Multiscale Methods for the Simulation of Turbulent Flows

Michael Griebel[1] and Frank Koster[1]

Universität Bonn, Institut für Angewandte Mathematik,
Wegelerstr. 6, D-53115 Bonn, Germany

Summary

In this paper we apply the finite difference method on adaptive sparse grids [9] to the simulation of turbulent flows. This method combines the flexibility and efficiency of finite difference schemes with the advantages of an adaptive approximation by tensor product multiscale bases. We shortly discuss the method. Then, we present numerical results for a simple linear convection problem for a validation of our scheme. Finally, results for three-dimensional turbulent shear layers are shown.

1 Introduction

Turbulent flows are a grand challenge for numerical simulations. In such flows there is a large range of active and important scales which all need to be resolved by a numerical mesh. Therefore, very fine meshes with large numbers of degrees of freedom (DOF) must be used. For example, for isotropic turbulence, Kolmogorov's K41 theory predicts that $O(Re^{9/4})$ spatial DOF are required for a sufficient resolution of the flow quantities. Besides the trivial (and expensive) way to use faster and larger computers, there are mainly two possibilities to reduce the computing times. The first possibility is to develop faster algorithms for, e.g., the solution of the linear systems of equations resulting from the discretization of the Navier-Stokes equations. Here, multigrid methods [24] are almost optimal, and it seems there is not much space for further improvements.

The second possibility is to reduce the required numbers of DOF by a sophisticated discretization scheme. Turbulence is characterized by coherent structures, like vortex tubes or vortex sheets, which govern the dynamics and the statistics of the flow. It is, therefore, quite intuitive to try to concentrate the numerical work on these structures. Since the coherent structures are spatially localized, this can be achieved by, for example, adaptive mesh refinement. In this sense, adaptive methods can overcome the $O(Re^{9/4})$-rule for the number of DOF. This rule is related to an *uniform* mesh refinement via an approximation of the flow quantities by means of Fourier-modes.

In this paper we describe a numerical scheme which follows this idea. However, instead of refined meshes we use adaptive multiscale bases for the representation of the numerical solution. Such bases provide powerful means to detect (quasi-)singularities in the solution, i.e. they directly lead to quite simple refinement strategies.

Furthermore, the coefficient matrices for, e.g., Poisson equations can be easily and effectively preconditioned. In this way, multiscale bases lead to an efficient adaptive representation of the solutions and also to fast solution methods. Another important issue is the discretization of the governing equations. For multiscale bases, this is usually more complicated than for, e.g., non-adaptive or adaptive finite difference or finite volume methods. Here, some progress has been made with the development of the so-called finite difference method on adaptive sparse grids [9, 10, 16, 20]. In the present paper we apply this method to the simulation of turbulent flows. The scheme is based on tensor products of special multiscale functions, the Interpolets. This allows to quickly switch between a multiscale representation of the solution and a representation by means of nodal values on certain grids. Hence, non-linear functionals can be easily evaluated. Finite differences are employed in a non-standard way for the discretization of the differential operators. We can use very general finite difference stencils, like WENO stencils or stabilized schemes for div-grad operators. Thus, the method combines the flexibility and efficiency of finite difference schemes and the advantages of multiscale representations.

This paper is organized as follows. In the next section we introduce two types of multivariate multiscale bases and we compare their approximation properties. This motivates why we use tensor product bases for our scheme. We point out the pros and cons of this approach. In Section 3 we give a brief introduction to the finite difference method on adaptive sparse grids for the example of a linear convection equation. The last two sections are devoted to numerical applications. First, we present the results of a linear convection problem (the Molenkamp test). We compare our method with a non-adaptive finite difference scheme. The results of the numerical simulation of a three-dimensional shear layer are shown in Section 5. In this case, statistical quantities are considered for the validation.

2 Multiscale Bases and Approximation of Functions

Let $\{\psi_{lt}\}_{l=l_0}^{\infty}$ be a basis of univariate multiscale functions defined on the unit interval $[0,1]$. The index l is the level of refinement, l_0 is the coarsest level and t is some spatial shift index. Examples for such bases are the orthogonal Daubechies wavelets [4], the biorthogonal spline wavelets [2] or generalizations of the hierarchical basis [8, 25] – the so-called Interpolets [5, 7]. In case of the hierarchical basis, the functions ψ_{lt} are just dilates and translates of the hat function $\phi(x) = \max(0, 1 - |x|)$, i.e.

$$\psi_{lt} := \begin{cases} \phi(2^l x - (2t+1)) & l > l_0 := 0,\ 0 \le t < 2^{l-1} \\ \phi(2^{l_0} x - t) & 0 \le t \le 2^{l_0} \end{cases} .$$

General Interpolets are defined in the same way, starting from smoother variants of the hat function, see Figure 1.

