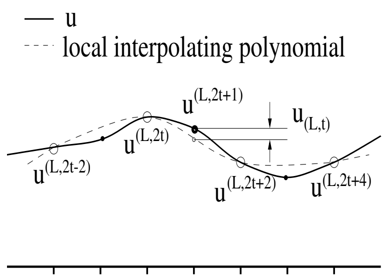


**Fig. 2.** Calculation of wavelet coefficients

To set up the above algorithm more formally it is appropriate to introduce the complete machinery of dual scaling functions and wavelets. Analogously to the decomposition $V^L = \oplus_l W_l$, we can decompose the dual space as

$$\tilde{V}_L = \mathrm{span}\{\tilde{\phi}^{(L,s)}\}_{s\in\mathbb{Z}} = \oplus_{l=l_0}^{L} \tilde{W}_l$$

where $\tilde{W}_l = \mathrm{span}\{\tilde{\psi}_{(l,t)}\}$ and the dual wavelets $\tilde{\psi}_{(l,t)}$ are linear combinations of the elements of $\tilde{\boldsymbol{\Phi}}^l$. More precisely we have

$$\tilde{\psi}_{(l,t)}(x) = \sum_{r\in\mathbb{Z}} \tilde{g}_{r-2t} \tilde{\phi}^{(l,r)}(x) \ , \tag{16}$$

where the mask coefficients $\tilde{g}_r$ are defined by

$$\tilde{g}_r := (-1)^{1-r} h_{1-r} \ . \tag{17}$$

This choice ensures that we have the following *biorthogonality relations*

$$\begin{aligned} \langle \tilde{\boldsymbol{\Phi}}^l, \boldsymbol{\Phi}^l \rangle &= I \ , \quad \langle \tilde{\boldsymbol{\Psi}}^l \ , \boldsymbol{\Phi}^{l-1} \rangle = \mathbf{0} \ , \\ \langle \tilde{\boldsymbol{\Psi}}^l, \boldsymbol{\Psi}^l \rangle &= I \ , \quad \langle \tilde{\boldsymbol{\Phi}}^{l-1}, \boldsymbol{\Psi}^l \ \rangle = \mathbf{0} \ . \end{aligned}$$

An immediate consequence of these relations are the general biorthogonality relations of the wavelets and their duals

$$\langle \tilde{\psi}_{(k,s)} \ , \ \psi_{(l,t)} \rangle = \delta(k-l)\delta(s-t) \ . \tag{18}$$

We also can recast the scaling relations (1), (7), (11) and (16) as matrix-vector products

$$\boldsymbol{\Phi}^{l-1} = \boldsymbol{\Phi}^l \cdot H^l \ , \quad \tilde{\boldsymbol{\Phi}}^{l-1} = \tilde{\boldsymbol{\Phi}}^l \cdot \tilde{H}^l \ , \quad \boldsymbol{\Psi}^l = \boldsymbol{\Phi}^l \cdot G^l \ , \quad \tilde{\boldsymbol{\Psi}}^l = \tilde{\boldsymbol{\Phi}}^l \cdot \tilde{G}^l \ , \tag{19}$$

where $H^l$, $\tilde{H}^l$, $G^l$ and $\tilde{G}^l$ are sparse matrices, $(H^l)_{r,s} = h_{r-2s}$, $(\tilde{H}^l)_{r,s} = \tilde{h}_{r-2s}$, $(G^l)_{r,s} = \delta(r-(2s+1))$ and $(\tilde{G}^l)_{r,s} = \tilde{g}_{r-2s}$. In terms of these matrices the biorthogonality relations read

$$\left(H^l, G^l\right) = \left(\tilde{H}^l, \tilde{G}^l\right)^{-T} \ . \tag{20}$$

Furthermore, we have the following algebraic representation of one step of the wavelet transform

$$\begin{aligned} \boldsymbol{\Phi}^L \cdot \mathbf{u}^L &= \boldsymbol{\Phi}^{L-1} \cdot \langle \tilde{\boldsymbol{\Phi}}^{L-1}, \ \boldsymbol{\Phi}^L \rangle \cdot \mathbf{u}^L \ + \ \boldsymbol{\Psi}^L \cdot \langle \tilde{\boldsymbol{\Psi}}^L, \ \boldsymbol{\Phi}^L \rangle \cdot \mathbf{u}^L \\ &= \boldsymbol{\Phi}^{L-1} \cdot (\tilde{H}^L)^T \cdot \mathbf{u}^L \qquad + \ \boldsymbol{\Psi}^L \cdot (\tilde{G}^L)^T \cdot \mathbf{u}^L \ , \end{aligned}$$

and, consequently, we get from (14)

$$\mathbf{u}^{L-1} = (\tilde{H}^L)^T \cdot \mathbf{u}^L \ , \quad \mathbf{u}_L = (\tilde{G}^L)^T \cdot \mathbf{u}^L \ .$$

The complete wavelet transform and its inverse read

| **algorithm (WT)** | **algorithm (IWT)** |
|---|---|
| given $\mathbf{u}^L$<br>`for` $l = L$ `downto` $l_0 + 1$<br>$\quad \mathbf{u}_l \ = (\tilde{G}^l)^T \cdot \mathbf{u}^l$<br>$\quad \mathbf{u}^{l-1} = (\tilde{H}^l)^T \cdot \mathbf{u}^l$<br>`end` | given $u^{l_0}$ and $\{\mathbf{u}_l\}_{l=l_0+1}^{l}$<br>`for` $l = l_0 + 1$ `to` $L$<br>$\quad \mathbf{u}^l = G^l \cdot \mathbf{u}_l \ + \ H^l \cdot \mathbf{u}^{l-1}$<br>`end` |

### 2.3 Wavelets on the Interval

The MRA for functions defined on $\mathbb{R}$ can not be used for the solution of PDEs on finite domains. To this end the above setting must be modified to function spaces defined on bounded domains. We consider the domain $[0,1]$ and briefly describe such a construction for Interpolets. A detailed description of a similar approach for orthogonal wavelets on the interval is given in [45].

The main idea is the following: We have to find scaling functions defined on $[0,1]$ which fulfill a scaling equation similar to (19) and which span the polynomials of degree less than $N$ for $0 \leq x \leq 1$ to maintain the approximation properties.

To this end consider the scaling functions $\{\phi^{(l,s)}\}_{N \leq s \leq 2^l - N}$ for sufficiently large $l$, i.e. $l \geq l_0$, where

$$l_0 = \lceil log_2(2N) \rceil \ .$$

Obviously these scaling functions have their support completely in $[0,1]$ and they also satisfy a scaling relation with $\{\phi^{(l+1,s)}\}_{N \leq s \leq 2^{l+1} - N}$. However, they do not span all polynomials of degree less than $N$ in $[0,1]$. The remedy is to augment $\{\phi^{(l,s)}\}_{N \leq s \leq 2^l - N}$ by additional scaling functions $\phi^{(l,0)}, .., \phi^{(l,N-1)}$ for the left boundary $x = 0$ and $\phi^{(l,2^l-N+1)}, .., \phi^{(l,2^l)}$ for the right boundary $x = 1$ which are no longer simple dilates and translates of $\phi$, but instead are defined by

$$\left(\phi^{(l,0)}, .., \phi^{(l,N-1)}\right)(x) := \left(\phi(2^l x + N - 2), .., \phi(2^l x - N + 1)\right) \cdot A \qquad (21)$$

and

$$\left(\phi^{(l,2^l-N+1)}, .., \phi^{(l,2^l)}\right)(x) := \left(\phi(2^l x - 2^l - N + 1), .., \phi(2^l x - 2^l + N - 2)\right) \cdot B \ .$$

Here, the matrices $A$ and $B$ consist of $N$ specially chosen eigenvectors of the matrix $\{h_{r-2s}\}_{-N+2 \leq r,s \leq N-1}$ such that, altogether, the boundary modified scaling functions ($0 \leq s < N$, $2^l - N < s \leq 2^l$) together with the ordinary translates and dilates ($N \leq s \leq 2^l - N$) satisfy:

- (*Scaling equation:*)

$$\mathrm{span}\{\phi^{(l,s)} \mid 0 \leq s \leq 2^l\} \subset \mathrm{span}\{\phi^{(l+1,s)} \mid 0 \leq s \leq 2^{l+1}\} \ . \qquad (22)$$

- (*Polynomial exactness:*) For $0 \leq x \leq 1$ and $0 \leq i < N$ there holds

$$x^i = \sum_{s=0}^{2^l} s^i \phi^{(l,s)}(x) \ . \qquad (23)$$

- (*Interpolation property:*)

$$\forall 0 \leq s, t \leq 2^l \quad : \quad \phi^{(l,s)}(2^{-l} t) = \delta(s - t) \ . \qquad (24)$$

In case of a prescribed homogeneous Dirichlet boundary condition in for example the point $x = 0$ there exist just $N - 1$ modified scaling functions $\phi^{(l,1)}, ..., \phi^{(l,N-1)}$. In this case $A$ consists of only $N - 1$ eigenvectors of the

matrix $\{h_{r-2s}\}_{-N+2\le r,s\le N-1}$ and (23) is satisfied only for $1 \le i < N$ and (24) only for $s,t \ne 0$.
Likewise, for a prescribed homogeneous Neumann boundary condition in for example $x = 0$ there exist just the $N-1$ boundary modified scaling functions $\phi^{(l,0)}, \phi^{(l,2)}, ..., \phi^{(l,N-1)}$. This slightly different labeling $(0,2,3,..,N-1)$ instead of $(1,2,..,N-1)$ is motivated by the fact that here (23) holds for $i \in \{0,2,...,N-1\}$ and (24) for $s,t \ne 1$. Analogous constructions hold for the point $x = 1$.

In order to simplify the further explanations we define row vectors of scaling functions again by

$$\boldsymbol{\Phi}^l = \{\phi^{(l,s)}\}_{s\in S(l)} \ ,$$

with the index sets $S(l) = \{0,..,2^l\}$ (or $\{1,2,..,2^l-1\}$ or $\{0,2,..,2^l-2,2^l\}$ depending on the boundary conditions). The corresponding dual scaling functions are

$$\tilde{\boldsymbol{\Phi}}^l := \{\tilde{\phi}^{(l,s)}\}_{s\in S(l)} \quad \text{where} \quad \tilde{\phi}^{(l,s)}(x) := \delta(2^l x - s) \ . \tag{25}$$

Similar to subsection 2.1, for $l > l_0$ the wavelets $\psi_{(l,t)}$ are essentially the odd numbered scaling functions $\phi^{(l,2t+1)}$

$$\boldsymbol{\Psi}_l := \{\psi_{(l,t)}\}_{t\in T(l)} \ , \quad T(l) = \{0,...,2^{l-1}-1\} \tag{26}$$

and

$$\psi_{(l,t)} := \begin{cases} \phi^{(l,2)} & \text{if } t = 0 \text{ and homog. Neumann b.c. in } x = 0 \ , \\ \phi^{(l,2^l-2)} & \text{if } t = 2^{l-1}-1 \text{ and homog. Neumann b.c. in } x = 1 \ , \\ \phi^{(l,2t+1)} & \text{else} \ . \end{cases} \tag{27}$$

For $l = l_0$ we simply define

$$\psi_{(l_0,t)} := \phi^{(l_0,t)} \ , \quad \text{for} \ \ t \in T(l_0) := S(l_0) \ . \tag{28}$$

For the scaling relations we use the same notation as in (19), but now, the matrices $H^l$, $\tilde{H}^l$, $G^l$, $\tilde{G}^l$ are finite, e.g. $H^l$ is a $\#S(l) \times \#S(l-1)$ matrix, etc. These matrices have essentially the same banded structure as their counterparts from subsection 2.2, but at the upper left and lower right corners there are blocks which contain the scaling relation coefficients for the boundary modified scaling functions and wavelets, respectively.

Periodic boundary conditions are even simpler. Here, we set

$$\phi^{(l,s)}(z) = \phi^{(l,s)}(x) \ , \quad \text{where } z := \begin{cases} x & : 0 \le x \le 1 \\ 1+x & : x < 0 \\ x-1 & : x > 1 \end{cases}$$

for $s \in S(l) = \{0,..,2^l-1\}$. The wavelets are $\psi^{(l,2t+1)}$ for $t \in T(l) := \{0,..,2^{l-1}-1\}$. Thus, the scaling matrices have a simple periodic structure. For example for $H^l$ we have

$$(H^l)_{(r,s)} = h_{mod(r-2s,2^l)} \ . \tag{29}$$

## 3 Adaptive Algorithms

In this section we will introduce algorithms for the adaptive wavelet transform, its inverse transform and the adaptive evaluation of constant coefficient linear differential operators. In section 4 we will see that the simple tensor product application of these univariate algorithms is enough to handle also the multivariate case.

Now assume that a function $u \in V^L$ is given in its wavelet representation

$$u = \sum_{l=l_0}^{L} \sum_{t \in T(l)} u_{(l,t)} \psi_{(l,t)} = \sum_{l=l_0}^{L} \boldsymbol{\Psi}_l \cdot \mathbf{u}_l \ .$$

We already noted that the coefficients $u_{(l,t)}$ decay fast where $u$ is locally smooth and decay less fast where $u$ is locally less smooth. Thus, it may be advantageous to discard all coefficients with an absolute value smaller than some prescribed $\epsilon$ and to work with the resulting adapted and compressed function approximation instead. This motivates to use certain subsets of the wavelet basis $\bigcup_{l=l_0}^{L} \boldsymbol{\Psi}_l$ for function representation. Analogously, the wavelet transform and its inverse can be generalized to the adaptive case. Also, further operations with functions like the addition, the multiplication of two functions or more general, the non-linear function evaluation can be considered directly for the adaptive setting. Furthermore, the discretization and application of a differential operator to a function can be treated in an adaptive fashion.

To this end, recall that $T(l)$ denotes the index set of the wavelets of level $l$ and that $S(l)$ denotes the index set for the scaling functions. Now, an adaptive basis out of $\bigcup_{l=l_0}^{L} \boldsymbol{\Psi}_l$ consists of

$$\bigcup_{l=l_0}^{L} \bigcup_{t \in \mathcal{T}(l)} \psi_{(l,t)} \quad \text{where} \quad \mathcal{T}(l) \subseteq T(l) \ . \tag{30}$$

Thus $\{\mathcal{T}(l)\}_{l=l-0}^{L}$ describes the set of active wavelet indices whose associated wavelets are 'active' in the adaptive wavelet basis. For the adaptive wavelet transforms we will also deal with subsets

$$\mathcal{S}(l) \subseteq S(l) \ , \text{ where in general } \ \mathcal{S}(l) \subset S(l) \ , \tag{31}$$

in order to handle adaptive versions of the auxiliary $\mathbf{u}^l$ vectors. Similar to subsection 2.2 we will use the following notation for row vectors of functions, column vectors of coefficient sequences and scalar products

$$\begin{aligned}
\mathbf{u}^{\mathcal{S}(l)} &:= \{u^{(l,s)}\}_{s \in \mathcal{S}(l)} \ , \quad \boldsymbol{\Phi}^{\mathcal{S}(l)} := \{\phi^{(l,s)}\}_{s \in \mathcal{S}(l)} \ , \\
\mathbf{u}_{\mathcal{T}(l)} &:= \{u_{(l,t)}\}_{t \in \mathcal{T}(l)} \ , \quad \boldsymbol{\Psi}_{\mathcal{T}(l)} := \{\psi_{(l,t)}\}_{t \in \mathcal{T}(l)} \ , \\
\mathbf{u}_{\mathcal{T}} &:= \{\mathbf{u}_{\mathcal{T}(l)}\}_{l=l_0}^{L} \ , \quad \boldsymbol{\Psi}_{\mathcal{T}} := \{\boldsymbol{\Psi}_{\mathcal{T}(l)}\}_{l=l_0}^{L} \\
\text{and} \quad & \boldsymbol{\Psi}_{\mathcal{T}} \cdot \mathbf{u}_{\mathcal{T}} = \sum_{(l,t) \in \mathcal{T}} u_{(l,t)} \psi_{(l,t)} \ .
\end{aligned} \tag{32}$$

In this section we will also use submatrices of $H^l$, $\tilde{H}^l$, $G^l$ and $\tilde{G}^l$. E.g. for $\mathcal{S}(l) \subseteq S(l)$ and $\mathcal{T}(l) \subseteq T(l)$ we define

$$G^l_{\mathcal{S}(l),\mathcal{T}(l)} := \left(G^l_{s,t}\right)_{s\in\mathcal{S}(l),t\in\mathcal{T}(l)} \ .$$

These submatrices will be used for the algebraic description of the adaptive wavelet transforms where we will work only with the sparse vectors $\mathbf{u}_{\mathcal{T}(l)}$ or $\mathbf{u}^{\mathcal{S}(l)}$ instead of $\mathbf{u}_l$ or $\mathbf{u}^l$, respectively.

### 3.1 Adaptive Wavelet Transform

In the following we will generalize the wavelet transform of the previous section and its inverse to the adaptive setting. Let us first describe the basic idea:

Each of our wavelets $\psi_{(l,t)}$ is also a scaling function $\phi^{(l,s(t))}$ with a certain $s(t)$ which is $s(t) = 2t+1$ (except for the case of homogeneous Neumann boundary conditions, see (27),(28)). Therefore, $\psi_{(l,t)}$ is one in $s(t)2^{-l}$ and vanishes in all other nodes of the dyadic grid $2^{-l}S(l)$. Thus, we can uniquely identify each of the wavelets with its associated point $s(t)2^{-l}$. Consequently, the adaptive grids

$$\Omega_l := \bigcup_{k=l_0}^{l} \bigcup_{t\in\mathcal{T}(l)} s(t)2^{-k} \ , \quad l_0 \le l \le L$$

are associated to our adaptive basis (30). To calculate the coefficient $u_{(l,t)}$ like in (15) we needed the nodal values of $u$ in $s(t)2^{-l}$ and the surrounding even nodes $(s(t)+\{-N+1,-N+3,...,N-1\})2^{-l}$. Thus, these nodes have to belong to $\Omega_{l-1}$ and therefore the corresponding wavelets $\psi_{(l',t')}$ $(l' \le l-1)$ have to belong to the adaptive basis. If this condition is satisfied we can run the adaptive versions of the algorithms **(WT)** and **(IWT)** by just restricting them to the index sets $\mathcal{T}(l) \subseteq T(l)$ on the wavelet side, and yet to be properly defined $\mathcal{S}(l) \subseteq S(l)$ on the scaling function side, respectively.

In order to formalize this approach it is convenient to introduce the concept of generating systems. Furthermore we need general operators $R$ and $C$ which act on matrices and index sets. Let $X, Y$ be index sets and $A$ be a $X \times Y$ matrix $A = \{a_{x,y}\}_{x\in X, y\in Y}$. Recall that the set of subsets of $X$ is denoted by $\mathcal{P}X$. Then we define

$$\begin{aligned} &R(A,.) : \mathcal{P}X \to \mathcal{P}Y \ , R(A,M) := \{y \in Y \mid \exists x \in M :\ a_{x,y} \neq 0\}, \text{'}R \cong \text{rows'}, \\ &C(A,.) : \mathcal{P}Y \to \mathcal{P}X \ , C(A,M) := R(A^T, M)\,, \quad \text{'}C \cong \text{ columns'}. \end{aligned} \tag{33}$$

The operators $R$ and $C$ will be used to determine dependencies of 'active' wavelets $\psi_{(l,t)}$ on scaling functions $\phi^{(l,s)}$ and conversely.
Consider the following typical step of the wavelet transform **(WT)**

$$u_{(l,t)} = \left(\tilde{G}^l_{.,t}\right)^T \cdot \mathbf{u}^l \ . \tag{34}$$

In the adaptive case we are interested in a few coefficients $u_{(l,t)}$ only (more precisely: $t \in \mathcal{T}(l)$). In this case we can not effort to store/calculate the complete vector $\mathbf{u}^l$, since this would destroy the advantages of adaptivity. Now, the problem is which of the coefficients $u^{(l,s)}$ are really required to calculate the above term (34). The answer is given by the matrix $\tilde{G}^l$: We need $u^{(l,s)}$ for all $s \in C(\tilde{G}^l, \{t\})$. so-called generating system.

