Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 197 (2004) 460–490

JOURNAL OF
COMPUTATIONAL
PHYSICS

www.elsevier.com/locate/jcp

An adaptive multiscale finite volume solver for unsteady and steady state flow computations [☆]

F. Bramkamp [a], Ph. Lamby [b], S. Müller [b,*]

[a] *Lehr- und Forschungsgebiet für Mechanik, RWTH Aachen, D-52056 Aachen, Germany*
[b] *Institut für Geometrie und Praktische Mathematik, RWTH Aachen, D-52056 Aachen, Germany*

Received 6 March 2003; received in revised form 5 December 2003; accepted 5 December 2003
Available online 23 January 2004

Abstract

In this paper we present the main conceptual ingredients and the current state of development of the new solver QUADFLOW for large scale simulations of compressible fluid flow and fluid–structure interaction. In order to keep the size of the discrete problems at every stage as small as possible, we employ a multiresolution adaptation strategy that will be described in the first part of the paper. In the second part we outline a new mesh generation concept that is to support the adaptive concepts as well as possible. A key idea is to understand meshes as parametric mappings determined by possibly few control points as opposed to store each mesh cell separately. Finally, we present a finite volume discretization which again is to support the adaptation concepts. We conclude with numerical examples of realistic applications demonstrating different features of the solver.

---

1. Introduction

Due to the increasing computer power more and more realistic and consequently more complex models have become tractable by numerical simulation. Studying the interaction of aerodynamics and structural dynamics is a typical example. Here several severe obstructions, such as time-dependency of the involved processes, varying complex geometries and the coupling of physical regimes with different characteristic features come together. In order to resolve a typically singular behavior of the solution meshes with several millions of cells are required. However, improved hardware or purely data oriented strategies such as parallel computing are not sufficient to overcome the arising difficulties. As important and necessary these

---

[☆] This work has been performed with funding by the Deutsche Forschungsgemeinschaft in the Collaborative Research Centre SFB 401 'Flow Modulation and Fluid–Structure Interaction at Airplane Wings' of the RWTH Aachen, University of Technology, Aachen, Germany.

[*] Corresponding author. Tel.: +49-241-806469.
E-mail address: mueller@igpm.rwth-aachen.de (S. Müller).

0021-9991/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2003.12.005

aspects may be they have to be complemented in the long run by mathematical concepts that aim at minimizing in the first place the size of arising discrete problems.

This paper summarizes some recent attempts in this direction. We present an *integral concept* for designing a finite volume solver for compressible flow computations. The three main components of this new solver QUADFLOW consist of (i) a standard finite volume discretization for arbitrary grid topologies, (ii) a block-structured grid generation using parametric mappings based on B-splines and (iii) a local grid adaptation based on a local multiscale analysis. These tools are appropriately adjusted to each others needs. Here the core ingredient is the adaptation strategy that is based on a mathematically well-founded concept. The ultimate goal of the present work is to verify that this concept is no longer confined to academic problems on Cartesian grids only but has become mature. In particular, it can be employed for the investigation of problems arising, for instance, in aerodynamics.

In the literature, several adaptive strategies have been discussed or are under current investigation. A standard strategy is to base local mesh refinements on *local indicators* which are typically related to gradients in the flow field, see [13,14], or local residuals, see [48,66,67]. Although these concepts turn out to be very efficient in practice they offer no reliable error control. For this purpose, *a posteriori estimates* have been derived which aim at equilibrating local errors. So far, this type of error estimator is only available for scalar problems, see [50]. In the present work, however, we employ recent *multiresolution techniques*. The starting point is a proposal by Harten [45] to transform the arrays of cell averages associated with any given finite volume discretization into a different format that reveals insight into the local behavior of the solution. The cell averages on a given highest level of resolution (reference mesh) are represented as cell averages on some coarse level where the fine scale information is encoded in arrays of *detail coefficients* of ascending resolution. This information is essentially used in Harten's original strategy to gain computational time by avoiding expensive flux evaluations in regions where the solution is smooth. Instead cheap finite differences are employed in major parts of the domain. The solution remains within the same accuracy as the *reference scheme*, i.e., the scheme on the finest computational mesh that uses the expensive flux evaluation throughout the entire domain. Successful implementations of this strategy have been carried out for two-dimensional Cartesian meshes [15,16,26,27,62], curvilinear meshes [33] and unstructured meshes [2,17,30]. However, since one works still on a uniform mesh the computational complexity stays proportional to the number of cells on the finest grid which in 3D computations with the above objectives is prohibited.

In contrast to this, the *detail coefficients* will be used here to create *locally refined meshes* on which the discretization is performed. Of course, the crux in this context is to arrange this procedure in such a way that at no stage of the computation there is ever made use of the fully refined uniform mesh. A central mathematical problem is then to show that the solution on the adapted mesh is of the same accuracy as the solution on the reference mesh. This genuine *adaptive* approach has been presented in [42] and has been investigated in [31]. A self-contained account of the adaptive concept for conservation laws can be found in [59]. By now the new adaptive multiresolution concept has been employed by several groups with great success to different applications, see [23,32,58,63].

The adaptive concept is based on a *hierarchy of meshes*. This requires a new grid generation strategy. Accepting the Navier–Stokes equations as the model of choice we give preference to quadrilateral and hexahedral cells that still facilitate best boundary fitted anisotropic meshes. Local refinement gives rise to meshes of quadtree and octree type, respectively. A key idea is to represent such meshes as parametric mapping from the computational domain into the physical domain. In this way one overcomes the restriction to Cartesian meshes as employed in the literature mentioned above. Such mappings can be realized using well established concepts from computer aided geometric design (CAGD), see for instance [19,39]. To retain sufficient geometric flexibility this is combined with block structuring.