There are essentially two different approaches to build multivariate multiscale bases on the d-dimensional unit cube $[0,1]^d$ from the univariate functions ψ_{lt}. The ansatz most commonly used in the wavelet community [18] leads to basis functions

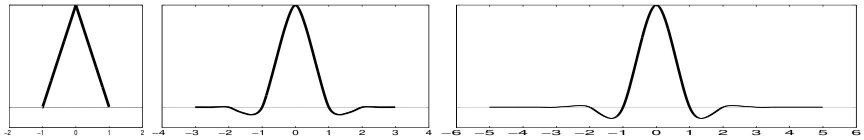


Figure 1 Mother functions ϕ for Interpolets of order $P = 2, 4$ and 6.

with a more or less isotropic support. In case of Interpolets, the multiscale functions are defined by

$$\psi_{l,\mathbf{t},\mathbf{e}}(\mathbf{x}) := \begin{cases} \prod_{i=1}^{d} \phi(2^l x_i - (2t_i + 1 + e_i)) \ l > l_0 \\ \prod_{i=1}^{d} \phi(2^{l_0} x_i - t_i) \ , \end{cases} \quad \text{(isotropic)} \tag{1}$$

where $\mathbf{e} \in \{0,1\}^d \backslash \mathbf{1}$ is a type index. Throughout this paper, vectors and multi-indices are written boldface. For example, $\mathbf{e}$ is the multi-index $(e_1, ..., e_d)$ and $\mathbf{x} = (x_1, ..., x_d)$. The second approach uses tensor products of the ψ_{lt}, i.e.

$$\psi_{\mathbf{lt}}(\mathbf{x}) := \prod_{i=1}^{d} \psi_{l_i t_i}(x_i) \ . \quad \text{(anisotropic)} \tag{2}$$

The support of these functions may be strongly anisotropic depending on how much the components of the level index $\mathbf{l}$ differ.

Both multiscale bases (1) and (2) have quite different approximation properties. In order to explain the differences, we consider the so-called best-n-term approximation of a given function u: $[0,1]^d \rightarrow \mathbb{R}$. Here, one takes the best selection of n basis functions out of the bases (1) or (2), and the best-approximation with respect to the resulting n-dimensional ansatz-space. Then, the error of best-n-term approximation is

$$\sigma_n(u) := \inf_{\Lambda : |\Lambda| = n} \ \inf_{v \in \mathrm{span}\{\psi_\lambda\}_{\lambda \in \Lambda}} \|u - v\|_{L^2} \ . \tag{3}$$

To unify notation, we denote by ψ_λ a wavelet of type (1) or (2) and λ is either an index $(l, \mathbf{t}, \mathbf{e})$ or $(\mathbf{lt})$. The basis which leads to the fastest decay of $\sigma_n(u)$ when n goes to ∞ is the most efficient one. The error σ_n is also a benchmark for the performance of a numerical scheme for the solution of PDEs based on the respective basis. No matter what discretization we are using and no matter what kind of refinement strategy we apply, the numerical error can not be better than σ_n in the L^2-norm. Usually it is sufficient that the numerical error is of the same order as σ_n. In this case, the scheme is quasi-optimal.

In this section, we consider the approximation error only. If u is in a function space X, e.g., a Sobolev space, then one typically has estimates for σ_n of the following type

$$\sigma_n(u) \lesssim n^{-\alpha} \|u\|_X \ . \tag{4}$$

For the bases (1) and (2), such estimates are known for Sobolev spaces H^s, Sobolev spaces of functions with bounded mixed derivatives H^s_{mix} [21], Besov spaces B^s [23]

and anisotropic Besov spaces $B^{(s_1,\ldots,s_d)}$ [13]. Sobolev spaces measure the global smoothness of functions. Therefore, they are not appropriate for the characterization of the approximation properties of functions with singularities or quasi-singularities. Besov spaces are suited to that. In the following we briefly recall some of the known estimates for σ_n. In general, the approximation rate α in (4) depends on the smoothness of u, i.e. X, the dimension d and the order P of polynomial exactness of the underlying univariate basis functions ψ_{lt}. Order P means that all polynomials of degree less than P can be written as linear combination of the basis functions $\{\psi_{l_0 t}\}$ of the coarsest level. For example, $P = 2$ for the hierarchical basis.

If u has bounded mixed derivatives or if u has anisotropic smoothness, e.g., if u has singularities aligned to the axes like boundary layers, then the approximation rate is much larger for the anisotropic basis (2) than for the the isotropic basis (1). For example, for $u \in H^s_{mix}$ we have

$$\sigma_n(u) \lesssim \|u\|_{H^s_{mix}} \begin{cases} n^{-\min(s,P)/d} & \text{for (1)} \\ (\frac{n}{\log^{d-1} n})^{-\min(s,P)} & \text{for (2)}\,, \end{cases}$$

i.e. the approximation rate is almost d times larger for the anisotropic basis than for the isotropic basis. This is the essential of the so-called sparse grid or hyperbolic cross approximation [21, 26].