The main idea of so-called generating systems is that an *incomplete* representation of $u = \sum_l \boldsymbol{\Psi}_{\mathcal{T}(l)} \cdot \mathbf{u}_{\mathcal{T}(l)}$ in terms of a scaling functions basis $\boldsymbol{\Phi}^{\mathcal{S}(l)} \cdot \langle \tilde{\boldsymbol{\Phi}}^{\mathcal{S}(l)},\ u\rangle$ may be *locally* identical to $u$ if $l$ is *locally* the finest level. This can be exploited to evaluate non-linear functionals or even differential operators.

A generating system for $\bigcup_{l=l_0}^{L} \boldsymbol{\Psi}_l$ is a set

$$G := \bigcup_{l=l_0}^{L} \boldsymbol{\Phi}^{\mathcal{S}(l)}\ , \quad \mathcal{S}(l) \subseteq S(l), \quad \text{such that} \quad \operatorname{span} \bigcup_{l=l_0}^{L} \boldsymbol{\Psi}_l \subseteq \operatorname{span} G\ .$$

The index sets $\mathcal{S}(l)$ shall be as small as possible (numerical efficiency) and they shall allow for the following adaptive wavelet transform:

**algorithm (AWT)**
given $\{\mathbf{u}^{\mathcal{S}(l)}\}_{l=l_0}^{L}$
`for` $l = L$ `downto` $l_0 + 1$

$$\begin{aligned}
\mathbf{u}_{\mathcal{T}(l)} &= \left(\tilde{G}^l_{\mathcal{S}(l),\mathcal{T}(l)}\right)^T \cdot \mathbf{u}^{\mathcal{S}(l)} \\
\Delta(l-1) &= \{t \in \mathcal{S}(l-1) \mid C(\tilde{H}^l, \{t\}) \subseteq \mathcal{S}(l)\} \\
\mathbf{u}^{\Delta(l-1)} &= \left(\tilde{H}^l_{\mathcal{S}(l),\Delta(l-1)}\right)^T \cdot \mathbf{u}^{\mathcal{S}(l)}
\end{aligned}$$

`end`

The idea of the $\Delta(l-1)$ correction is the following: Assume that the values of $\mathbf{u}^{\mathcal{S}(l)}$ are only approximations of the true values $\langle \tilde{\boldsymbol{\Phi}}^{\mathcal{S}(l)}\ ,\ u\rangle$. Then, it is likely that the values of $\mathbf{u}^{\mathcal{S}(l)}$ are more accurate for large $l$ than for small $l$. Thus, the $\Delta(l-1)$ correction uses the more accurate values of $\mathbf{u}^{\mathcal{S}(l)}$ to calculate an improved approximation of $\mathbf{u}^{\mathcal{S}(l-1)}$. From **(AWT)** we immediately obtain the condition

$$C(\tilde{G}^l, \mathcal{T}(l)) \quad \subseteq \quad \mathcal{S}(l)\ . \tag{35}$$

In the same fashion we carry the inverse wavelet transform over to the adaptive case:

```
algorithm (AIWT)
given {u_T(l)}_{l=l0}^L
for l = l0 + 1 to L
    u^S(l) = G^l_{S(l),T(l)} · u_T(l)  +  H^l_{S(l),S(l-1)} · u^S(l-1)
end
```

This algorithm shall give the same results $u^{(l,s)}$ for $s \in \mathcal{S}(l)$ as the non-adaptive algorithm **(IWT)**. Thus, the second condition on $\mathcal{S}(l)$ is

$$R(H^l, \mathcal{S}(l)) \quad \subseteq \quad \mathcal{S}(l-1) \ . \tag{36}$$

Combining conditions (35) and (36) we end up with an algorithm for the recursive determination of $\mathcal{S}(l)$:

```
algorithm (S)
given {T(l)}_{l=l0}^L
S(L) = C(G̃^L, T(L))
for l = L - 1 downto l0
    S(l) = C(G̃^l, T(l))  ∪  R(H^{l+1}, S(l+1))
end
```

A short calculation shows that the amount of operations involved in **(AWT)**, **(AIWT)** and **($\mathcal{S}$)** is proportional to the number of active wavelets.

### 3.2 Adaptive Non-Linear Function Evaluation

The three above algorithms are not restricted to just Interpolet wavelets but can be used in connection with general biorthogonal wavelets to solve a difficult problem in adaptive wavelet methods: the evaluation of non-linear terms. In the following we will briefly sketch the main idea of the associated algorithm.

Assume that we have a quite accurate adaptive wavelet approximation $\boldsymbol{\Psi}_{\mathcal{T}} \cdot \mathbf{u}_{\mathcal{T}}$ of a function $u$. Now we are looking for a wavelet approximation $\boldsymbol{\Psi}_{\mathcal{T}} \cdot \mathbf{v}_{\mathcal{T}}$ of $f(u(x))$, where $f: \mathbb{R} \to \mathbb{R}$ is a given smooth function. Thus, the coefficients

$$v_{(l,t)} \ \approx \ \langle \tilde{\psi}_{(l,t)} \ , \ f \circ u \rangle$$

are needed. The main idea for the calculation of these terms is to use the generating system representation of $\boldsymbol{\Psi}_{\mathcal{T}} \cdot \mathbf{u}_{\mathcal{T}}$. To this end, first the corresponding coefficients $\mathbf{u}^{\mathcal{S}(l)}$ are calculated by algorithm **(AIWT)**. These coefficients are approximations of $\langle \tilde{\phi}^{(l,s)} \ , \ u \rangle$

$$u^{(l,s)} \approx \langle \tilde{\phi}^{(l,s)} \ , \ u \rangle \ , \quad s \in S(l) \ . \tag{37}$$

All commonly used scaling functions have certain (quasi-) interpolation properties. More precisely, there is a $K \in \mathbb{N}$ such that for all $s \in S(l)$ there is a point $x(l,s)$ (in general $x(l,s) \approx s2^{-l}$) which satisfies

$$\langle \tilde{\phi}^{(l,s)} \ , \ u \rangle = u(x(l,s)) \ + \ O(2^{-lK} \frac{\partial^K u}{\partial x^K}) \ . \tag{38}$$

For the orthogonal Daubechies wavelets $K = \min(3, N)$, compare [61]. For Coiflets $K$ also depends on $N$ and is usually $\geq N$, see [51]. For Interpolets there is no restriction on $K$, since $\langle \tilde{\phi}^{(l,s)} \ , \ u \rangle = u(2^{-l}s)$. The above relation (38) applied to $f \circ u$ yields

$$\begin{aligned}
f(u^{(l,s)}) \quad & \overset{(37)}{\approx} \quad f\big(\langle \tilde{\phi}^{(l,s)} \ , \ u \rangle\big) \\
& \overset{(38)}{=} \quad f\big(u(x(l,s)) \ + \ O(2^{-lK} \frac{\partial^K u}{\partial x^K})\big) \\
& \overset{f \in C^1}{=} \quad f\big(u(x(l,s))\big) \ + \ O(2^{-lK} \frac{\partial^K u}{\partial x^K}) \\
& \overset{(38) \text{ for } f\circ u}{=} \langle \tilde{\phi}^{(l,s)} \ , \ f \circ u \rangle \ - \ O\big(2^{-lK} \frac{\partial^K f \circ u}{\partial x^K}\big) \ + \ O\big(2^{-lK} \frac{\partial^K u}{\partial x^K}\big) \ .
\end{aligned}$$

Therefore, if $f$ is sufficiently smooth there holds

$$f(u^{(l,s)}) \approx \langle \tilde{\phi}^{(l,s)} \ , \ f \circ u \rangle \ + \ O\big(2^{-lK} \|u\|_{C^K(\mathrm{supp}\tilde{\phi}^{(l,s)})}\big) \tag{39}$$

where

$$\|u\|_{C^K(\mathrm{supp}\tilde{\phi}^{(l,s)})} := \sum_{i=0}^{K} \sup\{|\frac{\partial^i u(x)}{\partial x^i}| \mid x \in \mathrm{supp}\tilde{\phi}^{(l,s)}\} \ .$$

If we would have sufficiently accurate approximations $v^{(l,s)}$ of the values $\langle \tilde{\phi}^{(l,s)} \ , \ f \circ u \rangle$, algorithm **(AWT)** applied to $\{v^{(l,s)}\}$ would calculate sufficiently accurate approximations $\mathbf{v}_{\mathcal{T}(l)}$.

Two cases are possible in (39). First, the $O(2^{-lK}\|u\|_{C^K})$-term is sufficiently small, i.e. it is of the same order as the error $u^{(l,s)} - \langle \tilde{\phi}^{(l,s)} \ , \ u \rangle$, see (37). Then, $f(u^{(l,s)})$ is the desired approximation $v^{(l,s)}$. In the other case the $O(2^{-lK}\|u\|_{C^K})$-term is not sufficiently small. But then, adaptive refinement in the vicinity of $\mathrm{supp}\tilde{\phi}^{(l,s)}$ was already needed in the wavelet approximation process to obtain $\boldsymbol{\Psi}_{\mathcal{T}} \cdot \mathbf{u}_{\mathcal{T}}$ with sufficient accuracy. Therefore, there are active wavelets in the vicinity of $\mathrm{supp}\tilde{\phi}^{(l,s)}$ on higher levels than $l$ in $\boldsymbol{\Psi}_{\mathcal{T}}$. Thus, we may assume that $s \in \Delta(l)$, compare algorithm **(AWT)**. Consequently, the approximation $v^{(l,s)}$ of $\langle \tilde{\phi}^{(l,s)} \ , \ f \circ u \rangle$ is not necessarily to be calculated by $f(u^{(l,s)})$, but can be calculated by the wavelet transform from sufficiently accurate values of higher levels. Now, the complete algorithm reads

| **algorithm (NL)** |
|---|
| given $\mathbf{u}_{\mathcal{T}}$ |
| Calculate $\{\mathbf{u}^{\mathcal{S}(l)}\}_{l=l_0}^{L}$ by means of **(AIWT)** |
| Calculate $\{\mathbf{v}^{\mathcal{S}(l)}\}_{l=l_0}^{L}$ by means of $v^{(l,s)} = f(u^{(l,s)})$ |
| Calculate $\mathbf{v}_{\mathcal{T}}$ by means of **(AWT)** from $\{\mathbf{v}^{\mathcal{S}(l)}\}_{l=l_0}^{L}$ |

An analysis of this algorithm for general biorthogonal wavelets is given in the paper [18].

Note that for Interpolets there holds

$$u^{(l,s)} \;=\; \langle \tilde{\phi}^{(l,s)} \;,\; u\rangle$$

and thus

$$f\big(u^{(l,s)}\big) \;=\; \langle \tilde{\phi}^{(l,s)} \;,\; f\circ u\rangle \;.$$

Therefore, the results $v_{(l,t)}$ of algorithm **(AWT)** in algorithm **(NL)** satisfy

$$v_{(l,t)} \;=\; \langle \tilde{\psi}_{(l,t)} \;,\; f\circ u\rangle$$

without any error. Thus, for Interpolets the exact values are obtained while for general biorthogonal wavelets one can only expect approximate values.

We will now introduce a further simplification for the Interpolet case which deals with the following problem: In general the number of degrees of freedom of the adaptive generating system is larger than the number of degrees of freedom of the adaptive Interpolet basis. But, if both numbers would be the same then we could carry out the multivariate analogue of **(AWT)** and **(AIWT)** 'in place' by the successive application of **(AWT)** or **(AIWT)** with respect to each coordinate direction. Now, consider the special properties of the generating system representation for the Interpolet case. Here, many of the coefficients $u^{(l,s)}$ from **(AIWT)** are the same as the nodal value $u(s2^{-l})$. Using this interpretation ($u^{(l,s)} = u(s2^{-l})$) we can consider the adaptive grid

$$\Omega_{\mathcal{T}} := \bigcup_{l} \bigcup_{s\in\mathcal{S}(l)} s2^{-l}$$

as the minimal grid required to run the adaptive transforms, if we interpret **(AWT)** and **(AIWT)** in terms of nodal values and grid points. However, in general

$$\#\Omega_{\mathcal{T}} \;>\; \#\mathcal{T} \qquad \text{where} \qquad \mathcal{T} := \bigcup_{l=l_0}^{L} \bigcup_{t\in\mathcal{T}(l)} (l,t) \;.$$

The point is that under some mild conditions on $\mathcal{T}$ there holds $\#\Omega_{\mathcal{T}} \;=\; \#\mathcal{T}$ and $\Omega_{\mathcal{T}}$ has those nodes just at the right places which are required to run **(AWT)** and **(AIWT)**.

**Lemma 1.** *Let*

$$\begin{aligned} &\sigma(l,t) := s \qquad \textit{if} \quad \phi^{(l,s)} = \psi_{(l,t)} \\ &x(l,t) := \sigma(l,t)2^{-l} \end{aligned}$$

*be the index of the unique scaling function which coincides with $\psi_{(l,t)}$ and the corresponding unique grid point. For $l = l_0$ there holds $x(l_0,t) = t2^{-l_0}$ and for $l > l_0$ there holds $x(l,t) = (2t+1)2^{-l}$ except for homogeneous Neumann b.c., see (27), (28). Since the grid point is unique one can define the inverse mapping*

$$x^{-1} \;:\; \bigcup_{l=l_0}^{L} 2^{-l}S(l) \;\;\to\;\; \bigcup_{l=l_0}^{L} (l,T(l)) \;, \qquad x^{-1}(x(l,t)) = (l,t) \;.$$

*Now, if*

$$(l,t) \in \mathcal{T} \quad \Rightarrow \quad x^{-1}\big(\ 2^{-l} C(\tilde{G}^l, \{t\})\ \big) \subseteq \mathcal{T} \tag{40}$$

*then there holds*

$$\mathcal{T} \quad = \quad x^{-1}\big(\Omega_{\mathcal{T}}\big)\ . \tag{41}$$

*Proof.* We will first show that $\mathcal{T} \subseteq x^{-1}\big(\Omega_{\mathcal{T}}\big)$. For Interpolets one has

$$\sigma(l,t) \ \in\ C(\tilde{G}^l, \{t\})\ . \tag{42}$$

In the periodic or shift invariant case there holds

$$\tilde{G}^l_{t,2t+1} = \tilde{g}_{(2t+1)-2t} = \tilde{g}_1 = h_0 = 1 \quad \Rightarrow \quad 2t+1 = \sigma(l,t) \in C(\tilde{G}^l, \{t\})\ .$$

Now (42) yields

$$x(l,t) \ \in\ 2^{-l} C(\tilde{G}^l, \{t\}) \quad \Rightarrow \quad (l,t) \ \in\ x^{-1}\big(\ 2^{-l} C(\tilde{G}^l, \{t\})\ \big) \subset \Omega_{\mathcal{T}}\ .$$

The proof for $x^{-1}\big(\Omega_{\mathcal{T}}\big) \subseteq \mathcal{T}$ goes by induction for $l$ of $(l,t) \in x^{-1}\big(\Omega_{\mathcal{T}}\big)$. Let $(L,t) \in x^{-1}\big(\Omega_{\mathcal{T}}\big)$. We use the following technical property of the matrices $\tilde{G}^l$

$$C(\tilde{G}^l, \{t\}) \bigcap \sigma(T(l)) \quad = \quad \{\sigma(l,t)\}\ . \tag{43}$$

For the inner scaling functions or for the periodic case this means that there is exactly one non-zero entry $\tilde{G}^l_{t,s}$ with odd s. This $s$ is exactly $s = 2t+1$, see (17) and (2). For the boundary Interpolets, (43) follows from their special construction and a from longer technical calculation which we omit here. From (43) one concludes

$$(L,t) \in x^{-1}\big(\Omega_{\mathcal{T}}\big) \quad \Rightarrow \quad \sigma(L,t) \in \mathcal{S}(L) = C(\tilde{G}^L, \mathcal{T}(L)) \quad \overset{(43)}{\Rightarrow} \quad t \in \mathcal{T}(L)\ .$$

Assume now that we have shown

$$\forall (k,r) \ \in\ x^{-1}\big(\Omega_{\mathcal{T}}\big) \qquad k > l \quad \Rightarrow \quad (k,r) \in \mathcal{T}\ .$$

Let $(l,t) \in x^{-1}\big(\Omega_{\mathcal{T}}\big)$ and define

$$l' := \max\{k \mid x(l,t) \in 2^{-k}\mathcal{S}(k)\}\ , \quad s := 2^{l'} x(l,t)\ . \tag{44}$$

Two cases are possible:

(i) $s \in C(\tilde{G}^{l'}, \{t'\})$ for some $t' \in \mathcal{T}(l')$. Then

$$(l,t) \in x^{-1}\big(\ 2^{-l'} C(\tilde{G}^{l'}, \{t'\})\ \big) \overset{(40)}{\subseteq} \mathcal{T}\ .$$

(ii) $s \in R(H^{l'}, \mathcal{S}(l'+1)) \backslash C(\tilde{G}^{l'}, \mathcal{T}(l'))$, i.e.

$$\exists\ s' \in \mathcal{S}(l'+1) \quad : \quad s \in R(H^{l'+1}, \{s'\})\ .$$

$s'$ must be odd. Because of $\tilde{H}^k = \{\delta(n-2m)\}_{m \in S(k-1), n \in S(k)}$ and $\big(\tilde{H}^k\big)^T \cdot H^k \overset{(20)}{=} I$, each even row of $H^k$ contains exactly one non-zero: $H^k_{2m,m} = 1$. Thus, if $s'$ is even, then $R(H^{l'+1}, \{s'\}) = \{s\}$ and $s' = 2s$ which contradicts the definition (44) of $l'$, $s$. But, if $s'$ is odd, then there is $t' \in \mathcal{T}(l'+1)$ with $\sigma(l'+1,t') = s'$. Since $l'+1 > l$ the assumption holds and $(l'+1,t') \in \mathcal{T}$. A close look on the matrices $\tilde{G}^k$ shows that $(l'+1, 2s) \in C(\tilde{G}^{l'+1}, \{t'\})$. This proves the assertion. □

We have now shown that the technical condition (40) results in the relation (41) which allows to carry out the algorithms **(AWT)** and **(AIWT)** *in place* by using the same data structure and the same storage for $\mathbf{u}_\mathcal{T}$ and $\{\mathbf{u}^{\mathcal{S}(l)}\}_{l=l_0}^L$ in a numerical code. Furthermore, (40) and (41) imply that not only the exact values $\langle\tilde{\psi}_{(l,t)}\ ,\ f \circ u\rangle$ are calculated by the algorithm **(NL)**, but that also the result $\boldsymbol{\Psi}_\mathcal{T} \cdot \mathbf{v}_\mathcal{T}$ coincides with $f \circ u$ on $\Omega_\mathcal{T}$:

$$\big(\boldsymbol{\Psi}_\mathcal{T} \cdot \mathbf{v}_\mathcal{T}\big)(\Omega_\mathcal{T}) = (f \circ u)(\Omega_\mathcal{T}) \ . \tag{45}$$

If (40) is not satisfied (45) does not hold in general. Nevertheless, the question how to fulfill (40) remains. The idea is now to enlarge slightly a given index set $\mathcal{T}^0$ which does not fulfill (40) a priori. To this end, the following recursive algorithm calculates from $\mathcal{T}^0$ an index set $\mathcal{T}$ which satisfies (40).