Finally, one needs a discretization scheme that meets the requirements of the adaptation concept and fits well with the mesh generation. The adaptive method crucially depends on the assumption, that the fluxes of the underlying discretization are *conservative*. Furthermore, to avoid complicated mesh management within each block and to keep the discretizations of the individual blocks as independent as possible and, in particular, to avoid global geometrical constraints, we insist on meshes with hanging nodes. Both requirements are met by the development of a finite volume scheme that can cope, in particular, with fairly general cell partitions.

We organize the remainder of this paper as follows: Section 2 gives a short description of the governing equations as used in the present work. Section 3 is the essential part of the paper describing the multiscale analysis and the grid adaptation algorithm. Section 4 is concerned with the representation and generation of parametric meshes using B-spline methods. Section 5 offers a self-contained account of the discretization scheme including the realization of spatial second order, the choice of limiters, the treatment of convective and viscous fluxes and the time integration. In Section 6 we present several applications to well-known fluid dynamical test cases that highlight the features of the whole flow solver. We conclude this paper in Section 7 with some remarks on future developments.

2. Governing equations

In the present study, laminar viscous fluid flow is described by the Navier–Stokes equations for a compressible gas. In order to solve problems in time dependent domains, including moving boundaries, we consider the governing equations in its arbitrary Lagrangian Eulerian (ALE) formulation. Neglecting body forces and volume supply of energy, the conservation laws for any control volume V with boundary ∂V and outward unit normal vector $\mathbf{n}$ on the surface element $\mathrm{d}S \subset \partial V$ can be written in integral form as:

$$\frac{\partial}{\partial t}\int_{V(t)} \mathbf{u}\,\mathrm{d}V + \oint_{\partial V(t)} \left(\mathbf{F}^{\mathrm{c}}(\mathbf{u}) - \mathbf{F}^{\mathrm{d}}(\mathbf{u})\right)\mathbf{n}\,\mathrm{d}S = \mathbf{0}. \tag{1}$$

To complete the posed problem initial values $\mathbf{u}(\mathbf{x}, t_0) = \mathbf{u}_0(\mathbf{x})$, $\mathbf{x} \in V$ and boundary conditions $\mathbf{u}(\mathbf{x}, t)|_{\partial V} = B(\mathbf{x}, t)$, $\mathbf{x} \in \partial V$ are to be prescribed.

Here $\mathbf{u} = (\varrho,\ \varrho\mathbf{v},\ \varrho e_{\mathrm{tot}})^{\mathrm{T}}$ denotes the vector of the unknown conserved quantities and $\mathbf{F}^{\mathrm{c}}$ and $\mathbf{F}^{\mathrm{d}}$ represent the convective flux including pressure and the diffusive flux function, respectively:

$$\mathbf{F}^{\mathrm{c}} = \begin{pmatrix} \varrho\mathbf{v}_r \\ \varrho\mathbf{v}_r \circ \mathbf{v} + p\mathbf{I} \\ \varrho e_{\mathrm{tot}}\mathbf{v}_r + p\mathbf{v} \end{pmatrix}, \quad \mathbf{F}^{\mathrm{d}} = \begin{pmatrix} 0 \\ \mathbf{T}^{\mathrm{v}} \\ \mathbf{v}\mathbf{T}^{\mathrm{v}} - \mathbf{q} \end{pmatrix}, \tag{2}$$

where ϱ denotes the density, p the static pressure, $\mathbf{v}$ the velocity vector of the fluid and e_{tot} the total energy. The motion of the grid is considered by the convective fluxes, where $\mathbf{v}_r = \mathbf{v} - \dot{\mathbf{x}}$ expresses the relative velocity between the fluid and the grid velocity $\dot{\mathbf{x}}$. The symbol $\circ$ means the dyadic product. The viscous stress tensor $\mathbf{T}^{\mathrm{v}}$ for an isentropic Newtonian fluid is defined as

$$\mathbf{T}^{\mathrm{v}} = \mu\left(\operatorname{grad}\mathbf{v} + (\operatorname{grad}\mathbf{v})^{\mathrm{T}}\right) - \frac{2}{3}\mu(\operatorname{div}\mathbf{v})\mathbf{I}. \tag{3}$$

Heat conduction is modeled by Fourier's law $\mathbf{q} = -\kappa \operatorname{grad} T$, where the thermal conductivity is assumed as $\kappa = c_p\mu/Pr$, with Prandtl number $Pr = 0.72$. The variation of the molecular viscosity μ as a function of temperature is determined by the Sutherland formula. The static pressure is related to the specific internal energy according to the equation of state for a perfect gas $p = \varrho(\gamma - 1)(e_{\mathrm{tot}} - 1/2|\mathbf{v}|^2)$, where γ is the ratio of specific heats, which is taken as 1.4 for air.

3. Grid adaptation concept

In this section we outline the grid adaptation concept. It is based on a multiscale analysis of an array of cell averages. For this purpose we first summarize the multiscale setting. Finally we explain how to perform an adaptive mesh refinement employing the multiscale analysis.

3.1. Multiscale setting

A finite volume discretization typically works on a sequence of cell averages. In order to analyze the local regularity of the data we employ the methodology developed by Dahmen et al. [25] which is based on biorthogonal wavelets and stable completions. In view of stability investigations, function spaces are introduced in order to benefit from functional analytic arguments. This approach may be seen as a natural generalization of Harten's discrete framework [4,5,46]. It provides the analytical tools, e.g., inverse estimates, with that rigorous estimates of the perturbation error can be derived, see [31,59]. From a discrete point of view both concepts are directly related as has been shown in [43].