If u is in a standard Sobolev- or Besov space, then the approximation rates are more or less the same for both types of bases. For example, let $s^* := \sup\{s : u \in H^s\}$ be the Sobolev-regularity of u. Then, we get the well-known Jackson inequality

$$\sigma_n(u) \lesssim n^{-\min(s,P)/d}\|u\|_{H^s} \quad \forall\, s < s^* \tag{5}$$

for (1) as well as for (2). In this case, the optimal sets in (3) have the form

$$\Lambda = \{(l, \mathbf{t}, \mathbf{e}) :\ l_0 \le l \le n^{1/d}\} \quad \text{or} \quad \Lambda = \{(\mathbf{l}, \mathbf{t}) :\ l_0 \le |\mathbf{l}|_\infty \le n^{1/d}\},$$

respectively, i.e. they correspond to a uniform mesh refinement with mesh size $n^{-1/d}$.

The worst case for the anisotropic basis are functions with singularities not aligned to the axes. In such cases, it is appropriate to characterize the smoothness of u by means of Besov spaces, since these allow for much sharper estimates of σ_n than the Jackson inequality for Sobolev spaces. Let $s^{**} := \sup\{s : u \in B^s\}$. This regularity parameter is much larger than s^* in (5), compare [6]. For the isotropic basis, we get $\sigma_n(u) \lesssim n^{-\min(s,P)/d}\|u\|_{B^s}$ for $s < s^{**}$. For the anisotropic basis, we get the estimate

$$\sigma_n(u) \lesssim \left(\frac{n}{\log^k n}\right)^{-\min(s,P)/d} \|u\|_{B^s} \quad \forall\, s < s^{**}.$$

Here, $k \le d-1$ is a parameter which depends on the dimension of the manifold on which the singularity lives. E.g., $k = 0$ for point singularities [16].

From the above relations we may draw the following conclusions. The anisotropic basis leads to a much higher approximation rate than the isotropic basis in many important cases. In the worst case, it just looses a logarithmic factor, since the basis functions can not efficiently represent singularities which are not aligned to the axes. However, in the general case, the anisotropic basis seems to be more promising than the isotropic basis. Therefore we use the anisotropic multiscale functions for our numerical scheme.

3 Numerical Method

We describe the numerical method for the solution of a simple linear convection equation

$$\begin{aligned} \partial_t u + \nabla \cdot (\mathbf{a}u) &= 0 \qquad (\mathbf{x}, t) \in [0,1]^2 \times [0,T] \\ u(\mathbf{x}, 0) &= u^0(\mathbf{x}) \end{aligned} \tag{6}$$

with a given divergence free velocity $\mathbf{a} = (a_1, a_2)$. This equation is discretized using an explicit time stepping scheme, e.g.

$$u^{n+1} = u^n - \Delta t \nabla \cdot (\mathbf{a}u^n) \; . \tag{7}$$

In the code we employ a third order Adams-Bashforth or a fourth order Runge-Kutta scheme. In contrast to common (adaptive) finite difference methods, the primary unknowns are not the nodal values of the time slabs u^n on certain grids, but the multiscale coefficients $u^n_{\mathbf{lt}}$ with respect to an adaptive (anisotropic) basis. The solution at the time $n\Delta t$ is written as

$$u^n = \sum_{\mathbf{lt} \in \mathcal{T}^n} u^n_{\mathbf{lt}} \psi_{\mathbf{lt}} \; .$$

The index sets $\mathcal{T}^n$ also evolve from time step to time step. Here, we use the following simple heuristics [11] based on a thresholding of the multiscale coefficients with a parameter $\delta > 0$:

$$\mathcal{T}^{n+1} = \mathcal{T}^n_\delta \;\cup \bigcup_{\mathbf{lt} \in \mathcal{T}^n_\delta} \mathcal{N}(\mathbf{lt}) \quad \text{where} \quad \mathcal{T}^n_\delta = \{\mathbf{lt} \in \mathcal{T}^n \,:\, |u^n_{\mathbf{lt}}| > \delta\} \; . \tag{8}$$