**algorithm** $(\mathcal{T})$
$\mathcal{T}(L) = \mathcal{T}^0(L)$
`for` $l = L$ `downto` $l_0 + 1$
  $\mathcal{T}(l-1) = \mathcal{T}^0(l-1) \ \bigcup \ \{t \in T(l-1) \mid 2\sigma(l-1,t) \in C(\tilde{G}^l, \mathcal{T}(l))\}$
`end`

**Remark:** In [64] another algorithm was introduced which does not need (40). E.g. for the adaptive wavelet transform one starts with the nodal values $u(\Omega_\mathcal{T})$ and tries to find the interpolant $\sum_l \boldsymbol{\Psi}_{\mathcal{T}(l)} \cdot \mathbf{u}_{\mathcal{T}(l)}$ such that

$$\Big(\sum_{l=l_0}^{L} \boldsymbol{\Psi}_{\mathcal{T}(l)} \cdot \mathbf{u}_{\mathcal{T}(l)}\Big)(\Omega_\mathcal{T}) = u(\Omega_\mathcal{T}) \ .$$

To this end one exploits the fact that except for $\psi_{(l_0,t)}$ all other wavelets vanish at $x(l_0,t)$. Therefore $u_{(l_0,t)}$ and thus, $\mathbf{u}_{\mathcal{T}(l_0)}$ are easy to obtain. Then, in a second step one considers the interpolation problem

$$\Big(\sum_{l=l_0+1}^{L} \boldsymbol{\Psi}_{\mathcal{T}(l)} \cdot \mathbf{u}_{\mathcal{T}(l)}\Big)(\Omega_\mathcal{T}) = \big(u - \boldsymbol{\Psi}_{\mathcal{T}(l_0)} \cdot \mathbf{u}_{\mathcal{T}(l_0)}\big)(\Omega_\mathcal{T}) \ ,$$

which allows to determine $\mathbf{u}_{\mathcal{T}(l_0+1)}$ in the same fashion as $\mathbf{u}_{\mathcal{T}(l_0)}$. This scheme is repeated for all levels from $l = l_0 + 1$ to $L$. One drawback of this approach is that we have to know and handle the values of all Interpolets on all grid points of the finest level employed. Furthermore, at least in the non-adaptive case this algorithm has a complexity of $\mathcal{O}(L2^L)$ whereas our above approach is of order $\mathcal{O}(2^L)$ only. On the other hand, there is no artificial fill-in needed to ensure (40). But anyway, our experience is that this fill-in is relatively small in practical situations.

### 3.3 Adaptive Evaluation of Differential Operators

We will now derive an algorithm for the fast evaluation of constant coefficient differential operators. As an example consider the operator $\partial_{xx}$. For the discretization a Petrov-Galerkin scheme is used where the dual wavelets are chosen as test functions. We then obtain

$$\langle \tilde{\psi}_{(l,t)} \ , \ \partial_{xx} u \rangle \qquad \forall (l,t) \in \mathcal{T} \ . \tag{46}$$

Of course the trial functions $\psi_{(l,t)}$ must be at least two times continuously differentiable such that the above expression makes sense. The smoothness of the Interpolets grows with increasing $N$. E.g. for $N = 4$ we have $\psi_{(l,t)} \in C^{1.617}$ and for $N = 6$ we have $\psi_{(l,t)} \in C^{2.255}$. Now, we test $\partial_{xx} u$ not by just one function $\tilde{\psi}_{(l,t)}$ as in (46), but by all test functions of one level $\mathcal{T}(l)$ of the adaptive basis. We then obtain

$$\begin{aligned} \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}(l)} \ , \ \partial_{xx} u \rangle &= \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}(l)} \ , \ \partial_{xx} \sum_{k=l_0}^{L} \boldsymbol{\Psi}_{\mathcal{T}(k)} \cdot \mathbf{u}_{\mathcal{T}(k)} \rangle \\ &= \quad \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}(l)} \ , \ \partial_{xx} \sum_{k \geq l} \boldsymbol{\Psi}_{\mathcal{T}(k)} \cdot \mathbf{u}_{\mathcal{T}(k)} \rangle \\ &\quad + \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}(l)} \ , \ \partial_{xx} \sum_{k < l} \boldsymbol{\Psi}_{\mathcal{T}(k)} \cdot \mathbf{u}_{\mathcal{T}(k)} \rangle \\ &=: \quad \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}(l)} \ , \ \partial_{xx} a_l \rangle \ + \ \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}(l)} \ , \ \partial_{xx} b_{l-1} \rangle \ . \end{aligned} \tag{47}$$

Now, the idea is to use generating systems for the calculations of the two terms in (47). For the second term in (47) we use an incomplete representation of $b_{l-1}$ :

$$\boldsymbol{\Phi}^{\mathcal{S}^b(l-1)} \cdot \mathbf{b}^{\mathcal{S}^b(l-1)} \ .$$

Here, $\mathcal{S}^b(l-1)$ must be large enough to ensure that

$$\langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}(l)} \ , \ \partial_{xx} \boldsymbol{\Phi}^{\mathcal{S}^b(l-1)} \rangle \cdot \mathbf{b}^{\mathcal{S}^b(l-1)} \ \stackrel{!}{=} \ \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}(l)} \ , \ \partial_{xx} b_{l-1} \rangle \ .$$

If we define $WS^l := \langle \tilde{\boldsymbol{\Psi}}_l \ , \ \partial_{xx} \boldsymbol{\Phi}^l \rangle$ this leads to the condition

$$R(WS^l, \mathcal{T}(l)) \quad \subseteq \quad \mathcal{S}^b(l-1)$$

and finally to the following algorithm:

**algorithm ($\mathcal{S}^b$)**
given $\{\mathcal{T}(l)\}_{l=l_0}^{L}$
$\mathcal{S}^b(L) = \emptyset$
`for` $l = L-1$ `downto` $l_0$
  $\mathcal{S}^b(l) \ = \ R(WS^l, \mathcal{T}(l)) \quad \bigcup \quad R(H^{l+1}, \mathcal{S}^b(l+1))$
`end`

The values $\mathbf{b}^{\mathcal{S}^b(l-1)}$ are calculated by algorithm **(AIWT)** with $\mathcal{S}(k)$ replaced by $\mathcal{S}^b(k)$.

A similar trick is applied to the first term in (47). Instead of a direct calculation of $\langle\tilde{\boldsymbol{\Psi}}_{\mathcal{T}(l)}\ ,\ \partial_{xx}a_l\rangle$ we there compute the auxiliary values

$$\mathbf{c}^{\mathcal{S}^a(l)} \ := \ \langle\tilde{\boldsymbol{\Psi}}^{\mathcal{S}^a(l)},\ \partial_{xx}a_l\rangle\ ,$$

by means of

**algorithm (AUX)**
given $\{\mathbf{u}_{\mathcal{T}(l)}\}_{l=l_0}^{L}$ and $\{\mathcal{S}^a(l)\}_{l=l_0}^{L}$
$\mathbf{c}^{\mathcal{S}^a(L)} = SW^L_{\mathcal{S}^a(L),\mathcal{T}(L)} \cdot \mathbf{u}_{\mathcal{T}(L)}$
`for` $l = L-1$ `downto` $l_0$
$\quad \mathbf{c}^{\mathcal{S}^a(l)} \ = \ \left(\tilde{H}^{l+1}_{\mathcal{S}^a(l+1),\mathcal{S}^a(l)}\right)^T \cdot \mathbf{c}^{\mathcal{S}^a(l+1)} \ + \ SW^l_{\mathcal{T}(l),\mathcal{S}^a(l)} \cdot \mathbf{u}_{\mathcal{T}(l)}$
`end`

where $SW^l := \langle\tilde{\boldsymbol{\Phi}}^l,\ \partial_{xx}\boldsymbol{\Psi}_l\rangle$. Here, the index sets $\mathcal{S}^a$ should be large enough to ensure that

$$\langle\tilde{\boldsymbol{\Psi}}_{\mathcal{T}(l)}\ ,\ \partial_{xx}a_l\rangle \ \stackrel{!}{=} \ \left(\tilde{G}_{\mathcal{S}^a(l),\mathcal{T}(l)}\right)^T \cdot \mathbf{c}^{\mathcal{S}^a(l)}\ .$$

Because of the regularity condition (40) on $\mathcal{T}$, the supports of $\boldsymbol{\Psi}_{\mathcal{T}(l)}$ fulfill $\mathrm{supp}\boldsymbol{\Psi}_{\mathcal{T}(l+1)} \subseteq \mathrm{supp}\boldsymbol{\Psi}_{\mathcal{T}(l)}$. Therefore

$$\mathcal{S}^a(l) \ = \ R(SW^l, \mathcal{T}(l))\ . \tag{48}$$

### 3.4 Adaptive Evaluation of Finite Difference Operators

Note that there is also another way for the adaptive evaluation of differential operators which is based on finite differences (FD). An advantage of FD is that there is no smoothness requirement for the Interpolets. Even the usual piecewise linear hat function ($N = 2$) which is only continuous can be used. The FD algorithm is similar to the algorithm **(NL)** for non-linear terms:

**algorithm (FD)**
given $\{\mathbf{u}_{\mathcal{T}(l)}\}$
Run **(AIWT)** to obtain the generating system representation $\{\mathbf{u}^{\mathcal{S}(l)}\}$
Apply the FD stencil (e.g. $2^{-l}[1\ \ -2\ \ \ 1]$) to $\mathbf{u}^{\mathcal{S}(l)}$ and obtain $\mathbf{v}^{\mathcal{S}(l)}$
Run **(AWT)** of $\{\mathbf{v}^{\mathcal{S}(l)}\}$ to obtain the final result

Here, one has to take into account that not for all points on a level $l$ we can apply the FD stencil. For example, if $\mathcal{S}(l) = \{4, ..., 10\}$ a three point stencil like $[1\ -2\ \ 1]$ can only be used for $\{5, ..., 9\}$. Therefore, the nodal values $\{4, 10\}$ must not be employed in the $\Delta$-correction in algorithm **(AWT)**. However, note that the condition (35) ensures that the index set $\mathcal{S}(l)$ is large enough to allow for the application of the FD stencil to sufficiently many points of level $l$.

A similar approach has been used in [33] and [56]. In connection with the classical MRA-approach and higher order Interpolets FD have been employed in [40].

## 4 Multivariate Wavelets

So far, we are able to deal with the adaptive wavelet transform, the non-linear function evaluation and the evaluation of a differential operator in the 1D case on the interval. Now, we generalize the wavelet construction to the multivariate setting. To this end we consider two different tensor product-type multivariate wavelet approaches.

### 4.1 The MRA-Approach

This approach is used almost exclusively in the wavelet literature, see e.g. [2,5,10,18,29,48,59,64] and the references cited therein. The key idea is to establish an ordered sequence of approximation spaces like in the one-dimensional case, see (10). The simplest construction of such multivariate approximation spaces are the tensor products of the univariate $V^l$, i.e. $V^l \otimes ... \otimes V^l$. The complementary spaces between $V^{l-1} \otimes ... \otimes V^{l-1}$ and $V^l \otimes ... \otimes V^l$ are defined by tensor products of the univariate spaces $V^{l-1}$ and $W_l$. For example in the two-dimensional case we obtain

$$
\begin{aligned}
V^l \otimes V^l &= \left(V^{l-1} \oplus W_l\right) \otimes \left(V^{l-1} \oplus W_l\right) \\
&= V^{l-1} \otimes V^{l-1} \\
&\quad \oplus \left(V^{l-1} \otimes W_l\right) \oplus \left(W_l \otimes V^{l-1}\right) \oplus \left(W_l \otimes W_l\right) .
\end{aligned}
$$

Thus, the complementary space is the direct sum of three different tensor product spaces. Now, it is simple to define bases of the complementary spaces by tensor products of different combinations of univariate wavelets and/or scaling functions.

In the $d$-dimensional case this reads

$$
\psi_{(\mathbf{e},l,\mathbf{t})}(\mathbf{x}) := \psi_{(e_1,l,t_1)}(x_1) \cdot ... \cdot \psi_{(e_d,l,t_d)}(x_d) \ , \tag{49}
$$

where $\psi_{(0,l,t)}(x) := \phi^{(l-1,t)}(x)$ and $\psi_{(1,l,t)}(x) := \psi_{(l,t)}(x)$. Here $\mathbf{e} = (e_1, .., e_d) \in \{0,1\} \times .. \times \{0,1\} \backslash \mathbf{0} \subset \mathbb{N}^d$ defines the type of the wavelet, $l \geq l_0$ denotes the level and $\mathbf{t} = (t_1, .., t_d) \in \mathbb{Z}^d$ denotes the spatial translation. Because of the linear ordering of the approximation spaces $V^{l-1} \otimes .. \otimes V^{l-1} \subset V^l \otimes .. \otimes V^l$ all the algorithms of the previous section can be used here as well. Of course the matrices $H^l, G^l, ...$ have to be replaced by their counterparts for the multivariate case. Due to the tensor product construction all these matrices are essentially Kronecker products [41] (sometimes called tensor products) of the matrices $H^l, G^l, ....$ We will not employ this approach in the remainder of this paper, but we will make use of the following pure tensor product technique.

### 4.2 The MRA-$d$-Approach

For this approach simple tensor products of univariate wavelets are employed as trial and test functions, i.e.

$$
\psi_{(\mathbf{l},\mathbf{t})}(\mathbf{x}) := \psi_{(l_1,t_1)}(x_1) \cdot ... \cdot \psi_{(l_d,t_d)}(x_d) \ , \tag{50}
$$

where $\mathbf{x} = (x_1, .., x_d)$, $\mathbf{l}, \mathbf{t}$ analogous. In the same way the multivariate dual wavelets $\tilde{\psi}_{(\mathbf{l},\mathbf{t})}$ are defined. In the rest of this paper we exclusively use the MRA-$d$-approach. The reason for this choice is its algorithmical simplicity (as we will see below) and its superior approximation properties [34,62].

Analogously to (30) and (32) the following notation for the adaptive multivariate case is used:

$$\boldsymbol{\mathcal{T}}(\mathbf{l}) \subseteq T(l_1) \otimes ... \otimes T(l_d) \ , \quad \boldsymbol{\mathcal{T}} = \bigcup_{\mathbf{l}} \{\mathbf{l}\} \times \boldsymbol{\mathcal{T}}(\mathbf{l}) \ ,$$
$$\mathbf{u}_{\mathcal{T}(\mathbf{l})} = \{u_{(\mathbf{l},\mathbf{t})}\}_{\mathbf{t} \in \mathcal{T}(\mathbf{l})} \ , \quad \mathbf{u}_{\mathcal{T}} = \{\mathbf{u}_{\mathcal{T}(\mathbf{l})}\}_{\mathbf{l}} \ . \tag{51}$$

Furthermore, we will denote by

$$V_{\mathcal{T}} = \mathrm{span}\{\psi_{(\mathbf{l},\mathbf{t})}\}_{(\mathbf{l},\mathbf{t}) \in \mathcal{T}}, \quad \tilde{V}_{\mathcal{T}} = \mathrm{span}\{\tilde{\psi}_{(\mathbf{l},\mathbf{t})}\}_{(\mathbf{l},\mathbf{t}) \in \mathcal{T}}$$

the adaptive multivariate trial and test spaces.

The tensor product approach has the advantage that the algorithms for the adaptive wavelet transforms, the application of discrete differential operators and, to some extent, the non-linear function evaluation just boil down to the 1D algorithms of the previous section, compare also [14]. Consider as an example the evaluation of the partial differential operator $\partial_{x_1 x_1}$. We employ a Petrov-Galerkin scheme with the dual wavelets as test functions. Therefore, we have to evaluate the terms

$$\langle \tilde{\psi}_{(\mathbf{l},\mathbf{t})} \ , \ \partial_{x_1 x_1} \sum_{(\mathbf{k},\mathbf{s}) \in \mathcal{T}} u_{(\mathbf{k},\mathbf{s})} \psi_{(\mathbf{k},\mathbf{s})} \rangle =$$
$$\begin{aligned}
&= \sum_{(\mathbf{k},\mathbf{s}) \in \mathcal{T}} u_{(\mathbf{k},\mathbf{s})} \langle \tilde{\psi}_{(l_1,t_1)} \ , \ \partial_{x_1 x_1} \psi_{(k_1,s_1)} \rangle \cdot \prod_{i=2}^{d} \langle \tilde{\psi}_{(l_i,t_i)} \ , \ \psi_{(k_i,s_i)} \rangle \\
&\overset{(18)}{=} \sum_{(\mathbf{k},\mathbf{s}) \in \mathcal{T}} u_{(\mathbf{k},\mathbf{s})} \langle \tilde{\psi}_{(l_1,t_1)} \ , \ \partial_{x_1 x_1} \psi_{(k_1,s_1)} \rangle \cdot \prod_{i=2}^{d} \delta(k_i - l_i) \delta(s_i - t_i) \\
&= \sum_{k_1,s_1} u_{((k_1,\hat{\mathbf{l}}),(s_1,\hat{\mathbf{t}}))} \langle \tilde{\psi}_{(l_1,t_1)} \ , \ \partial_{x_1 x_1} \psi_{(k_1,s_1)} \rangle \\
&= \langle \tilde{\psi}_{(l_1,t_1)} \ , \ \partial_{x_1 x_1} \sum_{k_1,s_1} u_{((k_1,\hat{\mathbf{l}}),(s_1,\hat{\mathbf{t}}))} \psi_{(k_1,s_1)} \rangle \ ,
\end{aligned}$$

where $\hat{\mathbf{l}} = (l_2, .., l_d)$ and $\hat{\mathbf{t}} = (t_2, .., t_d)$. Thus, unidirectional operators like $\partial_{x_1 x_1}$ are evaluated simply by an application of the 1D algorithms of subsection 3.3 to all rows $(\hat{\mathbf{l}}, \hat{\mathbf{t}})$ of the adaptive basis. This also holds for the finite difference technique. Here the 1D algorithm **(FD)** is simply applied to all rows $(\hat{\mathbf{l}}, \hat{\mathbf{t}})$ of the adaptive basis.

In the same fashion the adaptive wavelet transform and its inverse can be carried over to the multivariate case. We simply apply **(AWT)** or **(AIWT)** first with respect to all rows in $x_1$-direction, then to all rows with respect to the $x_2$-direction and so on. It is important to note that only because of the special properties of Interpolets this 'tensor product type' application of e.g. the univariate algorithm **(AWT)** yields the exact solution of the interpolation problem with respect to the adaptive wavelet basis and the associated adaptive grid. This does not hold for other wavelets like the orthogonal Daubechies wavelets or biorthogonal spline wavelets. However, the multivariate version of condition (40) must be satisfied, i.e. (40) must hold for all rows with respect to the $x_i$-direction $(1 \leq i \leq d)$.

In summary, Interpolet wavelets in connection with the MRA-$d$-approach lead to particularly simple algorithms for the adaptive wavelet transform, its inverse, the non-linear function evaluation and the evaluation of unidirectional differential operators. All these algorithms boil down to the univariate adaptive algorithms of the previous section. In the same way as the non-linear function evaluation one can also find the approximate wavelet expansion of the product $u \cdot v$ of $u, v \in V_{\mathcal{T}}$. Furthermore, mixed derivatives like $\partial_{x_1 x_2} u$ are calculated simply by $\partial_{x_1}(\partial_{x_2} u)$. Altogether we have quite simple and efficient algorithms for the evaluation of all terms which may occur in the discretization of general PDEs. This is the first and most important step towards our final adaptive wavelet solver for PDEs or the Navier-Stokes equations, respectively.