3.1.1. Hierarchy of meshes

First of all we introduce a sequence of grids $\mathcal{G}_j := \{V_{j,k}\}_{k\in I_j}$, $j = 0,\dots,L$, where the index set I_j represents the enumeration of the cells corresponding to the grid $\mathcal{G}_j$. The coarsest grid is indicated by $j = 0$ and the finest grid by $j = L$, respectively. A simple example is shown in Fig. 1 where a coarse grid is successively refined with increasing refinement level. This sequence is called a *nested grid hierarchy*, if (i) each grid represents a partition of the computational domain Ω and (ii) each cell $V_{j,k}$ can be decomposed into cells $V_{j+1,r}$ on the next finer resolution level determined by the index set $\mathcal{M}_{j,k} \subset I_{j+1}$.

Note that the framework presented here is not restricted to this simple configuration but can also be applied to *unstructured* grids and *irregular* grid refinements. Furthermore we assume without loss of generality that the computational domain and, hence, the grid hierarchy is time-independent throughout this section. Since the data analysis is performed by means of cell averages corresponding to a fixed but arbitrary time level, this is no constraint of the concept. In case of time-varying boundaries the grid hierarchy has to be adjusted after each time step.

Relative to the partitions $\mathcal{G}_j$ we introduce the so-called *box function*

$$\tilde{\varphi}_{j,k}(\boldsymbol{x}) := \frac{1}{|V_{j,k}|}\chi_{V_{j,k}}(\boldsymbol{x}) = \begin{cases} 1/|V_{j,k}|, & \boldsymbol{x} \in V_{j,k}, \\ 0, & \boldsymbol{x} \notin V_{j,k}, \end{cases} \tag{4}$$

defined as the L^1–scaled characteristic function with respect to $V_{j,k}$. By $|V|$ we denote the volume of a cell V. Then the cell average of a scalar, integrable function $u \in L^1(\Omega)$ can be interpreted as an inner product, i.e.,

$$\hat{u}_{j,k} := \langle u, \tilde{\varphi}_{j,k}\rangle_\Omega \quad \text{with} \quad \langle u, v\rangle_\Omega := \int_\Omega u v \,\mathrm{d}\boldsymbol{x}. \tag{5}$$

Obviously the nestedness of the grids as well as the linearity of integration imply the two-scale relations

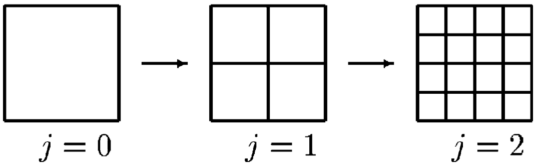


Fig. 1. Sequence of nested grids.

$$\tilde{\varphi}_{j,k} = \sum_{r\in\mathcal{M}_{j,k}} \frac{|V_{j+1,r}|}{|V_{j,k}|}\,\tilde{\varphi}_{j+1,r} \quad\text{and}\quad \hat{u}_{j,k} = \sum_{r\in\mathcal{M}_{j,k}} \frac{|V_{j+1,r}|}{|V_{j,k}|}\,\hat{u}_{j+1,r}. \tag{6}$$

The goal is to transform these data into a different format of cell averages corresponding to a sequence of resolution levels. This will be motivated by a simple univariate example.

3.1.2. Motivation: A univariate example

Haar basis. We now consider the unit interval $\Omega = [0,1]$ where the grid hierarchy is determined by a uniform dyadic partition of $[0,1]$, i.e., $V_{j,k} = 2^{-j}[k, k+1]$, $k \in I_j := \{0,\ldots,2^j-1\}$. Note that the refinement sets are $\mathcal{M}_{j,k} = \{2k, 2k+1\}$. Then the L^1-scaled box function (4) has the form

$$\tilde{\varphi}_{j,k}(x) = 2^j\,\chi_{[2^{-j}k,2^{-j}(k+1)]}(x),$$

and the two-scale relations (6) read

$$\tilde{\varphi}_{j,k} = \frac{1}{2}\,(\tilde{\varphi}_{j+1,2k} + \tilde{\varphi}_{j+1,2k+1}) \quad\text{and}\quad \hat{u}_{j,k} = \frac{1}{2}\,(\hat{u}_{j+1,2k} + \hat{u}_{j+1,2k+1}). \tag{7}$$

Introducing the box wavelet $\tilde{\psi}_{j,k}$ and the details $d_{j,k}$ as

$$\tilde{\psi}_{j,k} := \frac{1}{2}\,(\tilde{\varphi}_{j+1,2k} - \tilde{\varphi}_{j+1,2k+1}),\; d_{j,k} := \langle u, \tilde{\psi}_{j,k}\rangle_{[0,1]} = \frac{1}{2}\,(\hat{u}_{j+1,2k} - \hat{u}_{j+1,2k+1}), \tag{8}$$

any fine scale box function can be expressed by means of the box function $\tilde{\varphi}_{j,k}$ and the box wavelet $\tilde{\psi}_{j,k}$, i.e.,

$$\tilde{\varphi}_{j+1,2k} = \tilde{\varphi}_{j,k} + \tilde{\psi}_{j,k}, \quad \tilde{\varphi}_{j+1,2k+1} = \tilde{\varphi}_{j,k} - \tilde{\psi}_{j,k}. \tag{9}$$

This is illustrated in Fig. 2. Thus we can compute fine scale averages from coarse scale ones and details, i.e.,

$$\hat{u}_{j+1,2k} = \hat{u}_{j,k} + d_{j,k}, \quad \hat{u}_{j+1,2k+1} = \hat{u}_{j,k} - d_{j,k}. \tag{10}$$

To simplify notation we define the vectors $\tilde{\Phi}_j := (\tilde{\varphi}_{j,k})_{k\in I_j}$ and $\tilde{\Psi}_j := (\tilde{\psi}_{j,k})_{k\in I_j}$. Later we will use this notation also in the sense of a collection of functions.