$\mathcal{N}(\mathbf{lt})$ is the set of neighbours of $\mathbf{lt}$ in space and level [16]. The index set $\mathcal{T}^0$ is adapted to the approximation of the initial condition u^0. For the spatial discretization of (7) we compute an approximation c of the convective term $\nabla \cdot (\mathbf{a}u^n)$ with respect to the *old* basis, i.e. index set $\mathcal{T}^n$. Then we have $u^* := u^n - \Delta t c \in \operatorname{span}\{\psi_{\mathbf{lt}} \,:\, \mathbf{lt} \in \mathcal{T}^n\}$. The multiscale coefficients of u^{n+1} with respect to the new index set $\mathcal{T}^{n+1}$ are defined by

$$u^{n+1}_{\mathbf{lt}} := \begin{cases} u^*_{\mathbf{lt}} & \text{if } \mathbf{lt} \in \mathcal{T}^n \cap \mathcal{T}^{n+1} \\ 0 & \text{else} \; . \end{cases} \tag{9}$$

This explains why we also included the neighbours of the significant indices of $\mathcal{T}^n$ in $\mathcal{T}^{n+1}$: The index set $\mathcal{T}^{n+1}$ shall be large enough to allow for an accurate approximation of u^*. For the discretization of the convective term c, we proceed in a similar way as in pseudo-spectral codes. We describe the scheme for the term $\partial_x a_1 u^n \approx c^1$, only. Our aim is to find the multiscale coefficients $c^1_{\mathbf{lt}}$, $\mathbf{lt} \in \boldsymbol{\mathcal{T}}^n$, of c^1. In the first step, we compute the interpolant $v = \sum_{\mathbf{lt} \in \boldsymbol{\mathcal{T}}^n} v_{\mathbf{lt}} \psi_{\mathbf{lt}}$ of $a_1 u^n$ with respect to the nodes

$$\Omega_{\boldsymbol{\mathcal{T}}^n} := \{(x(l_1,t_1), x(l_2,t_2)) : \mathbf{lt} \in \boldsymbol{\mathcal{T}}^n\}$$
$$\text{where} \quad x(l,t) := \begin{cases} 2^{-l}(2t+1) & l > l_0, \\ 2^{-l}t & \text{else} . \end{cases}$$

I.e. v is defined by $v(\Omega_{\boldsymbol{\mathcal{T}}^n}) = a_1 u^n(\Omega_{\boldsymbol{\mathcal{T}}^n})$. Interpolets allow for a quite simple procedure to compute this interpolant: By means of the inverse multiscale transform applied to $\{u^n_{\mathbf{lt}}\}$ we get the nodal values of $u^n(\Omega_{\boldsymbol{\mathcal{T}}^n})$. Pointwise multiplication with the given nodal values of a_1 yields $a_1 u^n(\Omega_{\boldsymbol{\mathcal{T}}^n})$, and finally the multiscale transform applied to $a_1 u^n(\Omega_{\boldsymbol{\mathcal{T}}^n})$ yields to desired coefficients $v_{\mathbf{lt}}$. The simplicity of this scheme makes the Interpolets so unique among the various multiscale bases. The evaluation of products of two functions or more general non-linear functionals is much more involved for other bases like the biorthogonal spline wavelets [3].

In the second step the partial derivative $\partial_x v$ is discretized. The algorithm for the evaluation of this term essentially boils down to univariate multiscale transforms and an univariate finite difference scheme. Therefore, we decompose $\boldsymbol{\mathcal{T}}^n$ into *one-dimensional* index sets

$$\mathcal{T}_{l_2 t_2} := \{(l_1 t_1) : (l_1 l_2, t_1 t_2) \in \boldsymbol{\mathcal{T}}^n\}$$

for all $(l_2 t_2)$. Then, the algorithm consists of three substeps:

1. For a particular index $(l_2 t_2)$, we read out the coefficients $\{v_{l_1 l_2, t_1 t_2}\}_{(l_1 t_1) \in \mathcal{T}_{l_2 t_2}}$, and perform an univariate inverse multiscale transform. This gives the nodal values of a partial function on a univariate adaptive grid.
2. To these nodal values we apply the one-dimensional finite difference scheme we want, for example, a WENO scheme to minimize the spurious oscillations.
3. Then, we perform the univariate multiscale transform and we get the desired coefficients $c^1_{l_1 l_2, t_1 t_2}$ for all $(l_1 t_1) \in \mathcal{T}_{l_2 t_2}$.
4. The steps 1-3 repeat for all $(l_2 t_2)$ with $\mathcal{T}_{l_2 t_2} \neq \emptyset$.

In this way the discretization of partial derivatives is traced back to univariate finite difference schemes on non-uniform grids. The scheme is very flexible, since we can plug in very general finite difference stencils. Of course, one has to be careful with stencils on the non-uniform grids. We refer to [10, 16] for the algorithmical details. The scheme directly generalizes to higher dimensions. An analysis of the consistency error is given in [14, 16, 20].
The above discretization can be used also for elliptic problems like the Poisson equation. In this case, efficient solvers for the resulting linear systems of equations are needed. We have shown that a simple diagonal scaling of the coefficient matrices leads to condition numbers which grow slowly with the maximal refinement level. Furthermore, we introduced a more sophisticated preconditioner based on the Lifting scheme which results in condition numbers which are essentially independent of the maximal refinement level [10, 17].