Of course, the present approach is restricted to simple boxes, e.g. to $[0,1]^d$ as the computational domain. In order to deal with more general domains one can employ the usual parametric mappings where the computational domain is mapped to the physical domain. Another technique is to imbed the physical domain $\Omega$ in an enclosing box. Then, boundary conditions on $\partial\Omega$ are enforced by e.g. penalization techniques [47] or, in case of the Navier-Stokes equations, by special forcing terms on the right hand side [1].

## 5 Consistency and Stability

Now we will consider the convergence and the rate of convergence of our scheme. For the sake of simplicity we consider the model problem

$$\Delta u = f \ , \quad \text{in } \Omega = [0,1]^d \tag{52}$$

with appropriate boundary conditions. Our collocation scheme produces numerical solutions $u_{\mathcal{T}} = \sum_{\mathbf{l},\mathbf{t}} u_{(\mathbf{l},\mathbf{t})} \psi_{(\mathbf{l},\mathbf{t})}$ such that

$$\langle \tilde{\psi}_{(\mathbf{l},\mathbf{t})} \ , \ \Delta u_{\mathcal{T}} \rangle = \langle \tilde{\psi}_{(\mathbf{l},\mathbf{t})} \ , \ f \rangle \qquad \forall (\mathbf{l},\mathbf{t}) \in \boldsymbol{\mathcal{T}} \ .$$

Therefore, there holds

$$L_{\mathcal{T}} u_{\mathcal{T}} \ = \ I_{\mathcal{T}} f \ , \tag{53}$$

where

$$I_{\mathcal{T}} f := \sum_{(\mathbf{l},\mathbf{t}) \in \boldsymbol{\mathcal{T}}} \psi_{(\mathbf{l},\mathbf{t})} \langle \tilde{\psi}_{(\mathbf{l},\mathbf{t})} \ , \ f \rangle \quad \text{and} \quad L_{\mathcal{T}} u_{\mathcal{T}} := I_{\mathcal{T}} \Delta u \ .$$

Assume now that the discrete Laplacian $L_{\mathcal{T}}$ satisfies the following stability condition

$$\sup_{\tilde{v}\in\tilde{V}_{\mathcal{T}}} \frac{\langle \tilde{v}\ ,\ L_{\mathcal{T}}u\rangle}{\|\tilde{v}\|_{\mathcal{T}'}} \geq \alpha \|u\|_{\mathcal{T}} \ . \tag{54}$$

Here, $\tilde{V_{\mathcal{T}}}$ is the analogue of $V_{\mathcal{T}}$ for the dual wavelets, $\|.\|_{\mathcal{T}}$ is the maximum norm on the adaptive grid associated to $\boldsymbol{\mathcal{T}}$ and

$$\|\tilde{v}\|_{\mathcal{T}'} = \sup_{u\in V_{\mathcal{T}}} \frac{\langle \tilde{v}\ ,\ u\rangle}{\|u\|_{\mathcal{T}}} \ .$$

Then, the standard estimates yield

$$\begin{aligned}
\|u-u_{\mathcal{T}}\|_{\mathcal{T}} = \|I_{\mathcal{T}}u-u_{\mathcal{T}}\|_{\mathcal{T}} &\leq \frac{1}{\alpha} \sup_{\tilde{v}\in\tilde{V}_{\mathcal{T}}} \frac{\langle \tilde{v}\ ,\ L_{\mathcal{T}}(I_{\mathcal{T}}u-u_{\mathcal{T}})\rangle}{\|\tilde{v}\|_{\mathcal{T}'}} \\
&\leq \frac{1}{\alpha}\|L_{\mathcal{T}}(I_{\mathcal{T}}u-u_{\mathcal{T}})\|_{\mathcal{T}} \\
&= \frac{1}{\alpha}\|\big(L_{\mathcal{T}}I_{\mathcal{T}}u - I_{\mathcal{T}}\Delta u\big) - \underbrace{\big(L_{\mathcal{T}}u_{\mathcal{T}} - I_{\mathcal{T}}\Delta u\big)}_{=0\ ,\ \text{see (53)}}\|_{\mathcal{T}} \\
&= \frac{1}{\alpha}\|\big(L_{\mathcal{T}}I_{\mathcal{T}}u - I_{\mathcal{T}}\Delta u\big)\|_{\mathcal{T}} \ .
\end{aligned}$$

Therefore, the error can be estimated by the consistency error

$$\|L_{\mathcal{T}}I_{\mathcal{T}}u - I_{\mathcal{T}}\Delta u\|_{\mathcal{T}} \ .$$

For the special case of so-called regular sparse grids, where

$$\boldsymbol{\mathcal{T}} = \{(\mathbf{l},\mathbf{t}) \quad | \quad l_1 + .. + l_d \leq n\} \tag{55}$$

for some $n \in \mathbb{N}$, we have shown[1] in [44] that for sufficiently smooth $u$ there holds

$$\|L_{\mathcal{T}}I_{\mathcal{T}}u - I_{\mathcal{T}}\Delta u\|_{\mathcal{T}} \leq C(u)n^{d-1}2^{-(N-2)n} \ .$$

$C(u)$ depends on Hölder-like norms of higher order mixed derivatives of $u$. collocation method is a special higher order FD method. Now, the interesting point of sparse grids is that we have essentially the usual error estimates as for conventional grid discretizations. But the number of degrees of freedom we have to spend is only $O(n^{d-1}2^n)$ - compared to $O(2^{dn})$ for the usual FD or FEM methods on full grids. However, for the general case of an arbitrary adaptive basis and a solution which may have singularities we know of no satisfying consistency proof yet.

The same holds for the stability (54). We are convinced that stability holds, because we never experienced any problems. Furthermore, in [56] systematic experiments were made for the sparse grid FD technique that also indicate stability.

[1] In [44] the consistency of general FD methods on regular sparse grids was shown. Since for regular sparse grids the collocation method is a special higher order FD method this result applies here as well.

## 6 Preconditioning of Linear Systems

Consider the model problem

$$-\Delta u = f \quad \text{in } [0,1]^2 \tag{56}$$

with, for example, homogeneous Dirichlet or Neumann boundary conditions. We later have to deal with such problems in the projection step for the solution of the Navier-Stokes equations. For the numerical solution of (56) we have to solve a linear system of equations

$$\mathbf{A} \cdot \mathbf{u}_{\mathcal{T}} = \mathbf{f}_{\mathcal{T}} \tag{57}$$

with the coefficient matrix

$$\mathbf{A}_{(\mathbf{l},\mathbf{t}),(\mathbf{k},\mathbf{s})} = -\langle \tilde{\psi}_{(\mathbf{l},\mathbf{t})} \ , \ \Delta\psi_{(\mathbf{k},\mathbf{s})} \rangle \ , \quad (\mathbf{k},\mathbf{s}),(\mathbf{l},\mathbf{t}) \in \boldsymbol{\mathcal{T}} \ .$$

Alternatively we also consider the FD discretization of subsections 3.4 and 4.2 for this model problem. We then denote the resulting matrix by $\mathbf{A}$ as well.

From section 3 we have fast algorithms for the matrix vector multiplication with $\mathbf{A}$, but no such algorithm for the transpose of $\mathbf{A}$. Furthermore, $\mathbf{A}$ is non-symmetric and in our experiments we found that $\mathbf{A}$ is indefinite. For all these reasons we employ a GMRES, BiCGStab or BiCGStab(l) solver [25]. As one might expect, the performance of these solvers is not very good, since the condition number grows at least like $4^L$ if $L$ is the finest level of the adaptive basis. Therefore, some kind of preconditioning is required.

The first idea is to use preconditioning from the left [56], i.e. to solve

$$\mathbf{A}\mathbf{C}^{-1} \cdot \bar{\mathbf{u}}_{\mathcal{T}} = \mathbf{f}_{\boldsymbol{\mathcal{T}}} \ , \tag{58}$$

where $\mathbf{C}$ is the following diagonal matrix

$$\mathbf{C}_{(\mathbf{k},\mathbf{s}),(\mathbf{l},\mathbf{t})} = \delta(\mathbf{k}-\mathbf{l})\delta(\mathbf{s}-\mathbf{t}) \sum_{i=1}^{d} 4^{l_i} \ , \quad (\mathbf{k},\mathbf{s}),(\mathbf{l},\mathbf{t}) \in \boldsymbol{\mathcal{T}} \ . \tag{59}$$

This strategy has its origins in the preconditioning of Galerkin stiffness matrices for certain tensor product wavelets [35] which however are different from ours. We applied this preconditioner to the model problem for a non-adaptive basis with finest level $(L,..,L)$, i.e.

$$\boldsymbol{\mathcal{T}} = \{ \ (\mathbf{l},\mathbf{t}) \mid \mathbf{t} \in T(l_1) \otimes .. \otimes T(l_d) \ \text{ and } \ l_0 \le l_i \le L \ , \ (1 \le i \le d) \ \} \ . \tag{60}$$

The grid associated to $\boldsymbol{\mathcal{T}}$ is simply the cartesian mesh with mesh size $2^{-L}$ in both directions. The behaviour of the residual generated by the BiCGStab(2) solver is depicted in Figure 3. The right hand side of the model problem was chosen such that the solution is

$$u(x,y) = (x^2+y^2)^{1/4} \cos\Big(\frac{\arctan(x/y)}{2}\Big)(1-x(1-x)y(1-y)) \ . \tag{61}$$

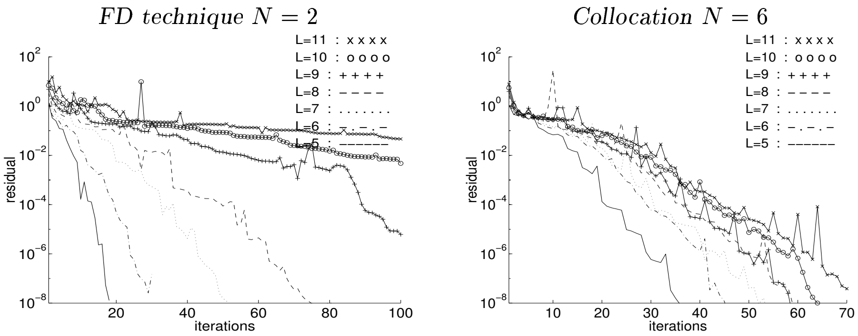


**Fig. 3.** BiCG(2) residuals for the simple diagonal preconditioner (58), the 2D model problem (61) and non-adaptive bases (60).

Obviously the performance of the preconditioned BiCG solver is quite different for the FD discretization with Interpolets ($N = 2$) and for the collocation discretization ($N = 6$). One explanation for this may be the following:

Although there is no satisfying convergence theory for the BiCGStab scheme, it seems to be reasonable that the performance depends on the condition number of the preconditioned coefficient matrix with respect to the $l_2$-norm:

$$\|\{u_l\}_{l\in\mathbb{N}}\|_{l_2}^2 := \sum_{l\in\mathbb{N}} |u_l|^2 \ .$$

The point is that the condition number scales like $2^L$ (or $2^{dL/2}$ in general) for the Interpolets ($N = 2$) with the FD discretization and like $\sqrt{2}^L$ (or $2^{(d-1)L/2}$ in general) for Interpolets ($N \geq 4$) with the FD or the collocation discretization.

In any case, we need a more efficient preconditioner for higher dimensional problems. In the following we will present the essentials of the Lifting Interpolet preconditioner which works for Finite Differences, the collocation method and (in slightly modified form) also for a Galerkin discretization with tensor product type trial functions (50). A more detailed description of this preconditioner will be given in a forthcoming paper [46].

As before, we apply diagonal preconditioning from the left, but we use it together with another multiscale basis $\{\hat{\psi}_{(\mathbf{l},\mathbf{t})}\}$ which is different form the Interpolets. This means that we solve the linear system

$$\mathbf{BAB}^{-1}\mathbf{C}^{-1} \cdot \bar{\mathbf{u}}_{\mathcal{T}} = \mathbf{B} \cdot \mathbf{f}_{\mathcal{T}} \ . \tag{62}$$

Here, $\mathbf{B}$ is the matrix which represents the basis transform

$$\{\psi_{(\mathbf{l},\mathbf{t})}\}_{(\mathbf{l},\mathbf{t})\in\mathcal{T}} \ \rightarrow \ \{\hat{\psi}_{(\mathbf{l},\mathbf{t})}\}_{(\mathbf{l},\mathbf{t})\in\mathcal{T}} \ . \tag{63}$$

The new wavelets $\hat{\psi}_{(\mathbf{l},\mathbf{t})}$ are again tensor products

$$\hat{\psi}_{(\mathbf{l},\mathbf{t})}(\mathbf{x}) = \hat{\psi}_{(l_1,t_1)}(x_1) \cdot \ldots \cdot \hat{\psi}_{(l_d,t_d)}(x_d) \ ,$$

where the univariate wavelets $\hat{\psi}$ are constructed by the lifting scheme [13], [60]. The idea is to modify the given univariate wavelets $\Psi_l$ by coarse scaling functions:

$$\hat{\Psi}_l = 2^{l/2}\big(\Psi_l \; + \; \Phi^{l-1} \cdot Q^l\big) \tag{64}$$

such that the elements of $\hat{\Psi}_l$ have $M$ vanishing moments, i.e.

$$\int \hat{\Psi}_l(x) x^i \, dx = 0 \quad \text{ for } \; 0 \leq i < M \; .$$

The parameter $M$ controls the stabilization effect of the lifting scheme. As we will see below $M = 2$ is a good choice for our purposes. The additional scaling by $2^{l/2}$ in (64) normalizes $\hat{\psi}_{(l,t)}$ with respect to the $L_2$-norm

$$\|\hat{\psi}_{(l,t)}\|_0 \sim 1 \; .$$

The matrix $Q^l$ is a sparse $\#S(l-1) \times \#T(l)$ matrix. E.g. for the non-boundary-modified wavelets we have

$$\hat{\psi}_{(l,t)} = 2^{l/2}\Big(\psi_{(l,t)} \; + \sum_{s=\lfloor \frac{M}{2}-1 \rfloor}^{\lfloor \frac{M}{2} \rfloor} q_s \phi^{(l-1,t+s)}\Big) \; ,$$

with certain properly chosen weights $q_s$. In case of $M = 2$ these weights are $q_0 = q_1 = -\frac{1}{4}$. Furthermore, note that the scaling functions for both wavelet bases are the same. Therefore, the change of basis (63) is possible. However, in the fully adaptive case additional measures [46] are required to ensure that

$$\operatorname{span}\{\hat{\psi}_{(l,t)} \mid (l,t) \in \mathcal{T}\} = \operatorname{span}\{\psi_{(l,t)} \mid (l,t) \in \mathcal{T}\} \; .$$

The effect of the lifting modification (64) has to do with so-called norm equivalences between Sobolev spaces and (coefficient) sequence spaces which later result in optimal $O(1)$ iterative solvers.

**Theorem 2.** *There exist $\gamma^* < \gamma$ such that for $s \in ]\gamma^*, \gamma[$ there are constants $\underline{\alpha}(s) < \bar{\alpha}(s)$ such that*

$$\underline{\alpha}(s)\|\hat{\Psi}_\mathcal{T} \cdot \mathbf{u}_\mathcal{T}\|_s \; \leq \; \Big(\sum_{(l,t)\in\mathcal{T}} |u_{(l,t)}|^2 4^{sl}\Big)^{1/2} \leq \; \bar{\alpha}(s)\|\hat{\Psi}_\mathcal{T} \cdot \mathbf{u}_\mathcal{T}\|_s \quad \forall \; \mathcal{T}, \; \mathbf{u}_\mathcal{T} \; . \tag{65}$$

*The constants $\underline{\alpha}(s)$, $\bar{\alpha}(s)$ are independent on the adaptive univariate basis $\hat{\boldsymbol{\Psi}}_\mathcal{T}$ and the finest level $L = \max\{l \mid (l,t) \in \mathcal{T}\}$ and the coefficient vector $\mathbf{u}_\mathcal{T}$.*

*Proof.* See [59].

The lifting modification significantly enlarges the range $]\gamma^*, \gamma[$ compared to that of the original Interpolets. The reason is that the lower bound $\gamma^*$ mainly depends on the smoothness and the degree of polynomial exactness of the dual scaling functions. By modifying the primal wavelets we implicitly modify the dual scaling functions. The new dual scaling functions $\tilde{\hat{\phi}}^{(\cdot,\cdot)}$ have a degree of polynomial exactness of $M$ and the smoothness parameter

$$-\gamma^* := \sup\{t \in \mathbb{R} \mid \tilde{\hat{\phi}}^{(\cdot,\cdot)} \in H^t\} \ . \tag{66}$$

For example, with $M = 2$ the new dual scaling functions $\tilde{\hat{\phi}}^{(\cdot,\cdot)}$ are in $H^t$ for some $t > 0$. Table 1 shows the associated values of $\gamma^*$ and $\gamma$ which were numerically estimated using Lemma 7.1.2 of [21].

| $M \setminus N$ | 2 | 4 | 6 |
|---|---|---|---|
| – | $]\ 0.500, 1.500[$ | $]\ 0.500, 2.171[$ | $]\ 0.500, 2.755[$ |
| 1 | $]\ 0.354, 1.500[$ | $]\ 0.401, 2.171[$ | $]\ 0.431, 2.755[$ |
| 2 | $]-0.171, 1.500[$ | $]-0.290, 2.171[$ | $]-0.358, 2.755[$ |

**Table 1.** Ranges $]\gamma^*, \gamma[$ of norm equivalences for univariate wavelets

We will now use these norm equivalences for $s = 0, 1$ and 2 to estimate the $l_2$-condition number of the preconditioned operator $\mathbf{BAB}^{-1}\mathbf{C}^{-1}$ for both, the finite difference technique and the collocation method. Therefore we will consider the case $M = 2$ in the following. In the remainder of this subsection we denote by $\mathcal{T}$ the index set

$$\mathcal{T} = \bigcup_{l=l_0}^{L} \{l\} \times T(l) \tag{67}$$

which corresponds to a *non-adaptive* univariate basis. Furthermore, we introduce in Table 2 the $\#S(L) \times \#S(L)$ matrices which form the univariate building blocks of the preconditioned operator.