Biorthogonal wavelets. We now introduce a dual system by the functions

$$\varphi_{j,k} := 2^{-j}\,\tilde{\varphi}_{j,k} = \chi_{[0,1]}(2^j\cdot - k), \quad \psi_{j,k} := 2^{-j}\,\tilde{\psi}_{j,k},$$

or in vector form $\Phi_j := 2^{-j}\,\tilde{\Phi}_j$ and $\Psi_j := 2^{-j}\,\tilde{\Psi}_j$. These are the L^∞-normalized counterparts of the box function and the box wavelet, respectively. Obviously, the duals also satisfy two-scale relations of the form (7)–(9). From this we infer that $\tilde{\Phi}_j \cup \tilde{\Psi}_j$ and $\Phi_j \cup \Psi_j$ are *biorthogonal*, i.e.,

$$\langle \Phi_j, \tilde{\Phi}_j\rangle_{[0,1]} = \langle \Psi_j, \tilde{\Psi}_j\rangle_{[0,1]} = \boldsymbol{I},\; \langle \Phi_j, \tilde{\Psi}_j\rangle_{[0,1]} = \langle \Psi_j, \tilde{\Phi}_j\rangle_{[0,1]} = \boldsymbol{0}, \tag{11}$$

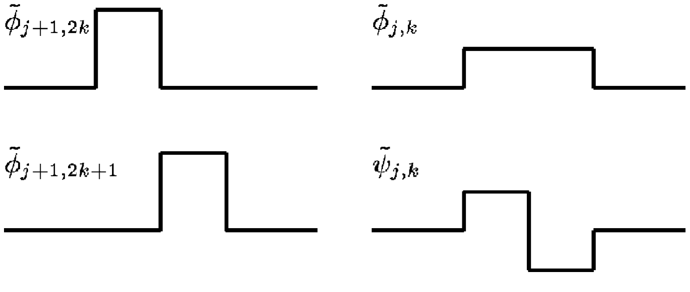


Fig. 2. Box function and box wavelet.

where we use the notation $\langle \Theta, \Phi \rangle := (\langle \theta, \varphi \rangle)_{\theta \in \Theta, \varphi \in \Phi}$ and $\boldsymbol{I}$ denotes the unity matrix and $\boldsymbol{0}$ the zero matrix, respectively. The basis functions Φ_j and Ψ_j are always referred to as *primal scaling functions* and *primal wavelets*, respectively. Analogously, the basis functions $\tilde{\Phi}_j$ and $\tilde{\Psi}_j$ are called the *dual scaling functions* and *dual wavelets*, respectively.

Change of basis. By means of the box function and the box wavelet as well as their L^∞-normalized counterparts we now introduce a function on the unit interval that is naturally related to the array of cell averages. To this end we consider a function $u \in L^1([0,1])$. Then the projection of u onto piecewise constants with respect to the refinement level j is determined by

$$u_j := \sum_{k \in I_j} \langle u, \tilde{\varphi}_{j,k} \rangle_{[0,1]} \; \varphi_{j,k} =: \Phi_j^{\mathrm{T}} \; \hat{\boldsymbol{u}}_j. \tag{12}$$

Equivalently the function u_j can be represented by

$$u_j = \Phi_{j-1}^{\mathrm{T}} \; \hat{\boldsymbol{u}}_{j-1} + \Psi_{j-1}^{\mathrm{T}} \; \boldsymbol{d}_{j-1}, \tag{13}$$

because the two-scale relations realize a change of basis and, in particular, the systems $\Phi_j \cup \Psi_j$ and $\tilde{\Phi}_j \cup \tilde{\Psi}_j$ are biorthogonal. This representation motivates that the details can be interpreted as the update when progressing to a higher resolution level.

Cancellation property. We now have to explain why the representation (13) is preferable to (12) for our purposes. To this end, we verify that the details become small when the underlying function is smooth. First of all, we conclude from (8) and (4) that

$$\langle 1, \tilde{\psi}_{j,k} \rangle_{[0,1]} = 0.$$

Since the box wavelets are L^1-normalized, i.e., $\|\tilde{\psi}_{j,k}\|_{L^1([0,1])} = 1$, we deduce

$$|d_{j,k}| \leqslant \inf_{c \in \mathbb{R}} \; |\langle u - c, \tilde{\psi}_{j,k} \rangle_{[0,1]}| \leqslant \inf_{c \in \mathbb{R}} \; \|u - c\|_{L^\infty(V_{j,k})} \leqslant C \; 2^{-j} \; \|u'\|_{L^\infty(V_{j,k})}, \tag{14}$$

i.e., the details decay with a rate at least of 2^{-j} provided the function u is differentiable. Hence the details may become small with increasing refinement level provided that the gradient of u is bounded. This motivates to neglect all sufficiently small details in order to compress the original data such that we control the loss of accuracy. In order to realize higher compression rates it will be convenient to improve the decay 2^{-j}. For this purpose not only constants have to be canceled by the wavelet $\tilde{\psi}_{j,k}$ but also all polynomials p of a fixed higher degree, i.e., $\langle p, \tilde{\psi}_{j,k} \rangle_{[0,1]} = 0$. This can be achieved by means of higher order biorthogonal systems instead of piecewise constants, see [29].