4 Validation for a Convection Problem

In this section we present the results of the multiscale-based finite difference method for the model problem (6) on the domain $[-1,1]^2$. The initial condition is a non-smooth cone centered in the point $(\frac{1}{4},\frac{1}{4})$

$$u^0(x,y) = \max(0, 4e^{-15((x-\frac{1}{4})^2+(y-\frac{1}{4})^2)} - 3)\ .$$

The velocity $\mathbf{a}(x,y) = (-y,x)^t$ corresponds to a circular rotation around the origin. The parameters of the adaptive scheme are shown in Table 1. In order to avoid very small time steps induced by the CFL-condition, we limited the maximal level of refinement in (8).

Table 1 Parameters of the adaptive finite difference scheme.

Interpolets	: order $P = 4$
stencil for convective term	: 5th order WENO
maximal level of refinement	: 8 and 9
threshold parameter	: $\delta = 3 \cdot 10^{-2}, ..., 3 \cdot 10^{-4}$
time discretization	: 3rd order Adams-Bashforth (AB3)
time step	: $\Delta t = 0.0005$

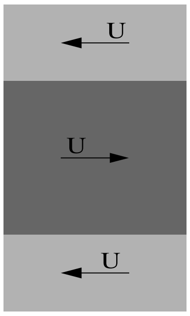
The results of the adaptive scheme are compared with the results of an analogous finite difference scheme (AB3,WENO5) on non-adapted, uniform grids with mesh sizes $h = 2^{-5}, ..., 2^{-9}$. In Figure 4 we compare the L^1- and the L^∞-error to the true solution after one rotation. The errors are plotted against the number of degrees of freedom (DOF) to point out the differences in the convergence behaviour. For the adaptive calculations, we observe a saturation effect. If the number of DOF is larger than a certain value, which depends on the maximal level of refinement, there is no further reduction of the error. Clearly, the adaptive calculation with maximal level L can not be more accurate than the non-adaptive calculation on a grid with mesh size 2^{-L}. In fact, the stagnation error for the adaptive runs with $L = 8$ and $L = 9$ is of the same order as the non-adaptive error for $h = 2^{-8}$ and 2^{-9}, respectively, compare Figure 4. However, apart from this saturation effect, the convergence rates for the adaptive scheme, 1.8 and 0.9, are twice as large as the convergence rates (0.9 and 0.45) for the non-adaptive scheme. Since the true solution is smooth, except for the bottom line of the cone, adaptive refinement is necessary in the vicinity of the cone's bottom, only. This is a one-dimensional region. In the non-adaptive case we uniformly refine the whole two-dimensional domain. This explains the doubling of the convergence rates. For the adaptive results an average number of DOF was used in Figure 4. This is justified, since the number of DOF was rather constant for all the time steps, compare Figure 6 (left).

Another important issue are spurious oscillations. Figure 5 shows close-ups of the numerical solutions after one rotation. Here, the non-adaptive scheme is in advantage. For the adaptive scheme, the oscillations are larger by a factor of about four. In our opinion this is due to the WENO stencils on non-uniform grids. Note however, that the oscillations are of the same order as for the non-adaptive scheme, even with the present adaptive scheme. As an example, in Figure 6 (right) the sizes of the largest undershoots for the finest adaptive calculation ($L = 9$, $\delta = 3 \cdot 10^{-4}$) and the finest non-adaptive calculation ($h = 2^{-9}$) are plotted against time for one complete rotation.

5 Three-dimensional Shear Flows

The multiscale-based finite difference scheme can be also used to solve the Navier-Stokes equations. Here, we employ a variant of Chorin's projection method with an explicit discretization of the convective terms in the momentum equation, and an implicit treatment of the diffusive terms. The Poisson equation for the pressure update is solved by an preconditioned BiCGstab scheme. To avoid spurious pressure oscillations we use special upwind/downwind discretizations of the divergence operator and the pressure gradient, respectively, see [10, 16] for details.

For our numerical experiment we consider a temporally evolving double shear layer with periodic boundary conditions on the domain $[0, 1] \times [0, 2] \times [0, 1]$. The initial condition are three parallel layers with opposite flow direction, see Figure 2. The maximal velocity is $U = 0.015$ and the viscosity is $\nu = 5.06 \cdot 10^{-6}$. Small perturbations were added in the initial shear layers to trigger Kelvin-Helmholtz instabilities. The parameters of the adaptive numerical scheme are shown in the right table in Figure 2. We used an automatic adjustment of the threshold parameter to keep the number of DOF below $3.5 \cdot 10^6$, because of memory limitations.