In the remainder of this paper it will be useful to denote by $X \sim Y$ that there are two constants $\underline{c}$ and $\bar{c}$ independent of $X$ and $Y$ and the various parameters $X$ and $Y$ may depend on such that

$$\underline{c}X \leq Y \leq \bar{c}X \ . \tag{68}$$

However, sometimes it is necessary to keep the constants explicitly.
From the norm equivalency (65) we deduce three important relations:

**Lemma 3.**

*(i) There holds*

$$(\mathbf{u}_\mathcal{T})^T \cdot \mathbf{u}_\mathcal{T} \sim (\mathbf{u}_\mathcal{T})^T \cdot \hat{T}^T \hat{T} \cdot \mathbf{u}_\mathcal{T} \ , \quad \forall \mathbf{u}_\mathcal{T} \ . \tag{69}$$

| | |
|---|---|
| $T$ | transformation matrix defined by $\boldsymbol{\Phi}^L(x) \cdot T \overset{!}{=} \boldsymbol{\Psi}_{\mathcal{T}}$ |
| $\hat{T}$ | transformation matrix defined by $2^{L/2}\boldsymbol{\Phi}^L(x) \cdot \hat{T} \overset{!}{=} \hat{\boldsymbol{\Psi}}_{\mathcal{T}}$ with the scaling $2^{L/2}$ because of $\|\phi_{L,s}\|_0 \sim 2^{-L/2}$ |
| $D$ | $= -\langle \tilde{\boldsymbol{\Phi}}^L \ , \ \partial_{xx}\boldsymbol{\Phi}^L \rangle$ (collocation) $= -4^L[1 \ \ -2 \ \ \ 1]$ (finite differences) |
| $C$ | diagonal scaling matrix with level dependent entries $4^l$, i.e. $C_{(k,s),(l,t)} = \delta(l-k)\delta(s-t)4^l$ |
| $E$ | identity matrix |

**Table 2.** Building blocks of operators

*(ii) There holds*

$$(\mathbf{u}_{\mathcal{T}})^T \cdot C \cdot \mathbf{u}_{\mathcal{T}} \sim (\mathbf{u}_{\mathcal{T}})^T \cdot D\hat{T}^T\hat{T} \cdot \mathbf{u}_{\mathcal{T}} \ , \quad \forall \mathbf{u}_{\mathcal{T}} \ . \tag{70}$$

*(ii) There are $\underline{\alpha}_2$ and $\bar{\alpha}_2$ such that for all $\mathbf{u}_{\mathcal{T}}$*

$$\underline{\alpha}_2(\mathbf{u}_{\mathcal{T}})^T \cdot C^2 \cdot \mathbf{u}_{\mathcal{T}} \ \le \ (\mathbf{u}_{\mathcal{T}})^T \cdot \hat{T}^T D^T D\hat{T} \cdot \mathbf{u}_{\mathcal{T}} \ \le \ \bar{\alpha}_2(\mathbf{u}_{\mathcal{T}})^T \cdot C^2 \cdot \mathbf{u}_{\mathcal{T}} \ . \tag{71}$$

*In case of Interpolets with $N \ge 4$ $\underline{\alpha}_2$ and $\bar{\alpha}_2$ are independent of the finest level L. In case of Interpolets with $N = 2$ there holds*

$$\underline{\alpha}_2(L) \quad \sim \quad 1 \quad \textit{and} \quad \bar{\alpha}_2(L) \quad \sim \quad 2^L \ .$$

*Proof.* We will only sketch the main ideas of the proof for the simple case of periodic boundary conditions.
*(i)* follows from

$$(\mathbf{u}_{\mathcal{T}})^T \cdot \mathbf{u}_{\mathcal{T}} \sim \|\hat{\boldsymbol{\Psi}}_{\mathcal{T}} \cdot \mathbf{u}_{\mathcal{T}}\|_0^2 = 2^L(\mathbf{u}_{\mathcal{T}})^T \cdot \hat{T}^T\langle \boldsymbol{\Phi}^L \ , \ \boldsymbol{\Phi}^L\rangle\hat{T} \cdot \mathbf{u}_{\mathcal{T}} \sim (\mathbf{u}_{\mathcal{T}})^T \cdot \hat{T}^T\hat{T} \cdot \mathbf{u}_{\mathcal{T}} \ ,$$

since the scaled mass matrix $2^L\langle \boldsymbol{\Phi}^L \ , \ \boldsymbol{\Phi}^L\rangle$ is equivalent to $E$.
*(ii)* $D$ and the stiffness matrix for $2^{L/2}\boldsymbol{\Phi}^L$ are circulant. The eigenvalues $\lambda_k$ of both matrices behave like $\sim k^2$. Therefore,

$$(\mathbf{u}^L)^T \cdot D \cdot \mathbf{u}^L \quad \sim \quad 2^L(\mathbf{u}^L)^T \cdot \langle \partial_x\boldsymbol{\Phi}^L \ , \ \partial_x\boldsymbol{\Phi}^L\rangle \cdot \mathbf{u}^L \ , \quad \forall \mathbf{u}^L \ .$$

This yields

$$(\mathbf{u}_{\mathcal{T}})^T \cdot \hat{T}^T D\hat{T} \cdot \mathbf{u}_{\mathcal{T}} \sim \|\hat{\boldsymbol{\Psi}}_{\mathcal{T}} \cdot \mathbf{u}_{\mathcal{T}}\|_1^2 \sim (\mathbf{u}_{\mathcal{T}})^T \cdot C \cdot \mathbf{u}_{\mathcal{T}} \ .$$

*(iii)* For Interpolets ($N \ge 4$) we can proceed as above using

$$(\mathbf{u}^L)^T \cdot D^T D \cdot \mathbf{u}^L \quad \sim \quad 2^L(\mathbf{u}^L)^T \cdot \langle \partial_{xx}\boldsymbol{\Phi}^L \ , \ \partial_{xx}\boldsymbol{\Phi}^L\rangle \cdot \mathbf{u}^L \ .$$

For the case $N = 2$ we refer to the paper [46]. □

Now, we consider the matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{BAB}^{-1}$ of (62) in case of our simple non-adaptive setting (67). For example, in 2D they read

$$\begin{aligned}
\mathbf{A} &= \big(T^{-1} \otimes T^{-1}\big)\big(D \otimes E + E \otimes D\big)\big(T \otimes T\big) \ , \\
\mathbf{B} &= 2^{-L}\big(\hat{T}^{-1} \otimes \hat{T}^{-1}\big)\big(T \otimes T\big) \ , \\
\mathbf{BAB}^{-1} &= \big(\hat{T}^{-1} \otimes \hat{T}^{-1}\big)\big(D \otimes E + E \otimes D\big)\big(\hat{T} \otimes \hat{T}\big) \ , \\
\mathbf{C} &= C \otimes E + E \otimes C \ .
\end{aligned} \tag{72}$$

Here, $X \otimes Y$ denotes the Kronecker product of two matrices [41]. Using all the prerequisites from above we can state the main Theorem of this section:

**Theorem 4.** *For the preconditioned operator* $\mathbf{BAB}^{-1}\mathbf{C}^{-1}$ *of the finite difference technique or the collocation method there holds*

$$\begin{aligned}
&\|\mathbf{BAB}^{-1}\mathbf{C}^{-1} \cdot \mathbf{u}_{\mathcal{T}}\|_{l_2}^2 = \\
&= (\mathbf{u}_{\mathcal{T}})^T \cdot \mathbf{C}^{-1}\mathbf{B}^{-T}\mathbf{A}^T\mathbf{B}^T\mathbf{BAB}^{-1}\mathbf{C}^{-1} \cdot \mathbf{u}_{\mathcal{T}} \sim (\mathbf{u}_{\mathcal{T}})^T \cdot \mathbf{u}_{\mathcal{T}} = \|\mathbf{u}_{\mathcal{T}}\|_{l_2}^2 \ .
\end{aligned} \tag{73}$$

*In case of Interpolets* $N = 2$ *the constants in the above equivalency relation are dependent on the finest level* $L$ *and their ratio scales like* $2^L$*. In all other cases* ($N > 2$) *we obtain by (62) an optimal preconditioner with condition numbers independent on* $L$*.*

*Proof.* Relation (73) follows immediately from

$$(\mathbf{u}_{\mathcal{T}})^T \cdot \mathbf{B}^{-T}\mathbf{A}^T\mathbf{B}^T\mathbf{BAB}^{-1} \cdot \mathbf{u}_{\mathcal{T}} \sim (\mathbf{u}_{\mathcal{T}})^T \cdot \mathbf{C}^2 \cdot \mathbf{u}_{\mathcal{T}} \quad \forall\ \mathbf{u}_{\mathcal{T}} \tag{74}$$

with the same equivalency constants. Now, for the proof of relation (74) we will use the following two general rules (see [35]):
Let $X_i$, $Y_i$ $(1 \le i \le d)$ be $R \times R$ matrices and $\underline{\alpha}_i$, $\bar{\alpha}_i$ such that for all vectors $\mathbf{u} \in \mathbb{R}^R$

$$\underline{\alpha}_i \mathbf{u}^T \cdot X_i \cdot \mathbf{u} \quad \le \quad \mathbf{u}^T \cdot Y_i \cdot \mathbf{u} \quad \le \quad \bar{\alpha}_i \mathbf{u}^T \cdot X_i \cdot \mathbf{u} \ .$$

Then there holds for the tensor product matrices $X_1 \otimes .. \otimes X_d$ and $Y_1 \otimes .. \otimes Y_d$

$$\begin{aligned}
\Big(\prod_{i=1}^d \underline{\alpha}_i\Big)\mathbf{v}^T \cdot \Big(X_1 \otimes .. \otimes X_d\Big) \cdot \mathbf{v} \ &\le \ \mathbf{v}^T \cdot \Big(Y_1 \otimes .. \otimes Y_d\Big) \cdot \mathbf{v} \\
&\le \ \Big(\prod_{i=1}^d \bar{\alpha}_i\Big)\mathbf{v}^T \cdot \Big(X_1 \otimes .. \otimes X_d\Big) \cdot \mathbf{v} \ .
\end{aligned}$$

Furthermore, there holds

$$\begin{aligned}
\min_i(\underline{\alpha}_i)\mathbf{u}^T \cdot \Big(X_1 + .. + X_d\Big) \cdot \mathbf{u} \ &\le \ \mathbf{u}^T \cdot \Big(Y_1 + .. + Y_d\Big) \cdot \mathbf{u} \\
&\le \ \max_i(\bar{\alpha}_i)\mathbf{u}^T \cdot \Big(X_1 + .. + X_d\Big) \cdot \mathbf{u} \ .
\end{aligned}$$

With these rules and the results of Lemma 3 it is straightforward to show that (74) holds. □

The above analysis can easily be carried over to the original preconditioned operator $\mathbf{AC}^{-1}$. There, we have to take into account that the condition number of the mass matrix and the condition number of $T^T T$ scale like $2^L$. This result cannot be improved by suitable diagonal scaling such that $\|\psi_{(l,t)}\|_0 \sim 1$. Therefore, the final results are the scaling laws for the condition numbers

$$\kappa_1 = \sup_{\mathbf{u}_\mathcal{T}} \frac{\|\mathbf{BAB}^{-1}\mathbf{C}^{-1} \cdot \mathbf{u}_\mathcal{T}\|_{l_2}}{\|\mathbf{u}_\mathcal{T}\|_{l_2}} \quad \text{and} \quad \kappa_2 = \sup_{\mathbf{u}_\mathcal{T}} \frac{\|\mathbf{AC}^{-1} \cdot \mathbf{u}_\mathcal{T}\|_{l_2}}{\|\mathbf{u}_\mathcal{T}\|_{l_2}}$$

of the preconditioned operators as given in Table 3.

| | $\mathbf{BAB}^{-1}\mathbf{C}^{-1}$ | $\mathbf{AC}^{-1}$ |
|---|---|---|
| FD ($N = 2$) | $\sim 2^{L/2}$ | $\sim 2^{dL/2}$ |
| FD ($N \geq 4$) | $\sim 1$ | $\sim 2^{(d-1)L/2}$ |
| Col ($N \geq 6$) | $\sim 1$ | $\sim 2^{(d-1)L/2}$ |

**Table 3.** Condition numbers of preconditioned operators on full grids (60).

It is noteworthy that for the preconditioning of the collocation / finite difference matrices one needs a larger range of norm equivalences (up to $\gamma > 2$ in Theorem 2) than for the preconditioning of the Galerkin stiffness matrix. This was also observed in [59] for the collocation method based on the MRA-approach. But in [59] the situation is even worse since norm equivalences for $s \geq 1 + d/2$ are required to obtain a $O(1)$ condition number. For our scheme there is no such dependency on the dimension.

Up to now, our analysis only covers the non-adaptive case. In Figure 5 we present the results of a few numerical experiments for sparse grids (55) which indicate the efficiency of our preconditioner also in more general adaptive cases. In summary, the use of MRA-$d$ Interpolet wavelets leads to quite simple algorithms for the non-linear function evaluation and the discretization of differential operators on one hand, but on the other hand the condition numbers of the resulting operator matrices are strongly dependent on the finest mesh size $2^{-L}$. The lifting preconditioner presented in this section solves this problem and leads to condition numbers which are not or only slightly dependent on the finest mesh size. Therefore, we can solve elliptic problems like (56) fast.

## 7 Time Dependent Convection Diffusion Problems

Consider now the model problem

$$\begin{aligned} \partial_t u + \nabla \cdot (\mathbf{a}u) &= \nu \Delta u \quad , \quad (\mathbf{x}, t) \in \Omega \times [0, T] \\ u(\mathbf{x}, 0) &= u^0(\mathbf{x}) \end{aligned} \tag{75}$$

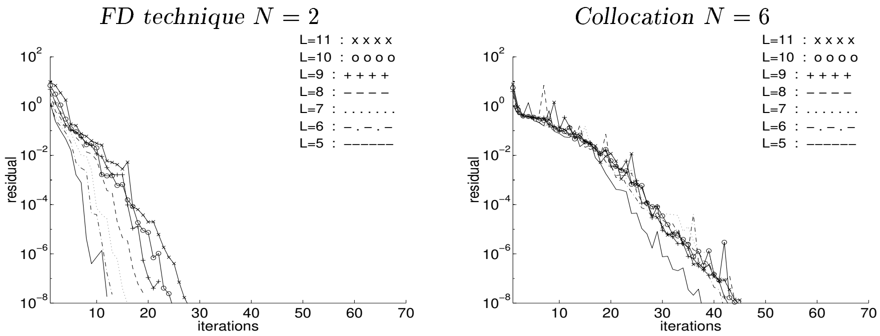


**Fig. 4.** BiCG(2) residuals for the lifting preconditioner (62), the 2D model problem (61) and non-adaptive bases (60).

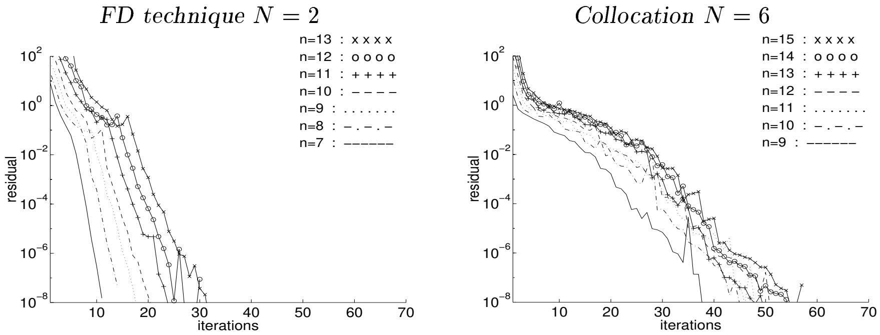


**Fig. 5.** BiCG(2) residuals for the lifting preconditioner (62), the 2D model problem (61) and so-called regular sparse grids (55).

with appropriate boundary conditions. Here, $u$ is a certain quantity which is subject to diffusion and convection. The given velocity field $\mathbf{a}$ is supposed to be divergence free, i.e. we have $\nabla \cdot \mathbf{a} = 0$.

Equation (75) is of the same type as the momentum part of the Navier-Stokes equations. It is common to use the discretizations developed and studied for (75) also for the Navier-Stokes equations. In this section we will combine simple and efficient time stepping methods with our adaptive wavelet collocation scheme. More precisely we will show how the *selection of the adaptive basis* can be incorporated into such methods. Special emphasis will be put on the mass/energy-conservation properties of the adaptive scheme.

### 7.1 Time Discretization

For the discretization of (75) there are essentially two possibilities: Either a simultaneous space time discretization (SPTD) by e.g. our wavelet Petrov-Galerkin scheme or, simpler, a time stepping method (TSM). SPTD in connection with adaptivity would allow to exploit smoothness in space and time (see e.g. [36]) while a TSM can only exploit spatial smoothness. In this sense adaptive SPTDs are a generalization of local time stepping methods or a nonlinear Galerkin scheme [6,52]. Of course, SPTDs also have drawbacks: One has to store and handle the complete solution on $\Omega \times [0,T]$ at once, while for TSM one only needs a few time slabs of the solution. Thus, SPTD may require significantly more memory. Furthermore, SPTDs are not well understood and the arising linear systems are quite nasty and not easy to solve. Hence, we will only consider a time stepping method. For reasons of simplicity we stick to the following simple approach:

$$\frac{u^{n+1}-u^n}{k} \; + \; C(u^n, n^{n-1}, ..) = \nu \Delta u^{n+1} \tag{76}$$

Here, $k$ is the size of the discrete time steps and $C(..)$ is an explicit approximation of the convective term, e.g. of Adams-Bashforth (ABx) or Runge-Kutta (RKx) type (see e.g. [12]). The *implicit* treatment of the diffusive term is relatively inexpensive since the arising linear systems $(id - k\nu\Delta)u = r.h.s.$ are easy to solve. Furthermore one overcomes the CFL type condition $k\nu 4^L \leq c$ ($L$ being the finest level) which is required to maintain the numerical stability of (76). The remaining stability constraint is the usual CFL condition $k2^L \leq C$. In principle we could also overcome this constraint by an implicit treatment of the convective term (e.g. $C = \nabla \cdot (\mathbf{a}u^{n+1})$), but usually the additional effort to solve the more difficult linear system $(id + k\nabla \cdot (\mathbf{a}.) - k\nu\Delta)u = r.h.s.$ (compared to $(id - k\nu\Delta)u = r.h.s.$) does not pay off. The reason is that $k$ and the finest mesh size $2^{-L}$ should be of the same order of magnitude for the sake of accuracy. Because of their simplicity, mixed implicit-explicit discretizations like (76) are used in many CFD codes [43,55].

### 7.2 Adaptive Basis Selection in Time Stepping Methods

An important question for adaptive methods is how to select the adaptive basis to achieve a maximum of accuracy with a given number of degrees of freedom, i.e. with a given amount of numerical work. This depends on the problem at hand. For elliptic problems an algorithm for the basis selection was developed in [15] which yields the optimum order of convergence $\alpha(s,u)$ in terms of the number of degrees of freedom (DOF):

$$\|u - u_{\mathcal{T}}\|_s \leq c(\#DOF)^{-\alpha(s,u)}$$

Here, the error is measured in a Sobolev norm $\|.\|_s$. In engineering applications one is usually not so much interested in minimizing the error in a Sobolev norm, but instead in minimizing the error of derived quantities, like for example the drag or lift coefficient of an airfoil or the mean rate of heat transfer at the surface of a turbo-engine. Based on this observation, an approach was presented in [4,5] which states the problem of adaptive basis selection as an optimization problem. The algorithm to solve this optimization problem involves local error estimators ([9,15,65]) and some properties of Green's function which are required to determine the source of the local errors. However, this algorithm is quite complicated to use in a rigorous way for strongly time-dependent problems like turbulent flows. Instead, we use the following simpler strategy, compare also [29]:

**algorithm (Adaptive Time Stepping Method (ATSM))**

given are the coefficients $\mathbf{u}^0_{\mathcal{T}^0}$ of $u^0$ and the initial adaptive basis $\boldsymbol{\mathcal{T}}^0$

`for` $n = 0$ `to` $T/k$ `do`

`// Time Step:`

Calculate the convective term $\mathbf{c}^n_{\mathcal{T}^n} := \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}^n} \ , \ \nabla \cdot \big(\mathbf{a}(\boldsymbol{\Psi}_{\mathcal{T}^n} \cdot \mathbf{u}^n_{\mathcal{T}^n})\big)\rangle$

Solve $\big(id - k\nu \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}^n} \ , \ \Delta \boldsymbol{\Psi}_{\mathcal{T}^n} \rangle\big) \cdot \mathbf{u}^{n+1}_{\mathcal{T}^n} \ = \ \mathbf{u}^n_{\mathcal{T}^n} - k\mathbf{c}^n_{\mathcal{T}^n}$

`// Adaptivity: new adaptive basis`

Determine $\overline{\boldsymbol{\mathcal{T}}}^{n+1} := \{(\mathbf{l},\mathbf{t}) \ \big| |(u^{n+1}_{\mathcal{T}^n})_{(\mathbf{l},\mathbf{t})}| > \epsilon\}$

Determine $\boldsymbol{\mathcal{T}}^{n+1} := \{(\mathbf{k},\mathbf{s}) \big| \exists \ (\mathbf{l},\mathbf{t}) \in \overline{\boldsymbol{\mathcal{T}}}^{n+1} \text{with } |(\mathbf{l},\mathbf{t}) - (\mathbf{k},\mathbf{s})|_\infty \leq K\}$

`// Adaptivity: prolongation to new basis`

Calculate $\mathbf{u}^{n+1}_{\mathcal{T}^{n+1}}$ by

$$(u^{n+1}_{\mathcal{T}^{n+1}})_{(\mathbf{l},\mathbf{t})} := \begin{cases} (u^{n+1}_{\mathcal{T}^n})_{(\mathbf{l},\mathbf{t})} & \text{if } (\mathbf{l},\mathbf{t}) \in \boldsymbol{\mathcal{T}}^{n+1} \bigcap \boldsymbol{\mathcal{T}}^n \\ 0 & \text{if } (\mathbf{l},\mathbf{t}) \in \boldsymbol{\mathcal{T}}^{n+1} \backslash \boldsymbol{\mathcal{T}}^n \end{cases}$$

`end`

The first part of this algorithm is the time stepping method (76). Here, we use an Euler scheme for the sake of simplicity. Note however that other higher order time discretization methods can be plugged in straightforwardly.