3.1.3. Multiscale transformation

The ultimate goal is to transform an array of cell averages corresponding to a finest uniform discretization level into a different format in order to compress data. This is achieved by means of a change of basis. For this purpose we introduce the vectors

$$\hat{\boldsymbol{u}}_j := (\hat{u}_{j,k})_{k \in I_j} \quad \text{and} \quad \boldsymbol{d}_j := (d_{j,k})_{k \in I_j},$$

where the averages and the details are determined by functionals of a function $u \in L^1(\Omega)$, i.e., $\hat{u}_{j,k} := \langle u, \tilde{\varphi}_{j,k} \rangle_\Omega$ and $d_{j,k} := \langle u, \tilde{\psi}_{j,k} \rangle_\Omega$. Note that in the multidimensional case there is not only one detail related to a cell $V_{j,k}$ as in the univariate example. In the multivariate case the number of wavelet types corresponds to the number of subcells $\mathcal{M}_{j,k}$ reduced by one for the box function, i.e., $\#\mathcal{M}_{j,k} - 1$. For readers who are not familiar with multivariate wavelets we summarize the wavelet construction for a Cartesian grid hierarchy, see Appendix A. For sake of simplicity we suppress the wavelet type in the following but keep in mind that $d_{j,k}$ and $\tilde{\psi}_{j,k}$, respectively, represent a sequence of details and wavelet functions. According to (7) and (10) we obtain two-scale relations for the coefficients inherited from the two-scale relations of the box functions and the wavelet functions. These can be written in vector–matrix representation as

$$\hat{\boldsymbol{u}}_{j+1} = \boldsymbol{G}_j \begin{pmatrix} \hat{\boldsymbol{u}}_j \\ \boldsymbol{d}_j \end{pmatrix}, \quad \begin{pmatrix} \hat{\boldsymbol{u}}_j \\ \boldsymbol{d}_j \end{pmatrix} = \boldsymbol{M}_j \hat{\boldsymbol{u}}_{j+1}, \tag{15}$$

where $\boldsymbol{G}_j$ and $\boldsymbol{M}_j$ denote the mask matrices of the filter coefficients of $\tilde{\varphi}_{j,k}$, $\tilde{\psi}_{j,k}$ and $\tilde{\varphi}_{j+1,k}$, respectively, see [59]. Applying these relations iteratively, see Fig. 3, the array $\hat{\boldsymbol{u}}_L$ of cell averages on level L can be decomposed into a sequence of coarse scale cell averages $\hat{\boldsymbol{u}}_0$ and details $\boldsymbol{d}_j$, $j = 0, \ldots, L-1$. We refer to this transformation as *multiscale transformation*. It is reversed by the *inverse multiscale transformation*. If there is a biorthogonal system $\Phi_0 \cup \Psi_0 \cup, \ldots, \cup \Psi_{L-1}$ to the multiscale basis $\tilde{\Phi}_0 \cup \tilde{\Psi}_0 \cup, \ldots, \cup \tilde{\Psi}_{L-1}$, then the multiscale transformation realizes the change of basis, i.e.,

$$u_L := \Phi_L^{\mathrm{T}} \hat{\boldsymbol{u}}_L = \Phi_0^{\mathrm{T}} \hat{\boldsymbol{u}}_0 + \sum_{j=0}^{L-1} \Psi_j^{\mathrm{T}} \boldsymbol{d}_j. \tag{16}$$

Here u_L is the projection of any function $u \in L^1(\Omega)$ onto the space spanned by the basis Φ_L. Note that we will actually construct only the bases $\tilde{\Phi}_j$ and $\tilde{\Psi}_j$ that are dual in the above sense to process the cell averages. The corresponding primal bases will enter only the analysis but will never be needed explicitly for algorithmic realizations. At this point we emphasize that the existence of a primal wavelet basis is essential for proving the reliability of the adaptive finite volume scheme. For instance, we need inverse estimates, see [28], which are not directly accessible in Harten's discrete framework.

3.2. Grid adaptation

We will now summarize the six steps of the local grid refinement procedure, namely, (i) *local multiscale transformation*, (ii) *thresholding*, (iii) *prediction*, (iv) *grading*, (v) *local grid refinement* and (vi) *local inverse multiscale transformation*. Here the ultimate goal is to provide an algorithm that can be realized with an optimal complexity, i.e., the number of floating point operations is proportional to the number of cells in the adaptive grid. In particular, we never access to the *finest* mesh.

Since the grid adaptation tool is supposed to dynamically adapt the mesh to an underlying flow field, we start with data corresponding to a certain time step n. At this time step the locally refined grid is characterized by the index set $\mathcal{G}_{L,\varepsilon}^n \subset \{(j,k);\ k \in I_j, j = 0, \ldots, L\}$, i.e.,

$$\Omega = \bigcup_{(j,k) \in \mathcal{G}_{L,\varepsilon}^n} V_{j,k},$$

see, for instance, Fig. 4. It is required that the set $\mathcal{G}_{L,\varepsilon}^n$ has the structure of a graded tree, i.e., neighboring cells differ at most by one level of refinement. The grid is provided with cell averages $\{\hat{u}_{j,k}^n\}_{(j,k) \in \mathcal{G}_{L,\varepsilon}}$.

Local multiscale transformation. In a first step we perform a multiscale analysis of the data at hand. For this purpose we proceed level by level from fine to coarse similar to (7) and (8). Note that the two-scale transformation is performed locally only for the indices corresponding to the adaptive grid instead of the full levels. In particular, applying the local two-scale transformation can be interpreted as a successive

$$\begin{tikzcd}
\hat{\boldsymbol{u}}_L \arrow[r, leftrightarrow] \arrow[dr, leftrightarrow] & \hat{\boldsymbol{u}}_{L-1} \arrow[r, leftrightarrow] \arrow[dr, leftrightarrow] & \cdots \arrow[r, leftrightarrow] \arrow[dr, leftrightarrow] & \hat{\boldsymbol{u}}_1 \arrow[r, leftrightarrow] \arrow[dr, leftrightarrow] & \hat{\boldsymbol{u}}_0 \\
 & \boldsymbol{d}_{L-1} & \cdots & \boldsymbol{d}_1 & \boldsymbol{d}_0
\end{tikzcd}$$

Fig. 3. Multiscale transformation.