Interpolets	: order $P = 6$
convective term	: 5th order WENO
maximal refinement level	: 10
threshold parameter	: automatically adjusted
time discretization	: 4th order Runge-Kutta
time step	: $\Delta t = 0.01$

Figure 2 Left: initial configuration for the double shear layer problem. Right: parameters for the adaptive numerical simulation.

In the last decades, there has been considerable interest in the structures and statistics of turbulent shear layers, see [12]. For example, in [1] apparently two-dimensional rollers have been discovered in turbulent shear layers. Similar to two-dimensional flows, pairings of such rollers may occur along the span of a roller, see [15] for numerical results in the 2D case. Furthermore these structures are unstable to three-dimensional disturbances which lead to the formation of rib vortices in the braid regions between the rollers [19]. We observe such a behaviour also in in our numerical simulation. Figure 7 shows the isosurfaces of the vorticity for three different times steps. Cuts through the respective 3D adaptive grids are given in Figure 8. Obviously, the grid points are clustered in the vicinity of the coherent structures.

It is well known that a fully developed turbulent shear layer exhibits a self similar state, where, e.g., the momentum thickness defined by

$$\delta_m(t) := \int_y \left(\frac{1}{4} - \frac{\overline{u}(y,t)}{4U^2} \right) dy$$

$$\overline{u}(y,t) := \int_{x,z} u(x,y,z,t) dx dz \quad \text{(mean streamwise velocity)}$$

grows linearly [22]. In our simulations this was the case between $t = 60$ and $t = 100$, see Figure 3 (left). In the time span $t < 60$ the shear layer has to forget the initial conditions first, and the two-dimensional rollers must decay. Beyond $t > 100$ the two opposite shear layers start to interact and we can not expect a linear growth of δ_m then. One of the predictions of the statistical theory is that the profiles of the mean velocity $\overline{u}(.,t)$ are independent on time when plotted in self-similar coordinates $y/\delta_m(t)$. In fact, the profiles for our simulation collapse even for the larger time span $60 \le t \le 120$, see Figure 3 (right). A similar behaviour is predicted for, e.g., r.m.s. values. Here we encounter the problem that, even in the latest stages of the simulation, there were mostly large organized structures in the flow. Therefore, there are not enough samples of typical small eddies which are needed to achieve a well converged statistics. One of the reasons might be the relatively low Reynolds number related to the momentum thickness which was between 150 and 500. This is much smaller than, for example, in the numerical experiments reported in [19]. Another reason might be the implicit filtering operation in the mesh refinement step (8,9). There, we actually throw out small multiscale coefficients of the velocity which resembles a denoising of the current solution. The relation of this filtering and long term simulations and LES methods is an interesting field for future research.

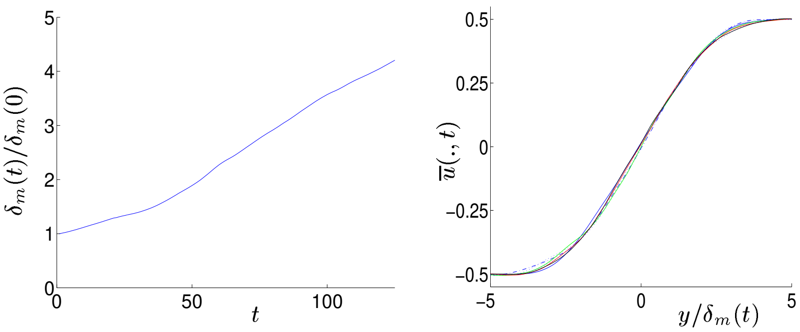


Figure 3 Left: Growth of the momentum thickness re-scaled by the initial momentum thickness. Left: Profiles of the mean velocity in self-similar coordinates for $t = 60, 70, ...120$.