In the second part, we first determine the set $\overline{\boldsymbol{\mathcal{T}}}^{n+1}$ of active basis functions. The analog of this step in adaptive FE/FV methods would be the selection of elements which should be refined. The refinement criterion we use is whether the magnitude of a particular wavelet coefficient is larger than a given threshold $\epsilon$, i.e. $|u_{(\mathbf{l},\mathbf{t})}| > \epsilon$. [2] The use of the magnitude of the wavelet coefficients can also be interpreted as a traditional method of local error estimation. If one takes (3) and (4) into account it is easy to see that the magnitude of a wavelet coefficient $u_{(\mathbf{l},\mathbf{t})}$ is a measure of the magnitude of a local finite difference approximation of some higher order derivative of $u$. Thus, we work with a gradient type error indicator.

Now consider the second step of the adaptivity strategy, i.e. the determination of $\boldsymbol{\mathcal{T}}^{n+1}$ by insertion of basis functions which are close in space and scale to the active basis functions of $\overline{\boldsymbol{\mathcal{T}}}^{n+1}$. We assume that the current solution $u^n$ is quite accurate and we want to preserve this property also for the next time slab $u^{n+1}$. For the sake of simplicity, let us assume that we calculate $u^{n+1}$ by the following explicit Petrov–Galerkin scheme:

$$\langle \tilde{\psi}_{(\mathbf{l},\mathbf{t})} \ , \ u^{n+1} \rangle = \langle \tilde{\psi}_{(\mathbf{l},\mathbf{t})} \ , \ (I - \Delta t \nabla \cdot (\mathbf{v}.) + \Delta t \nu \Delta) u^n \rangle \ .$$

[2] Another possible strategy would be to choose the $M$ largest wavelet coefficients, where $M$ is a fixed given number. In this case the work count would be almost constant for all time steps.

The best index set $\boldsymbol{\mathcal{T}}^{n+1}$ with a given number $N$ of degrees of freedom is that with the $N$ largest entries $\langle\tilde{\psi}_{(\mathbf{l},\mathbf{t})}\ ,\ u^{n+1}\rangle$. Now, the a priori known locality properties of the differential operator $(I - \Delta t\nabla\cdot(\mathbf{v}.) + \Delta t\nu\Delta)$ show that $\langle\tilde{\psi}_{(\mathbf{l},\mathbf{t})}\ ,\ u^{n+1}\rangle$ can only be significant if $(\mathbf{l},\mathbf{t})$ is near to one of the significant $(\mathbf{k},\mathbf{s})\in\boldsymbol{\mathcal{T}}^{n}$. In this sense, the above method is closely related to [4,5] or [15].

### 7.3 Conservation Properties

Consider the numerical simulation of a high Reynolds number turbulent flow. In this setting we can never expect that the numerical solution is close to the real one with respect to a Sobolev norm. However, we don't really need this. Instead, it is sufficient that the numerical solution exhibits the same statistics as the real solution. Obviously one of the prerequisites for a correct prediction of the statistics is that the global mass/energy budget of the numerical and the real solution agree, i.e.

$$\int_\Omega u_{\mathcal{T}}(\mathbf{x},t)\,d\mathbf{x}\approx\int_\Omega u(\mathbf{x},t)\,d\mathbf{x}\quad\text{and}\quad\int_\Omega\big(u_{\mathcal{T}}(\mathbf{x},t)\big)^2\,d\mathbf{x}\approx\int_\Omega\big(u(\mathbf{x},t)\big)^2\,d\mathbf{x}\ .$$

In adaptive schemes there are two main sources of mass or energy defects. The first is the remeshing step for each time step and the second is the discretization of the convective term. For the simple cases of periodic or homogeneous Dirichlet boundary conditions there holds

$$\int_\Omega\nabla\cdot(\mathbf{a}u)\,d\mathbf{x}=0\ ,\quad\int_\Omega\nabla\cdot(\mathbf{a}u)u\,d\mathbf{x}=0\ . \tag{77}$$

In case of Galerkin or non-adaptive collocation discretizations the relations (77) usually also hold for the discretized convective term. But this is not the case for adaptive collocation methods, since here the usual partial summation arguments do not apply. Nevertheless it is relatively simple to enforce at least the first relation in (77) by a $L_2$-projection of $\boldsymbol{\Psi}_{\mathcal{T}}\cdot c^n_{\boldsymbol{\mathcal{T}}}$ to the space of functions with vanishing mean

$$\boldsymbol{\Psi}_{\mathcal{T}}\cdot\hat{c}^n_{\boldsymbol{\mathcal{T}}}\ =\ \boldsymbol{\Psi}_{\mathcal{T}}\cdot c^n_{\boldsymbol{\mathcal{T}}}\ -\ \frac{1}{Vol(\Omega)}\int_\Omega\boldsymbol{\Psi}_{\mathcal{T}}\cdot c^n_{\boldsymbol{\mathcal{T}}}\,d\mathbf{x}\ .$$

Note that this projection even improves the accuracy of the discrete convective term with respect to the $L_2$-norm. Also, the conservation of mass can be easily maintained in the remeshing step. To this end, one switches to the Lifting-Interpolet basis in the second part of algorithm **(ATSM)**. Except for the wavelets of the coarsest scale $(l_0,..,l_0)$ which are always kept in the adaptive basis, all Lifting Interpolet wavelets have a vanishing mean. Therefore the removal or insertion of basis functions has no effect on the mass balance.

Unfortunately such tricks are not available for the energy conservation for both, the convective term and the remeshing step. However, in our numerical experiments we observed that the energy defect introduced by the prolongation in **(ATSM)** and the non-energy-conservative discretization of the convective term is quite small in the temporal mean. Thus, the correct rate of energy dissipation is predicted very well.

As a numerical example we consider the standard test problem of a hill rotating around the origin [66], i.e.

$$\partial_t u + \nabla \cdot \begin{pmatrix} -yu \\ xu \end{pmatrix} = \nu \Delta u \quad \text{in} \quad [-1,1]^2 \times [0, 2\pi]$$
$$u(\mathbf{x}, 0) = \max(0, -3 + 4e^{-30|\mathbf{x}-(.25,.25)|^2}) \ ,$$

see Figure 7 (right). Compared to [66] we increased the difficulty of the problem by using a non-smooth initial condition. Furthermore, we set $\nu = 10^{-5}$ and $\nu = 10^{-6}$ to show that even for such high Reynolds numbers the true energy dissipation $\nu \int |\nabla u|^2 \, d\mathbf{x}$ dominates the energy defects produced by our scheme - at least if the resolution is sufficiently fine. Because of the small dissipative effects the solution is small at the boundaries. Therefore, periodic boundary conditions are used.

In Figure 6 the energy for three different adaptive solutions is compared to the energy of a reference solution which was calculated by a mass- and energy-conservative Fourier spectral code. Obviously the errors of the energy budget are relatively small and remain bounded. Nevertheless, the perturbations which appear for the coarsest adaptive solutions seem to indicate an unsufficient spatial resolution. Such effects are also known from higher order finite difference methods.

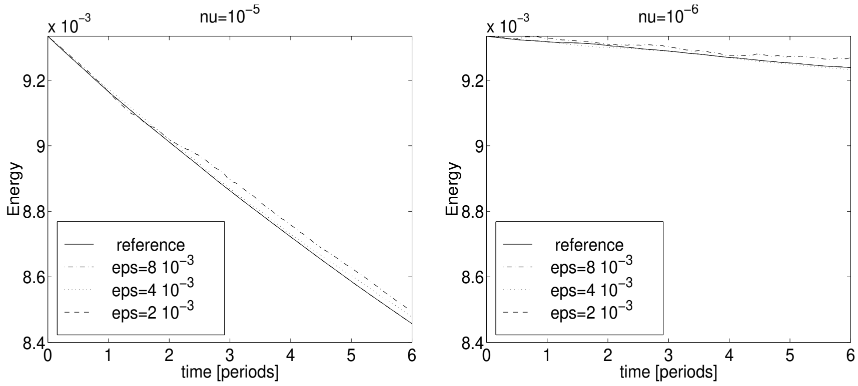


**Fig. 6.** Evolution of energy. Left: $\nu = 10^{-5}$. Right: $\nu = 10^{-6}$.

The evolution of the number of degrees of freedom for the adaptive solutions is shown in Figure 7 (left). At the beginning the number of degrees of freedom (DOF) is high, because of the non-smooth initial condition. Then, the smoothing by the diffusive term allows for a reduction of the number of DOF. However, in the adaptive case a significant part of the DOF is spend

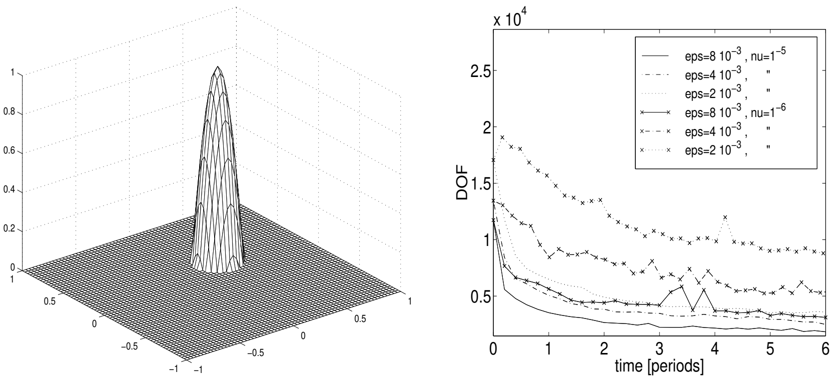


**Fig. 7.** Left: initial condition. Right: evolution of number of degrees of freedom.

to represent small spurious oscillations (wiggles) which are typical for higher order finite difference schemes. These wiggles are suppressed very well in case of $\nu = 10^{-5}$, but not so well for $\nu = 10^{-6}$. This, together with the weaker smoothing of the hill, explains why the number of DOF is larger for $\nu = 10^{-6}$ than for $\nu = 10^{-5}$.

## 8 Incompressible Navier-Stokes Equations

Now, we consider the incompressible Navier-Stokes equations. There, momentum and continuity equation are coupled. This makes them numerically more complicated to treat than the advection-diffusion problem of the previous section. A first idea to overcome this difficulty is to use (weakly) divergence-free trial functions. Then the Navier-Stokes equation boil down to an advection-diffusion problem. In the FE world this approach has been advocated by several authors (see [31] for a recent overview). In the wavelet world the analogue are divergence-free wavelets. They have been introduced in [3] and in [63]. Of course these wavelets are no longer scalar functions but vector valued functions where each component is the tensor product (either of type (49) or of type (50)) of univariate wavelets. One difference to our setting in section 4 is that different types of univariate wavelets are used in each coordinate direction. However the construction of these different univariate wavelets is not practical if the divergence operator contains variable coefficient terms which is the case if the domain $\Omega$ is a parametric image of e.g. $[0,1]^3$.

Another solution to the coupling problem is to switch to the velocity-vorticity formulation of the Navier-Stokes equations (see [38,39,43])

$$
\begin{aligned}
\partial_t \omega + \nabla \cdot (\omega \times \mathbf{u}) &= -\nu \nabla \times \nabla \times \omega \ , \quad \text{in } \Omega \\
\Delta \mathbf{u} &= -\nabla \times \omega \ , \\
\omega &= \nabla \times \mathbf{u} \ , \quad \text{at } t = 0 \ .
\end{aligned}
\tag{78}
$$

The most expensive part of a discretization based on the velocity-vorticity formulation is the evaluation of the nonlinear term $\nabla \cdot (\omega \times \mathbf{u})$ which requires the calculation of the velocity $\mathbf{u}$ for a given vorticity field. To this end, three Poisson equations must be solved in (78) in the three-dimensional case. Besides of rather analytical difficulties such as the lack of boundary conditions for $\omega$ or the treatment of irrotational forcing terms, the costs for the treatment of these three Poisson equations are the main drawback of the velocity-vorticity formulation. In contrast to that only one Poisson equation must be solved in splitting methods which we will describe below.

### 8.1 Splitting Methods and Stabilization

The Navier-Stokes equations in primitive variable formulation are given by

$$\begin{aligned} \partial_t \mathbf{u} \ + \ \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) &= \mathbf{f} - \nabla p + \nu \Delta \mathbf{u} \ , \quad (\mathbf{x}, t) \in \Omega \times [0, T] \ , \\ \nabla \cdot \mathbf{u} &= 0 \ , \\ \mathbf{u}(\mathbf{x}, 0) &= u^0(\mathbf{x}) \end{aligned} \tag{79}$$

with appropriate boundary conditions. Now, the discretization of the Navier-Stokes equations is split into two subproblems:

(Transport step) Calculate the auxiliary velocity $\hat{\mathbf{u}}^{n+1}$ by

$$\frac{\hat{\mathbf{u}}^{n+1} - \mathbf{u}^n}{k} \ + \ \mathbf{C}(\mathbf{u}^n, \mathbf{u}^{n-1}, ..) = \mathbf{f}^{n+1} - \nabla p^n + \nu \Delta \hat{\mathbf{u}}^{n+1} \ . \tag{80}$$

For this problem we apply the **(ATSM)** algorithm of the last section.

(Projection step) Calculate $\mathbf{u}^{n+1}$, $p^{n+1}$ as the solution of

$$\begin{aligned} \frac{\mathbf{u}^{n+1} - \hat{\mathbf{u}}^{n+1}}{k} &= -\nabla (p^{n+1} - p^n) \ , \\ \nabla \cdot \mathbf{u}^{n+1} &= 0 \ . \end{aligned} \tag{81}$$

Of course both subproblems have to be augmented by boundary conditions; not only for the $\mathbf{u}$ and $\hat{\mathbf{u}}$ but also for the pressure $p$. Especially the (in general non-physical) pressure boundary conditions are the subject of a controversy (cf. [31,43,54]) which is beyond the scope of the paper. Therefore we will assume for the rest of this work that periodic boundary conditions hold for $\mathbf{u}$ and $p$.

The saddle point problem (81) is treated by solving the Schur complement equation

$$\nabla \cdot \nabla (p^{n+1} - p^n) = \frac{1}{k} \nabla \cdot \hat{\mathbf{u}}^{n+1} \ , \tag{82}$$

followed by the correction of the velocity

$$\mathbf{u}^{n+1} = \hat{\mathbf{u}}^{n+1} - k \nabla (p^{n+1} - p^n) \ . \tag{83}$$

Since the equations of the saddle point problem (81) must hold only in the weak sense (i.e. tested by $\tilde{\boldsymbol{\Psi}}_{\mathcal{T}}$), the discrete Schur complement operator $\nabla \cdot \nabla$ is not the usual discrete Laplacian $\langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}} \ , \ \Delta \boldsymbol{\Psi}_{\mathcal{T}} \rangle$, but the product of a discrete gradient and a discrete divergence operator. Thus, the matrix representation of the discrete $\nabla \cdot \nabla$ operator reads

$$\begin{aligned} \mathbf{A} &= \sum_{i=1}^{d} \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}} \ , \ \partial_{x_i} \boldsymbol{\Psi}_{\mathcal{T}} \rangle \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}} \ , \ \boldsymbol{\Psi}_{\mathcal{T}} \rangle^{-1} \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}} \ , \ \partial_{x_i} \boldsymbol{\Psi}_{\mathcal{T}} \rangle \\ &= \sum_{i=1}^{d} \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}} \ , \ \partial_{x_i} \boldsymbol{\Psi}_{\mathcal{T}} \rangle \langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}} \ , \ \partial_{x_i} \boldsymbol{\Psi}_{\mathcal{T}} \rangle \ . \end{aligned} \tag{84}$$

Obviously the matrix-vector multiplication with $\mathbf{A}$ is approximately twice as expensive as that with $\langle \tilde{\boldsymbol{\Psi}}_{\mathcal{T}} \ , \ \Delta \boldsymbol{\Psi}_{\mathcal{T}} \rangle$. However as it was mentioned in [31] for FEM the replacement of $\nabla \cdot \nabla$ by $\Delta$ in (82) is no good advise since non-physical solutions may occur [30]. Furthermore in our own experiments with unsteady problems and the naive discretization of $\Delta$ we observed that the projection step constantly injects an amount of divergence into the velocity field $\mathbf{u}^{n+1}$ which is of the order of the discretization error. This leads to a relatively large number of BiCG iterations required for the accurate solution of the pressure Poisson equation.

However, the use of the operator $\nabla \cdot \nabla$ also causes some problems. Consider the matrix representation $\mathbf{A}$ for the 2D non-adaptive case:

$$\mathbf{A} = \left(T^{-1} \otimes T^{-1}\right)\left(G^2 \otimes E + E \otimes G^2\right)\left(T \otimes T\right) \ ,$$

where $G = \langle \tilde{\boldsymbol{\Phi}}^L \ , \ \partial_x \boldsymbol{\Phi}^L \rangle$. Here, $-G^2$ is not spectrally equivalent to the stiffness matrix $\langle \partial_x \boldsymbol{\Phi}^L \ , \ \partial_x \boldsymbol{\Phi}^L \rangle$ (unlike $-\langle \tilde{\boldsymbol{\Phi}}^L \ , \ \partial_{xx} \boldsymbol{\Phi}^L \rangle$). See Figure 8 for a comparison of the scaled eigenvalues of the one-dimensional operators for Interpolets ($N = 6$). As usual the eigenvalues $\lambda_j$ are normalized by the factor $4^{-L}$ and they are plotted against the so-called wavenumber $2^{-L} \cdot 2\pi j$. Hence, the resulting plots are almost independent on the level $L$.

Now, the point is that spectral equivalence was one of the prerequisites for the efficiency of both the $\mathbf{AC}^{-1}$ and the $\mathbf{BAB}^{-1}\mathbf{C}^{-1}$ preconditioning technique of section 6. Hence, even the preconditioned version of (82) is very ill conditioned. Furthermore, $G^2$ has an additional zero eigenvalue for the even-odd mode. Thus, spurious checker-board modes occur in the solution of (82). The FE analog of the above problems is the lack of a LBB condition for the saddle point problem (81).