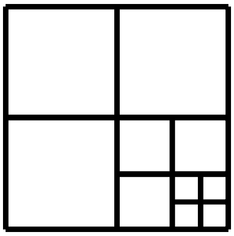


Fig. 4. Locally refined grid.

coarsening of the grid where fine-grid cells are agglomerated to a coarse-grid cell and the difference information is stored by the detail coefficients.

Thresholding. The idea is simply to discard all coefficients $d_{j,k}$ that fall in absolute value below a certain threshold. For this purpose, we introduce the index set

$$\mathscr{D}_{L,\varepsilon} := \left\{(j,k); \quad |d_{j,k}| > \varepsilon_j, \quad k \in I_j, j \in \{0,\ldots,L-1\}\right\},$$

corresponding to what will be referred to as *significant details*. Here $\varepsilon_j = 2^{j-L}\varepsilon$ is a level-dependent threshold value which is smaller on coarser levels. The choice of the threshold parameter ε is discussed in [59]. The ideal strategy would be to balance the discretization error of the reference scheme on the uniform finest grid with the perturbation error introduced by the thresholding. Note that the thresholding procedure is slightly modified for vector-valued functions arising in systems of conservation laws. Here we determine for each conserved quantity its absolute global maximum and scale the corresponding details by that. Then a cell is refined if there is at least one significant scaled detail corresponding to at least one of the conserved quantities.

Prediction. So far the multiscale analysis does not take into account the underlying problem, namely, the evolution equations. The current prediction step plays the role of an interface between the data analysis and the problem at hand by which the data are produced. To perform the evolution step, we have to determine the adaptive grid on the *new* time level. Since the corresponding averages, respectively details are not yet available, we have to *predict* all details on the new time level $n+1$ that may become significant due to the evolution by means of the details on the *old* time level n. In order to guarantee the adaptive scheme to be *reliable* in the sense that no significant future feature of the solution is missed, the prediction set $\tilde{\mathscr{D}}^{n+1}_{\varepsilon}$ has to satisfy

$$\mathscr{D}^{n}_{L,\varepsilon} \cup \mathscr{D}^{n+1}_{L,\varepsilon} \subset \tilde{\mathscr{D}}^{n+1}_{L,\varepsilon}, \tag{17}$$

where, of course $\mathscr{D}^{n+1}_{L,\varepsilon}$ is not known at the old time level. In [45] Harten suggests a heuristic approach taking into account that (i) details in a local neighborhood of a significant detail may also become significant within one time step due to the finite speed of propagation and (ii) gradients may become steeper causing significant details on a higher refinement level due to the developing of discontinuities. So far Harten's approach could not be rigorously verified to satisfy (17). However, a slight modification of Harten's prediction strategy has recently been shown to lead to a reliable prediction strategy in the sense of (17), at least for a certain class of *explicit* finite volume schemes applied to *one-dimensional scalar* conservation laws on *uniform dyadic grids* as base hierarchies, see [31]. So far an analogue for implicit schemes is not available yet.

Grading. In order to perform the grid adaptation procedure level by level we need that the index set of significant details corresponds to a *graded tree*, i.e., the levels of neighboring cells differ at most by one. Since the sets $\mathscr{D}_{L,\varepsilon}$ and $\mathscr{D}^{n+1}_{L,\varepsilon}$, respectively, are in general not graded, we have to apply in addition a grading procedure. This will slightly inflate the index set of significant details but has so far been observed not to spoil the complexity reduction in any significant way. In fact, from the nature of singularities occurring in

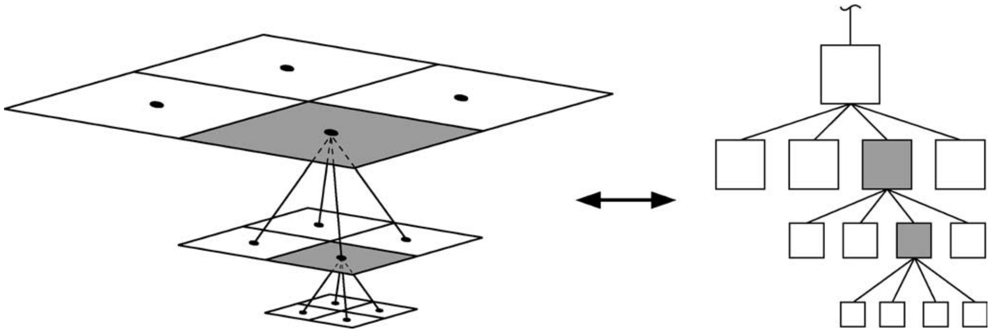


Fig. 5. Grid adaptation.

flow computations one expects the distribution of significant wavelet coefficients to exhibit at least nearly tree structure. If a high level wavelet overlaps a discontinuity this will be seen also by its coarser ancestors in the same region. For details on the grading procedure we refer to [59].

Grid adaptation. Then we exploit the inflated set $\tilde{\mathscr{D}}^{n+1}_{L,\varepsilon}$ to determine an associated index set $\mathscr{G}^{n+1}_{L,\varepsilon}$ which characterizes the adaptive grid at the new time level. The index set $\mathscr{G}^{n+1}_{L,\varepsilon}$ is initialized by all indices of the coarsest discretization. Then, traversing through the levels from coarse to fine we proceed as follows: if $(j,k)\in\tilde{\mathscr{D}}^{n+1}_{L,\varepsilon}$ then the cell $V_{j,k}$ is locally refined, i.e., the index (j,k) is removed from $\mathscr{G}^{n+1}_{L,\varepsilon}$ and the indices of the subcells on the finer level are added to $\mathscr{G}^{n+1}_{L,\varepsilon}$. Finally we obtain the locally adapted grid which naturally corresponds to the leaves of the graded tree of significant details. In Fig. 5 this procedure is described schematically. We note that the locally adapted grid naturally corresponds to the leaves of the graded tree of significant details characterized by the shaded cells.