References

[1] Brown G.L., Roshko A. (1974) On density effects and large structure in turbulent mixing layers. J. Fluid Mech. **64**.
[2] Cohen A., Daubechies I. et al. (1992) Biorthogonal bases of compactly supported wavelets. Comm. Pure Appl. Math. **45**, 485-560
[3] Dahmen W., Schneider R. et al. (1998) Nonlinear Functionals of Wavelet Expansions. IGPM, RWTH Aachen, to appear in Numerische Mathematik.
[4] Daubechies I. (1988) Orthonormal bases of compactly supported wavelets. Comm. Pure Appl. Math. **41**, 909–996
[5] Deslaurier G., Dubuc S. (1989) Symmetric iterative interpolation processes. Constr. Appr. **5**, 49-68
[6] DeVore R. (1999) Nonlinear Approximation. Acta Numerica **8**.
[7] Donoho D. (1992) Interpolating wavelet transform. Preprint Stanford University.
[8] Faber G. (1909) Über stetige Funktionen. Math. Annalen **66**, 81-94.
[9] Griebel M. (1998) Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences. Computing **61**, 151-180.
[10] Griebel M., Koster F. (2000) Adaptive Wavelet Solvers for the Unsteady Incompressible Navier-Stokes Equations. in Malek J., Rokyta M. (eds.), Advanced Mathematical Theories in Fluid Mechanics.
[11] Harten A. (1995) Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. Comm. Pure Appl. Math. **48**, 1305-1342.
[12] Ho C.-M., Huerre P. (1984) Perturbated free shear layers. Ann. Rev. Fluid Mech. **16**, 365-424.
[13] Hochmuth R. (1999) Wavelet Bases in Numerical Analysis and Restricted Nonlinear Approximation. Habilitationsschrift, Freie Universität Berlin.
[14] Koster F. (2000) A Proof of the Consistency of the Finite Difference Technique on Sparse Grids. Computing **65**, 247-261.
[15] Koster F., Schneider K., Griebel M., Farge M. (2000) Adaptive Wavelet Methods for the Navier-Stokes equations. in E.H. Hirschel (ed.) Notes on Numerical Fluid Mechanics.
[16] Koster F. (2001) Multiskalen-basierte Finite Differenzen Verfahren auf adaptiven Dünnen Gittern. PhD thesis, Universität Bonn.
[17] Koster F. (2001) Preconditioners for Sparse Grid Discretizations. Preprint SFB-256 Universität Bonn.
[18] Mallat S. (1989) A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans. Patt. Anal. and Mach. Intell. **7**, 674-693
[19] Rogers M.M., Moser R.D. (1993) Direct simulation of a self-similar turbulent mixing layer. Phys. Fluids **6**, 903-923.
[20] Schiekofer T. (1998) Die Methode der finiten Differenzen auf dünnen Gittern zur Lösung elliptischer und parabolischer PDEs. PhD thesis, Universität Bonn.
[21] Temlyakov V. (1993) Approximation of Periodic Functions. Nova Science Publishers.
[22] Townsend A.A. (1976) Structure of Turbulent Shear Flow. Cambridge University Press.
[23] Triebel H. (1992) Theory of function spaces II. Birkhäuser Verlag.
[24] Wesseling P. (2001) Principles of Computational Fluid Dynamics. Springer Verlag.
[25] Yserentant H. (1986) On the multilevel splitting of finite element spaces. Num. Math. **49**, 379-412.
[26] Zenger C. (1991) Sparse Grids. in Hackbusch W. (ed.) Notes on Numerical Fluid Mechanics **31**.

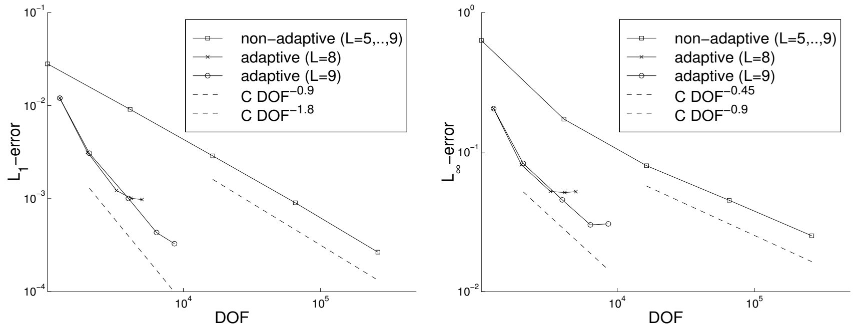


Figure 4 L^1 and L^∞-error after one rotation for the convection problem. In the adaptive case, the maximal level of refinement was limited to $L = 8$ or $L = 9$. This causes the stagnation of the error.

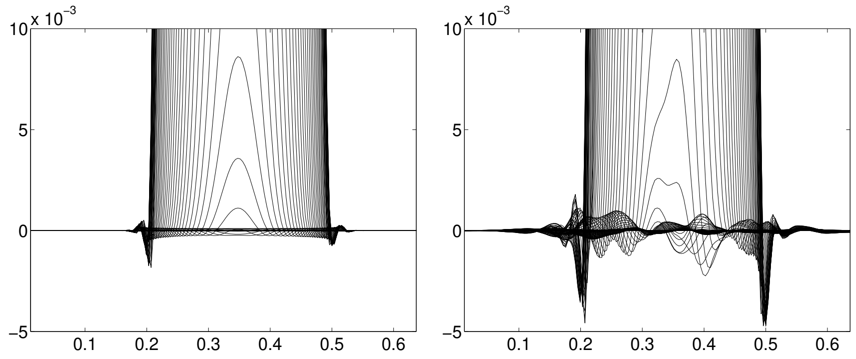


Figure 5 Spurious oscillations in the numerical solution after one rotation. Left: non-adaptive. Right: adaptive scheme.