Our solution to this problem is to replace $G$ in the gradient operator by $G + 2^{-6L}F$ where $F$ is a central FD discretization of the 6th order derivative. Conversely, in the divergence operator we replace $G$ by $G - 2^{-6L}F$. Then, $F$ and $G$ commute in the periodic case and

$$(G - 2^{-6L}F)(G + 2^{-6L}F) = (G^2 - 2^{-12L}F^2) \ . \tag{85}$$

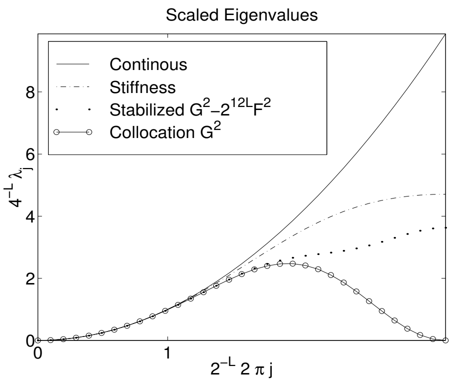


**Fig. 8.** Eigenvalues of 1D operators

Hence, the matrix representation of the stabilized Schur complement operator reads

$$\mathbf{A} = \left(T^{-1} \otimes T^{-1}\right)\left((G^2 - 2^{-12L}F^2) \otimes E + E \otimes (G^2 - 2^{-12L}F^2)\right)\left(T \otimes T\right) .$$

Figure 8 shows that the eigenvalues of the stabilized one-dimensional operator are equivalent (in the sense of (68)) to the stiffness matrix. Therefore this operator does not lead to spurious pressure oscillations and it ensures the efficiency of the preconditioners of section 6.

In contrast to the usual stabilization methods known from the FE literature [8,42,67] where the original operator

$$\begin{pmatrix} \Delta & \nabla \\ \nabla\cdot & 0 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} I & \nabla \\ \nabla\cdot & 0 \end{pmatrix} \text{, respectively,}$$

is stabilized by

$$\begin{pmatrix} \Delta & \nabla \\ \nabla\cdot & S \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} I & \nabla \\ \nabla\cdot & S \end{pmatrix} \text{, respectively,} \tag{86}$$

the above method replaces the gradient and the divergence operator by up- and down-winded operators $\nabla^+$, $\nabla^-\cdot$ leading to the stabilized operator

$$\begin{pmatrix} I & \nabla^+ \\ \nabla^-\cdot & 0 \end{pmatrix} .$$

Hence the Schur complement operator $\nabla^- \cdot \nabla^+$ has the same structure as the original operator $\nabla \cdot \nabla$. This is not the case for $\nabla \cdot \nabla - S$. Therefore our velocity field is always (weakly) divergence-free with respect to $\nabla^-\cdot$ and we do not run into all that trouble which comes with non-divergence-free velocity fields.

In the adaptive case the stabilization terms are evaluated by the adaptive FD algorithm (**FD**). In all our experiments this method performed very well.

### 8.2 Numerical Experiment: Three Vortices Interaction

Now, we will apply the adaptive wavelet solver developed in the previous sections to the model problem of the interaction of three vortices in a two-dimensional flow, see [58]. We use MRA-$d$ Interpolet wavelets where $N = 6$.

The initial velocity is induced by three vortices each with a Gaussian vorticity profile

$$\omega(\mathbf{x},0) = \omega_0 + \sum_{i=1}^{3} \omega_i \exp\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{\sigma_i^2}\right) , \quad \mathbf{x} \in [0,1]^2 .$$

The first two vortices have the same positive sign $\omega_1,\ \omega_2 > 0$ and the third has a negative sign, see Figure 9. The different parameters $\omega_0,\ \omega_i$ and $\sigma_i$ are

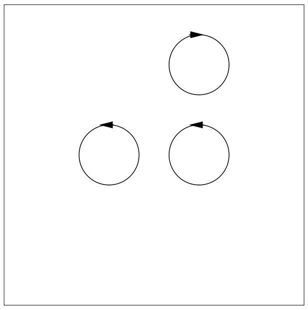

**Fig. 9.** Initial configuration for the three vortices interaction.

chosen such that the mean value of $\omega(.,0)$ vanishes and that $\omega(.,0)\,|_{\partial[0,1]^2}$ is almost $\omega_0$ to allow for periodic boundary conditions.

Due to the induced velocity field the three vortices start to rotate around each other. In a later stage the two same-sign vortices merge which leads to a configuration of two counter-rotating vortices. This process is the basic mechanism in 2D turbulence and it takes place e.g. in the shear layer problem of the next subsection or during the convergence of $\omega$ to the solution of the Joyce-Montgomery equation [17] for random initial vorticity fields.

In our numerical experiments the viscosity was set to $\nu = 3.8 \cdot 10^{-6}$. The maximum velocity of the initial condition is $7 \cdot 10^{-2}$ which corresponds to a Reynolds number of $\approx 55200$. A reference solution was obtained with a Fourier spectral code ($512 \times 512$ modes) which is based on the splitting method with a third order Adams-Bashforth scheme for the convective term. This third order scheme was also used for the adaptive wavelet solver. Here MRA-$d$ Interpolet wavelets of 6th order (i.e. $N = 6$) were employed in a collocation method for the spatial discretization. The values of the threshold parameter $\epsilon$ were set to $\{8,\ 4,\ 1\} \cdot 10^{-4}$ and the finest level $\mathbf{l}$ was limited to $(10, 10)$ to avoid very fine time steps. The time step used in both codes was $dt = 10^{-2}$.

In Figure 12 the contour lines of the vorticity of the spectral and the adaptive solutions are shown. The contour levels are equally spaced from $-1.5$ to 3 which is approximately the minimum/maximum value of $\omega(.,0)$.

Although the reference solutions and the adaptive solutions are qualitatively the same, the adaptive solution seems to overestimate the rotation of the cores of the vortices. Also in the time behaviour of the energy $E(t)$ and the enstrophy $Z(t)$

$$E(t) = \int_\Omega |\mathbf{u}|^2 \, d\mathbf{x} \ , \quad Z(t) = \frac{1}{2} \int_\Omega |\omega|^2 \, d\mathbf{x}$$

there are differences, see Figure 10. However, these differences are very small. Further experiments have shown that this error stems from the spatial discretization and may be reduced by relaxing the level limitation and by decreasing the threshold parameter $\epsilon$.

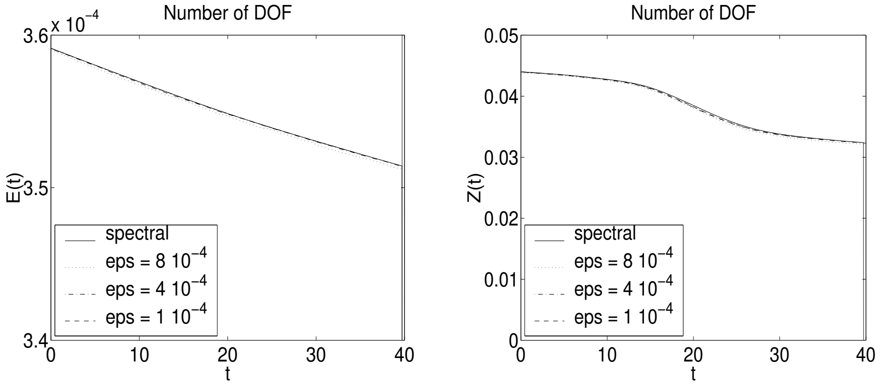


**Fig. 10.** Left: Evolution of energy. Right: Evolution of enstrophy.

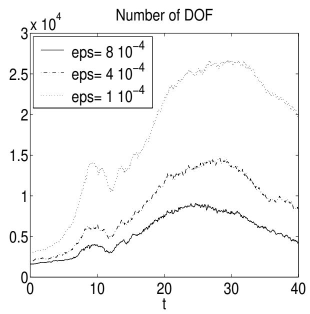


**Fig. 11.** Evolution of the number of DOF.

The evolution of the number of degrees of freedom is shown in Figure 11. Obviously our method recognizes the arising complicated flow patterns in space and time (see Figure 12) and spends there much more degrees of

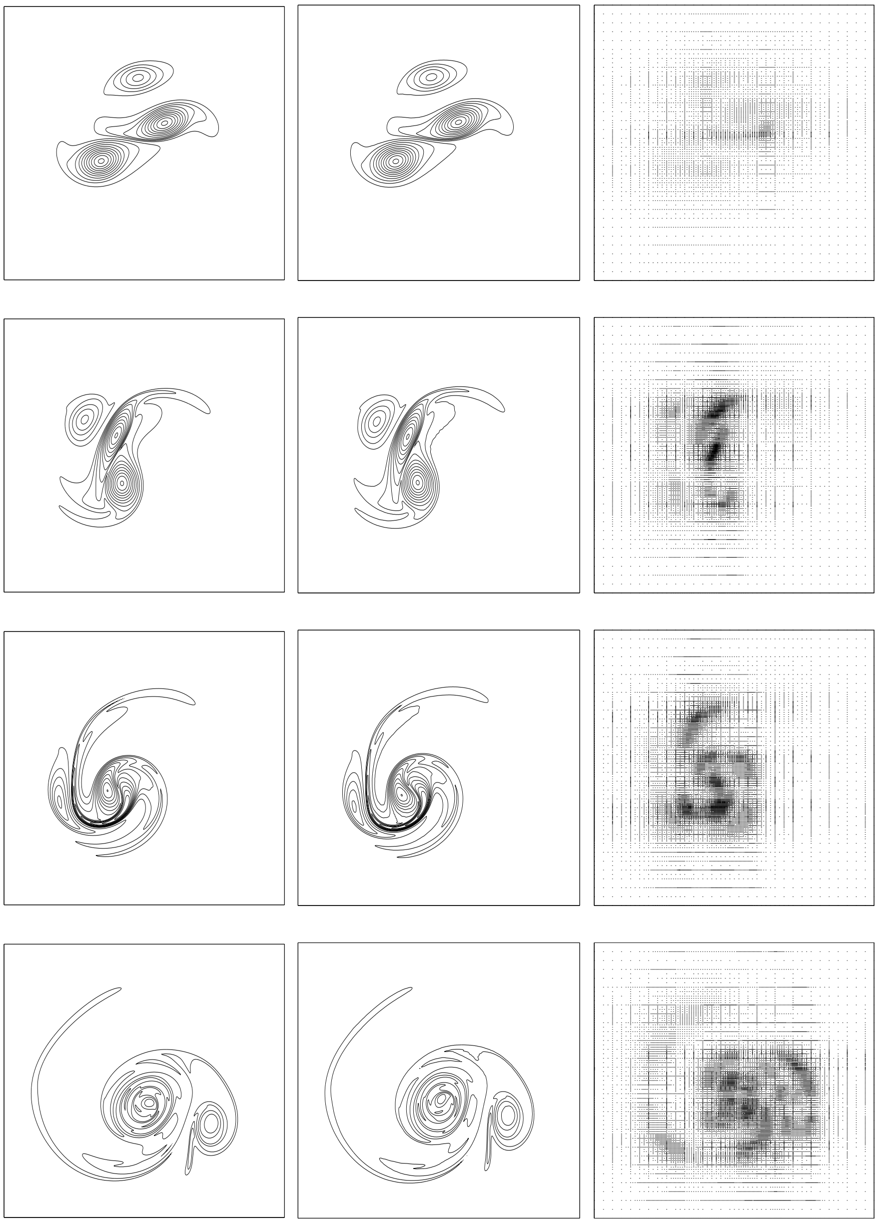

**Fig. 12.** Isolines of $\omega$ at $t = 5,\ 10,\ 15,\ 35$ (top–bottom), left: spectral code, middle: adaptive wavelet solver ($\epsilon = 4e-4$), right: adaptive grid.

freedom than in smooth regions. Note that the left and middle pictures of Figure 12 are enhanced by 20 %.

Despite of small discrepancies the results of the adaptive wavelet solver are quite accurate. To achieve this accuracy only a small number of degrees of freedom was needed – less than 1% in comparison to the reference solution.

### 8.3 Numerical Experiment: the Shear Layer

Now we apply the adaptive wavelet solver to a shear layer model problem. The initial configuration of the temporally developing shear layer is a velocity field with a hyperbolic–tangent profile $u(y) = U\tanh(2y/\delta)$, where $\delta$ is the vorticity shear layer thickness $\delta = 2U/\max(du/dy)$, see Figure 13. From

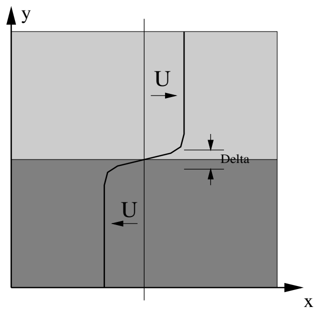


**Fig. 13.** Initial configuration for the mixing layer.

linear stability analysis this initial configuration is known to be inviscidly unstable. Slight perturbations are amplified by Kelvin-Helmholtz instabilities, where the most amplified mode corresponds to a wavelength $\lambda = 7\delta_0$ [50].

In our numerical experiments the vorticity thickness $\delta$ was chosen such that 10 vortices should develop in the computational domain $[0,1]^2$. The maximum velocity was $U = 1.67 \cdot 10^{-2}$ and the viscosity was $\nu = 3.8 \cdot 10^{-6}$. The instabilities were triggered by a superimposed weak white noise in the shear layer, compare [57].

The numerical simulations were applied to the periodized version of the problem with two shear layers on $[0,1] \times [0,2]$. The reference simulation was carried out with a Fourier spectral code (splitting method, AB3) with $512 \times 1024$ modes. We used the same wavelet solver as in the previous experiment. The values of the threshold parameter $\epsilon$ were set to $\{12,\ 8,\ 4\} \cdot 10^{-4}$ and the finest level $\mathbf{l}$ was limited to $(10,10)$. Both, the spectral and the adaptive solver used a time step of $5 \cdot 10^{-3}$.

In Figure 16 the results of the run for $\epsilon = 8 \cdot 10^{-4}$ are compared with the reference solution. Both results agree quite well. For $t = 36$ we observe a phase shift for the right-most two vortices in the adaptive simulation. As in the first experiment, the adaptive code seems to overestimate the rotation of the vortices. Despite of these slight differences in the instantaneous vorticity fields, the statistical quantities energy and enstrophy agree very well, see Figure 14.

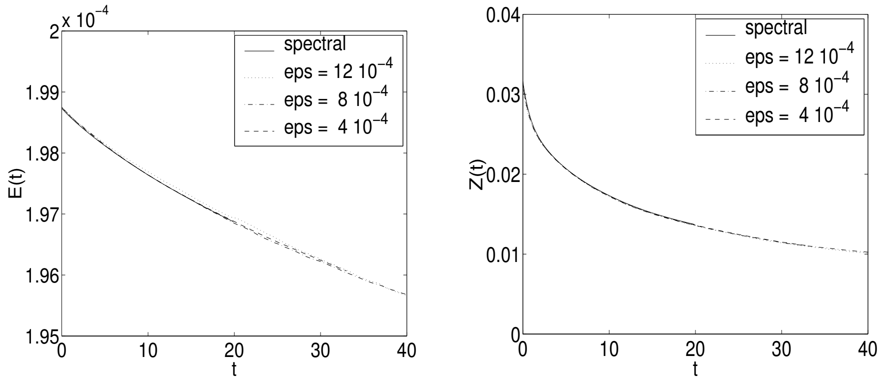


**Fig. 14.** Left: Evolution of energy. Right: Evolution of enstrophy.

The evolution of the number of degrees of freedom (DOF) is shown in Figure 15. The initial velocity is not very smooth due to the white noise added for triggering the instability. Therefore, a large number of DOF is required to resolve the initial velocity in the shear layer sufficiently well. Then, in a first phase, the diffusion smooths the velocity very fast which

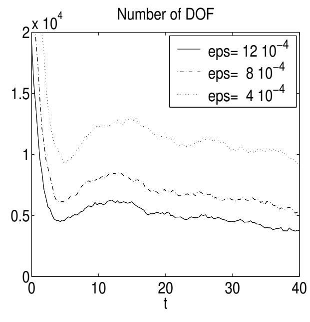


**Fig. 15.** Evolution of the number of DOF.

leads to the strong decay of the number of DOF. This process is stopped by the development of the Kelvin-Helmholtz instabilities leading to an increase of the number of DOF ($4 < t < 10$). In the last phase ($t > 10$) the number of coherent vortices constantly decreases by successive merging. Therefore, in this stage the number of DOF decreases almost constantly.

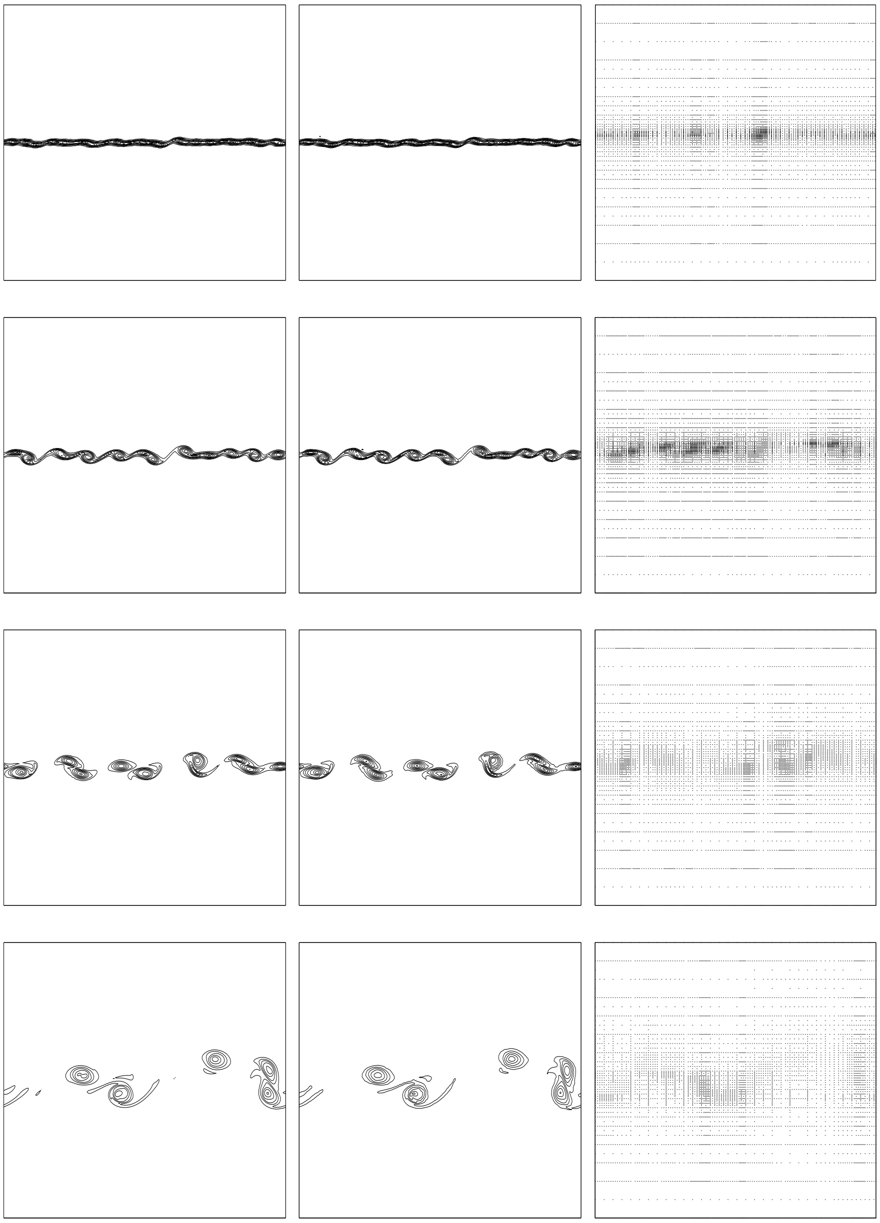

**Fig. 16.** Isolines of $\omega$ at $t = 4,\ 8,\ 16,\ 36$ (top–bottom), left: spectral code, middle: adaptive wavelet solver ($\epsilon = 8e - 4$), right: adaptive grid.

The numerical experiments of subsections 8.2 and 8.3 show that our adaptive wavelet scheme is a quite efficient and accurate solution method. The adaptive time stepping method predicts very well the locations where adaptive refinement is needed. Furthermore, the consistent discretization of the Schur complement operator in connection with our stabilization technique and the preconditioners of section 6 lead to an efficient solution of the projection problem which is usually the most time consuming part of a Navier-Stokes solver.