Local inverse multiscale transformation. By the previous step the grid has locally changed due to local refinement and coarsening. In order to determine the cell averages $\{\hat{u}^n_{j,k}\}_{(j,k)\in\mathscr{G}^{n+1}_{L,\varepsilon}}$, we employ a local inverse multiscale transformation similar to (10) interrelating the local cell averages $\{\hat{u}^n_{j,k}\}_{(j,k)\in\mathscr{G}^{n}_{L,\varepsilon}}$ and the significant details $\{d^n_{j,k}\}_{(j,k)\in\mathscr{D}_{L,\varepsilon}}$. Again we proceed level by level from coarse to fine where we locally replace a cell average on the coarse scale by the cell averages of its subcells. This is done whenever there is a significant detail associated to this coarse cell in $\tilde{\mathscr{D}}^{n+1}_{L,\varepsilon}$. Note that the computation of these cell averages can be simultaneously determined when performing the grid adaptation.

4. Mesh generation

4.1. Parametric meshes

The multiscale setting outlined in Section 3 is based on a *hierarchy of nested grids*. From this point of view the most natural way to discretize the flow domain would be to employ adaptive Cartesian grids. On the other hand it is widely accepted that boundary conforming meshes are preferable for the discretization of viscous flows because they facilitate best the generation of anisotropic grid cells that are necessary for a stable and accurate resolution of boundary layers. In this approach one has to observe, however, that different levels of discretization by hexahedral cells do not result in a hierarchy of nested grids, since at least at a curved boundary the fine grid points do not lie on the faces of the coarse grid cells. Indeed it can be shown that the notion of a nested grid hierarchy is equivalent to the requirement that the grid can be described analytically by an invertible *parametric mapping* from a logical space to the physical domain, see Fig. 6.

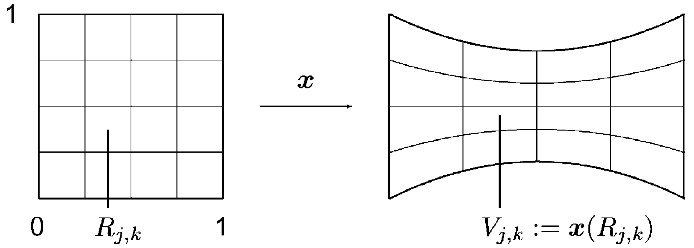


Fig. 6. Parametric mappings.

In this setting grid cells are the images of the corresponding cells in logical space. Grid refinement can be interpreted simply as function evaluation. Furthermore the grid generation process can completely be separated from the discretization process because the grid generator needs only to provide a (possibly sparse) representation of the grid function. In particular, this is convenient in the case of moving grids.

The main costs of this method stem from the fact that the cells may have curved edges now. This complicates the computation of some geometric quantities that are needed by the finite volume scheme, for instance, cell volumes and centroids. They have to be computed accurately using quadrature formulas.

4.2. B-spline-representations

For the representation of the curvilinear coordinate systems we use patches of tensor product B-splines. In the above terminology this means that we choose the unit cube as logical space and in order to enhance the flexibility embed several of such mappings into a multiblock concept. B-splines seem to be a very appropriate tool for this task, because they possess excellent approximation properties. Modeling with B-splines is intuitive and the evaluation of B-splines is fast and numerically stable. Indeed, configuration data is frequently given in Spline (or NURBS) form, though this data generally has to undergo a post-processing before it can be used for a numerical simulation. For references on splines we refer to [34,61].

For $i = 0, 1, \ldots, N$ we denote with $N_{i,p,T}$ the ith normalized B-spline of order p with respect to the knot vector T. Here $T = (t_i)_{i=0}^{N+p-2}$ is a non-decreasing and non-stationary sequence of real numbers, i.e., $t_i \leqslant t_{i+1}$ and $t_i < t_{i+p}$. The B-splines are piecewise polynomials of degree $p-1$ and can be defined by the recursion

$$N_{i,1,T}(t) = \chi_{[t_i,t_{i+1})}(t) = \begin{cases} 1 & \text{if } t_i \leqslant t < t_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

$$N_{i,p,T}(t) = \frac{t - t_i}{t_{i+p-1} - t_i} N_{i,p-1}(t) + \frac{t_{i+p} - t}{t_{i+p} - t_{i+1}} N_{i+1,p-1}(t). \tag{19}$$

Usually we choose $p = 4$, i.e., cubic splines. From this we build surfaces (or planar grids) and volume-mappings by tensor products of the form

$$x(u,v) = \sum_{i=0}^{N} \sum_{j=0}^{M} p_{i,j}, N_{i,p,U}(u) N_{j,q,V}(v) \quad \text{resp.} \tag{20}$$

$$x(u,v,w) = \sum_{i=0}^{N} \sum_{j=0}^{M} \sum_{k=0}^{L} p_{i,j,k} \; N_{i,p,U}(u) \; N_{j,q,V}(v) \; N_{k,r,W}(w). \tag{21}$$

The p_{ij} resp. p_{ijk} are called *control points*. They are not grid points but can be considered as discrete approximation of the grid function, see Fig. 7. Due to the development of fast interpolation and

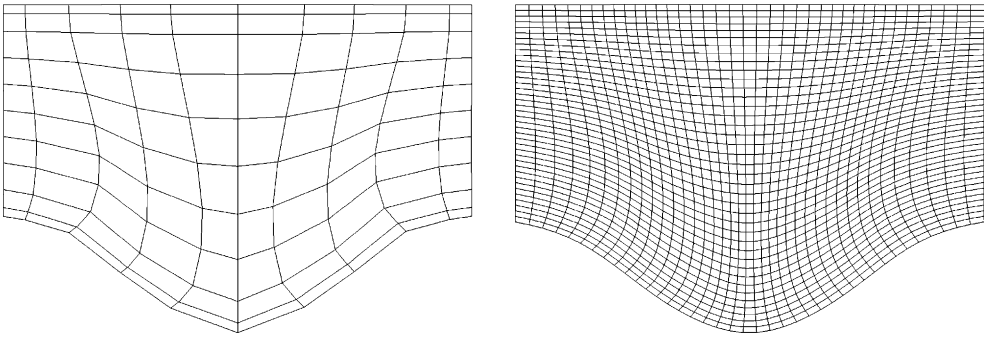


Fig. 7. Control points and evaluation of grid function.

approximation algorithms [20] discrete, logically Cartesian grids can efficiently be converted to B-spline form so that existing grid generation tools, for example elliptic or hyperbolic methods can easily be integrated into the current concept [21,22].