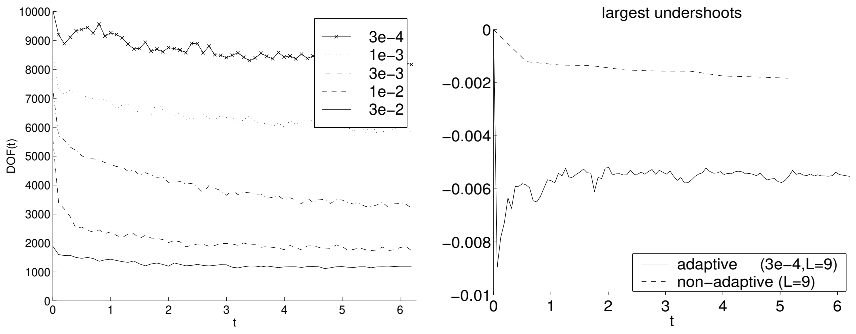


Figure 6 Left: number of DOF vs. time. Right: largest undershoots vs. time for the finest adaptive and the finest non-adaptive calculations.

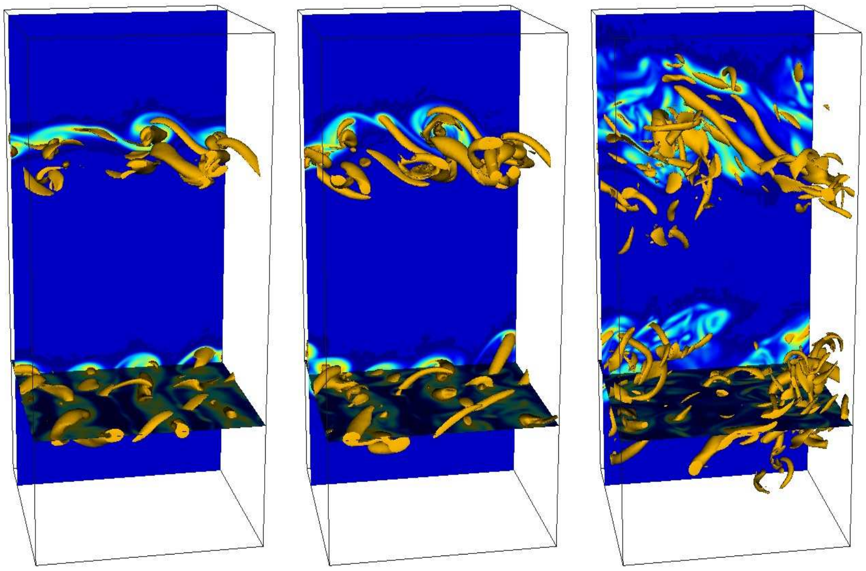


Figure 7 Isosurfaces of vorticity at $t = 20, 35$ and 125.

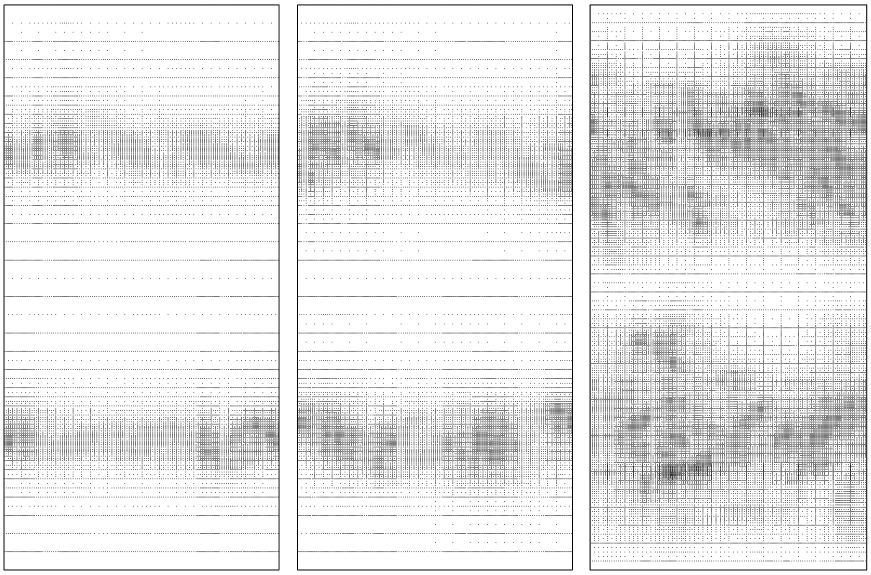


Figure 8 Cuts ($z = 0.5$) through adaptive grids at $t = 20, 35$ and 125.