## 9 Concluding Remarks

In this paper we presented a new method for the numerical treatment of the Navier-Stokes equations which uses simple tensor products of univariate Interpolets as trial functions in an adaptive Petrov-Galerkin scheme. The tensor product approach allows to carry the univariate adaptive algorithms (and codes) directly over to the multivariate case for arbitrary dimensions. Furthermore this MRA-$d$-approach leads to superior approximation properties of the resulting multivariate basis. We also presented a simple preconditioning technique for the linear systems of equations which stem from the discretization of elliptic PDEs. This technique is based on the lifting scheme and results in optimal $O(1)$ condition numbers. Then, a simple time-stepping method is employed for the discretization of time-dependent problems. There, also the algorithm for the adaptive basis selection is incorporated and the adaptive basis is now evolved from time slab to time slab. In our numerical experiments this method proved to be robust and efficient.

In summary, we always used the simplest and most efficient parts of the wavelet approaches to build up our method. The Interpolets lead to simple algorithms for evaluation of non-linear terms. The MRA-$d$ tensor product approach and the Petrov-Galerkin discretization with the dual wavelets as test functions form the most efficient and easiest way to carry the univariate adaptive algorithms over to the case of arbitrary dimensions. We could even use the complete C++ code developed for the univariate case in the multivariate code as well. Another implication of the MRA-$d$-approach is the pronounced locality of data within the univariate algorithms. This is important for modern microprocessor architectures. Finally the lifting preconditioner solved in a simple way the remaining problem of our approach: the badly conditioned linear systems.

All these techniques (tensor product wavelets, lifting preconditioner, time stepping method and adaptive basis selection) form the building blocks of our adaptive wavelet solver for the Navier-Stokes equations. For the discretization of these equations we used a Chorin-type projection method. In comparison to simple advection-diffusion problems the additional difficulty is the stable discretization of the pressure Poisson equation. To this end we employed an upwind-downwind scheme for the pressure gradient and the divergence operator, respectively. This approach mimics the structure the continuous operator which is required to avoid spurious oscillations and to maintain the efficiency of our preconditioners. Finally we demonstrated the efficiency and accuracy of our method for two examples of high Reynolds-number flows in 2D.

However, there is still a lot to do. On the theoretical side this is mainly the proof of the consistency and stability of our Petrov-Galerkin scheme and the proof of the efficiency of our lifting preconditioner in the fully adaptive case. On the practical side the main improvements will concern the parallelization of the method. Here we intend to use the space-filling curve technique [37]. Also a generalization of the method to more complicated geometries by means of parametric mappings and block structuring has to be accomplished in the next future.

**Acknowledgements.** The authors would like to thank Marie Farge and Kai Schneider for proposing the numerical experiments of section 8 and for kindly providing the initial conditions of the mixing layer simulation.

## Notations

| | |
|---|---|
| $c$ | generic constant |
| $C$ | diagonal matrix, see Table (2) |
| $\mathbf{C}$ | diagonal matrix, (59) |
| $C(.,.)$ | $C(<\text{matrix}>, <\text{index set}>)$, (33) |
| $C(.,.)$ | $C(u^n, u^{n-1}, ..)$, convective term, (76) |
| $\mathbf{C}(.,.)$ | $\mathbf{C}(u^n, u^{n-1}, ..)$, convective term, (80) |
| $d$ | dimension of computational domain, e.g. $[0,1]^d$ |
| $\mathbf{e}$ | $= (e_1, .., e_d)$, $e_i \in \{0,1\}$, (49) |
| $g_s$ | scaling coefficient for $\psi$, similar to (16) |
| $\tilde{g}_s$ | " , for $\tilde{\psi}$, (16) |
| $G^l$ | scaling matrix for level $l$, (19) |
| $\tilde{G}^l$ | " , for dual wavelets, (19) |
| $h_s$ | scaling coefficient for $\phi$, (1) |
| $\tilde{h}_s$ | " , for $\tilde{\phi}$, (7) |
| $\mathbf{h}$ | $= \{h_s\}_{s\in\mathbb{Z}}$, sequence/vector of scaling coefficients |
| $H^l$ | scaling matrix for level $l$, (19,29) |
| $\tilde{H}^l$ | " , for dual scaling functions, (19) |
| $k$ | level of refinement, see $l$ |
| $\mathbf{k}$ | $= (k_1, .., k_d)$, see $\mathbf{l}$ |
| $l$ | refinement level of scaling function $\phi^{(l,s)}$ or wavelet $\psi_{(l,t)}$, (9,11) |
| $\mathbf{l}$ | $= (l_1, .., l_d)$, multivariate refinement level, (50) |
| $L$ | a certain finest level in e.g. an adaptive basis |
| $n$ | number of time step |
| $N$ | degree of polynomial exactness of $\phi$, (4) |
| $M$ | " of $\tilde{\hat{\phi}}$, (4,64) |

| | |
|---|---|
| $p$ | the pressure, (79) |
| $R(.,.)$ | $R(<\text{matrix}>, <\text{index set}>)$, (33) |
| $s$ | (in most cases) translation index for scaling functions, (9) |
| $\mathbf{s}$ | $= (s_1, .., s_d)$ |
| $S(l)$ | index set of all scaling functions on level $l$, $S(l) = \{0, .., 2^l\}$ or $\{1, .., 2^l - 1\}$, $\{0, 2, .., 2^l - 2, 2^l\}$,... depending on the b.c., (25) |
| $S$ | pressure stabilization term, (86) |
| $\mathcal{S}(l)$ | $\subseteq S(l)$, (31) |
| $\mathcal{S}^a(l)$ | $\subseteq S(l)$, (48) |
| $\mathcal{S}^b(l)$ | $\subseteq S(l)$, see algorithm $(\mathcal{S}^b)$ |
| $t$ | translation index for wavelets, (11,12,13) |
| $\mathbf{t}$ | $= (t_1, .., t_d)$, " for multivariate wavelets, (50) |
| $T(l)$ | index set of all wavelets on level $l$, $T(l_0) = \{0, .., 2^{l_0}\}$ and $T(l) = \{0, .., 2^{l-1} - 1\}$ for $l > l_0$, (26) |
| $T$ | see Table 2 |
| $\hat{T}$ | see Table 2 |
| $\mathcal{T}(l)$ | $\subseteq T(l)$, (30) |
| $\boldsymbol{\mathcal{T}}(\mathbf{l})$ | $\subseteq T(l_1) \otimes ... \otimes T(l_d)$, (51) |
| $\boldsymbol{\mathcal{T}}$ | see (51) |
| $u$ | a function $[0, 1]^d \to \mathbb{R}$ |
| $\mathbf{u}$ | the velocity, (79) |
| $u^{(l,s)}$ | a single scaling function coefficient, (13,32) |
| $u_{(l,t)}$ | a single wavelet coefficient, (13,32) |
| $u_{(\mathbf{l},\mathbf{t})}$ | " , multivariate case, (51) |
| $\mathbf{u}^l$ | sequence/vector of scaling function coefficients, (13) |
| $\mathbf{u}^{\mathcal{S}(l)}$ | " , adaptive case, (32) |
| $\mathbf{u}_l$ | sequence/vector of wavelet coefficients, (13) |
| $\mathbf{u}_{\mathcal{T}(l)}$ | " , adaptive case, (32) |
| $\mathbf{u}_{\boldsymbol{\mathcal{T}}(\mathbf{l})}$ | " , adaptive, multivariate case, (51) |
| $\mathbf{u}_{\mathcal{T}}$ | " , for complete adaptive basis, (32) |
| $\mathbf{u}_{\boldsymbol{\mathcal{T}}}$ | " , for complete, adaptive, multivariate basis, (51) |
| $\phi$ | mother function for primal scaling functions, (1) |
| $\tilde{\phi}$ | " , for dual scaling functions, (6) |
| $\phi^{(l,t)}$ | a single scaling function, (9,21) |
| $\boldsymbol{\Phi}^l$ | row vector of all scaling functions $\phi^{(l,s)}$ on level $l$, (12,32) |
| $\boldsymbol{\Phi}^{\mathcal{S}(l)}$ | " , adaptive case, (51) |
| $\psi$ | mother function for primal wavelets |
| $\tilde{\psi}$ | " , for dual wavelets |
| $\hat{\psi}$ | " , for Lifting-wavelets |
| $\tilde{\hat{\psi}}$ | " , for dual Lifting-wavelets |
| $\psi_{(l,t)}$ | a single wavelet, (11) |
| $\hat{\psi}_{(l,t)}$ | a single Lifting-wavelet, (11) |
| $\boldsymbol{\Psi}_l$ | row vector of all wavelets $\psi_{(l,t)}$ on level $l$, (26) |
| $\boldsymbol{\Psi}_{\mathcal{T}(l)}$ | " , adaptive case, (32) |

## References


1. Angot P., Bruneau C., Fabrie P.: *A penalization method to take into account obstacles in incompressible flows*; Preprint No. 97017, Mathematiques Appliquees, University Bordeaux (1997).
2. Barinka A., Barsch T., Charton P., Cohen A., Dahlke S., Dahmen W., Urban K.: *Adaptive wavelet schemes for elliptic problems – implementation and numerical experiments*; Institut für Geometrie und praktische Math., RWTH Aachen (1999), preprint.
3. Battle G., Federbush P.: *Divergence-free vector wavelets*; Michigan Math. J., 40 (1993) pp. 181–195.
4. Becker R., Rannacher R.: *Weighted a posteriori error control in FE methods*; in ENUMATH-95 in Proc. ENUMATH-97, World Scientific Publ., Singapore (1998).
5. Becker R.: *Weighted a posteriori error estimators for FE approximations of the incompressible Navier-Stokes equations*; Universität Heidelberg, IWR, SFB 359, Preprint 48 (1999).
6. Burie J.M, Marion M.: *Multilevel Methods in Space and Time for the Navier-Stokes Equations*; SIAM J. Num. Anal. 34 No. 4 (1997), pp. 1574–1599.
7. Bramble J.H., Pasciak J.E., Xu J.: *Parallel multilevel preconditioners*; Math. Comp. 55 (1990), pp. 1–22.
8. Brezzi F., Pitkäranta J.: *On the Stabilization of Finite Element Approximations of the Stokes Problem*; in Efficient Solutions of the Stokes Problem, Hackbusch W. (Eds.), Vieweg (1984).
9. Bank R., Weiser A.: *Some a posteriori error estimators for elliptic PDEs*; Math. Comp. 44 (1985), pp. 283–301.
10. Bertoluzza S.: *An adaptive collocation method based on interpolating wavelets*; Istituto di analisi numerica del consiglio nazionale delle ricerche, Pavia (1997)
11. Beylkin G., Keiser J.M.: *An Adaptive Pseudo-Wavelet Approach for Solving Nonlinear Partial Differential Equations*; in Multi Scale Wavelet Methods for Partial Differential Equations, Vol. 6 in the Wavelet Analysis and Applications series, Academic Press.
12. Canuto C., Hussaini M., Quarteroni A., Zang T.: *Spectral Methods in Fluid Dynamics*; Springer Verlag (1987).
13. Carnicer J.M., Dahmen W., Pena J.M.: *Local decomposition of refinable spaces*; Appl. Comp. Harm. Anal. 3 (1996), pp. 127–153.
14. Charton P., Perrier V.: *A Pseudo-Wavelet Scheme for the two-dimensional Navier-Stokes Equations*; Matematica Aplicada e Computacional (1996)
15. Cohen A., Dahmen W., DeVore R.: *Adaptive wavelet methods for elliptic operator equations – convergence rates*; IGPM-Report, (1998), RWTH Aachen.
16. Cohen A., Daubechies I., Feauveau J.-C.: *Biorthogonal bases of compactly supported wavelets*; Comm. pure and Appl. Math. 45 (1992), pp.485–560.
17. Chorin A.: *Vorticity and Turbulence*; Springer Verlag (1998).
18. Dahmen W., Schneider R., Xu Y.: *Nonlinear Functionals of Wavelet Expansions – Adaptive Reconstruction and Fast Evaluation*; IGPM-Report 160, (1998), RWTH Aachen, to appear in Numerische Mathematik.
19. Daubechies I.: *Orthonormal Bases of compactly supported Wavelets I*; Comm. Pure Appl. Math. (1988).
20. Daubechies I.: *Orthonormal Bases of compactly supported Wavelets II*; SIAM J. Math. Anal. 24 (1993).

21. Daubechies I. : *Ten lectures on wavelets*; CBMS-NSF Regional Conference Series in Appl. Math. 61 (1992).
22. Deslauriers G., Dubuc S. : *Symmetric iterative interpolation processes*; Constr. Approx. 5 (1989), pp. 49–68.
23. DeVore R. : *Nonlinear Approximation*; Acta Numerica, 7 (1998), pp. 51–150.
24. DeVore R., Konyagin S.V., Temlyakov V.N. : *Hyperbolic wavelet approximation*; Constr. Approx. 14 (1998), pp. 1–26.
25. Dongarra J., Duff I., Sorensen D., van der Vorst H. : *Numerical Linear Algebra for High-Performance Computers*; SIAM (1998).
26. Donoho D. : *Interpolating wavelet transform*; Stanford University (1992), preprint.
27. Donoho D., Yu P. : *Deslauriers-Dubuc: Ten Years After*; in Deslauriers G., Dubuc S. (Eds), CRM Proceedings and Lecture Notes Vol. 18 (1999).
28. Frisch U. : *Turbulence*; Cambridge University press (1995).
29. Fröhlich J., Schneider K. : *An Adaptive Wavelet-Vaguelette Algorithm for the Solution of PDEs*; J. Comput. Physics 130 (1997), pp. 174–190.
30. Gresho P., Chan S. : *On the theory of semi-implicit projection methods*; Int. J. Numer. Meth. Fluids 11(5) (1990).
31. Gresho P., Sani R. : *Incompressible Flow and the Finite Element Method*; Wiley (1998).
32. Griebel M. : *Multilevelmethoden als Iterationsverfahren ueber Erzeugendensystemen*; Teubner Verlag (1994).
33. Griebel M. : *Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences*; Computing 61 No. 2 (1998), pp. 151–180.
34. Griebel M., Knapek M. : *Optimized tensor-product approximation spaces*; Constr. Approx., in print.
35. Griebel M., Oswald P. : *Tensor product type subspace splitting and multilevel iterative methods for anisotropic problems*; Adv. Comp. Math. 4 (1995), pp. 171–206.
36. Griebel M., Zumbusch G. : *Adaptive Sparse Grids for Hyperbolic Conservation Laws.*; Proceedings of Seventh International Conference on Hyperbolic Problems: Theory, Numerics, Applications, ISNM 129, Birkh"user (1999), Vol. 1, pp. 411–421.
37. Griebel M., Zumbusch G. : *Parallel Adaptive Subspace Correction Schemes with Applications to Elasticity*; accepted for Computer Methods in Applied Mechanics and Engineering, Elsevier, (1999).
38. Gunzburger M., Mundt M., Peterson P. : *Experiences with FEM for the velocity-vorticity formulation of 3D viscous incompressible flows*; in Computational methods for viscous flows Vol 4. Brebbia C. (Eds.) (1990).
39. Hafez M., Dacles J., Soliman M. : *A velocity/vorticity method for viscous incompressible flow calculations*; in Lecture Notes in Physics Vol. 323 Dwoyer D., Hussaini M. (Eds.) (1989).
40. Holmstroem M. : *Solving Hyperbolic PDEs Using Interpolating Wavelets*; SIAM J. Sc. Comp. 21/2 (1999) pp. 405–420.
41. Horn R., Johnson C. : *Topics in Matrix Analysis*; Cambridge Univ. press (1989).
42. Hughes T., Franca L. : *A new finite element formulation for computational fluid dynamics: VII*; Comput. Meth. Appl. Mech. Eng. Vol 65 (1987) pp. 85–96.
43. Karniadakis G., Sherwin S. : *Spectral/hp Element Methods for CFD*; Oxford University Press (1999).

44. Koster F.: *A Proof of the Consistency of the Finite Difference Technique on Sparse Grids*; University Bonn, (2000) SFB-Report No. 642.
45. Koster F., Griebel M.: *Orthogonal wavelets on the interval*; University Bonn, (1998) preprint No. 576.
46. Koster F., Griebel M.: *Efficient Preconditioning of Linear Systems for the Finite Difference and the Collocation Method on Sparse Grids*; University Bonn, (2000), in preparation.
47. Kunoth A.: *Fast Iterative Solution of Saddle Point Problems in Optimal Control Based on Wavelets*; University Bonn, (2000) SFB-Report.
48. Lippert R., Arias T., Edelman A.: *Multi Scale Computation with Interpolating Wavelets*; J. Comp. Physics 140 (1998), pp. 278–310.
49. Metais O., Lesieur M.: *Turbulence and Coherent Structures*; Kluwer (1991).
50. Michalke A.: *On the inviscid instability of the hyperbolic tangent velocity profile*; J. Fluid Mech. 19 (1964) 543–556.
51. Monzon L., Beylkin G., Hereman W.: *Compactly supported wavelets based on almost interpolating and nearly linear phase filters (Coiflets)*; Appl. Comp. Harm. Anal., 7, (1999), pp. 184–210.
52. Marion M., Temam R.: *Nonlinear Galerkin Methods*; SIAM J. Num. Anal. 26 (1989), pp. 1139–1157.
53. Oswald P.: *Multilevel Finite Element Approximation*; Teubner Verlag (1994).
54. Prohl A.: *Projection and Quasi-Compressibility Methods for Solving the Incompressible Navier-Stokes Equations*; Teubner Verlag (1997).
55. Peyret R., Taylor T.: *Computational Methods for Fluid Flow*; Springer (1983).
56. Schiekofer T.: *Die Methode der finiten Differenzen auf dünnen Gittern zur Lösung elliptischer und parabolischer PDEs*; PhD thesis, Bonn (1998).
57. Schneider K., Farge M.: *Numerical simulation of a temporally growing mixing layer in an adaptive wavelet basis*; C.R.A.S. Paris, Ser. IIb, (2000)
58. Schneider K., Kevlahan N., Farge M.: *Comparison of an Adaptive Wavelet Method and Nonlinearly Filtered Pseudospectral Methods for Two-Dimensional Turbulence*; Theor. Comput. Fluid Dyn. 9, (1997), pp. 191-206
59. Schneider R.: *Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur effizienten Lösung großer vollbesetzter Gleichungssysteme*; Advances in Numerical Mathematics, Teubner Stuttgart (1998).
60. Sweldens W.: *The lifting scheme: a construction of second generation wavelets*; SIAM J. Math. Anal. 29,2 (1997), pp. 511–546.
61. Sweldens W., Piessens R: *Quadrature formulae and asymptotic error expansions for wavelet approximations of smooth functions*; SIAM J. Num. Anal. 36,3 (1994), pp. 377–412.
62. Temlyakov V.: *Approximation of periodic functions*; Nova Science Publ. New York (1994).
63. Urban K.: *On Divergence-free Wavelets*; Adv. Comp. Math. Vol. 4 (1995).
64. Vasilyev O., Paolucci S.: *Fast Adaptive Wavelet Collocation Algorithm for Multidimensional PDEs* ; J. Comp. Phys. 138/1 (1997) pp. 16–56.
65. Verfürth R.: *A Review of a Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*; Teubner Verlag (1996).
66. Vreugdenhil C.B., Koren B.: *Numerical methods for Advection-Diffusion Problems*; Notes on Numerical Fluid Mechanics Vol. 45, Vieweg Verlag (1993).
67. Zienkiewicz O., Wu J.: *Incompressibility without tears – how to avoid restrictions of mixed formulations*; Int. J. Numer. Meth. Eng. Vol 32 (1991), pp. 1189–1203.