4.3. Anisotropic grids

For viscous flows it is useful to preadapt the grids to the boundary layers. Such a grid grading can easily be incorporated into an existing isotropic grid by applying a stretching function to the original grid, i.e., in 2D one considers

$$\tilde{\boldsymbol{x}}(u,v) = \boldsymbol{x}(\varphi(u), \varrho(v)), \tag{22}$$

as grid function where φ and ϱ are monotonous functions of the unit interval. A collection of appropriate stretching functions is discussed in [52], good results are generally achieved by the logarithmic function

$$\varphi^{\varepsilon}(\xi) = \frac{\ln(1+\xi\sqrt{\varepsilon})}{\ln(1+\sqrt{\varepsilon})}, \tag{23}$$

where the parameter ε controls the slope of the boundary at the point $\xi = 0$. Changing the stretching function does not require a remeshing. Since the flow solver must be able to handle hanging nodes anyway, there does not arise the necessity to preserve the continuity of grid lines over the internal block boundaries, see Fig. 8. This freedom generally simplifies the generation of valid block decompositions enormously. Additionally the fact that the stretching of the grid cells does not need to protrude into the far field, as is generally the case when conforming block boundaries are required, improves the convergence rates.

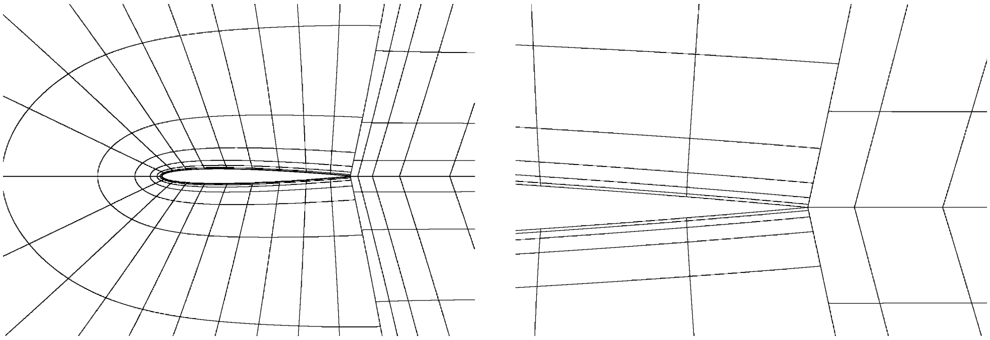


Fig. 8. Adjacent blocks with different parameterizations.

4.4. Moving grids

For the simulation of fluid–structure interaction the grid generator has to cope with time dependent domain boundaries. In this section we show how the B-spline representation can be used to accomplish this task efficiently.

Usually the deformation of the configuration surfaces is provided by an external structural solver based on finite elements. This data will be converted into B-spline form by means of approximation. Due to the high approximation order of B-splines only few control points are needed for an appropriate representation of the geometry, seldom more than 20 in each coordinate direction per block. Since the control points provide a discrete approximation to the grid function it is possible to apply well known algebraic methods in order to compute the new positions of the remaining interior control points. These methods only require minor modifications, which we will demonstrate for the following two examples.

Displacement TFI. The first technique is similar to that proposed in [38,65]. It is based on transfinite interpolation (TFI) as proposed by Gordon and Hall in [40]. With $\boldsymbol{p}^n_{i,j,k}$ and $\boldsymbol{p}^{n+1}_{i,j,k}$ we denote the position of the control points at time level t_n and the new timelevel t_{n+1} respectively. We assume that the deformed positions of all six faces (the four edges in 2D) of a block are given. In Fig. 9 we have just kept the far field fixed and connected the vertices at the trailing edge and the leading edge of the profile with the farfield edges by straight lines. First one computes the displacements of the control points at the block boundaries:

$$\Delta\boldsymbol{p}_{i,j,k} = \boldsymbol{p}^{n+1}_{i,j,k} - \boldsymbol{p}^{n}_{i,j,k}. \tag{24}$$

Then the displacements of the interior grid points are derived by the well known transfinite interpolation recursion

$$\Delta\boldsymbol{p}^1_{i,j,k} = (1 - u_{i,j,k})\Delta\boldsymbol{p}_{0,j,k} + u_{i,j,k}\Delta\boldsymbol{p}_{N,j,k}, \tag{25}$$

$$\Delta\boldsymbol{p}^2_{i,j,k} = \Delta\boldsymbol{p}^1_{i,j,k} + (1.0 - v_{i,j,k})(\Delta\boldsymbol{p}_{i,0,k} - \Delta\boldsymbol{p}^1_{i,0,k}) + v_{i,j,k}(\Delta\boldsymbol{p}_{i,M,k} - \Delta\boldsymbol{p}^1_{i,M,k}), \tag{26}$$

and, finally,

$$\Delta\boldsymbol{p}_{i,j,k} = \Delta\boldsymbol{p}^2_{i,j,k} + (1.0 - w_{i,j,k})(\Delta\boldsymbol{p}_{i,j,0} - \Delta\boldsymbol{p}^2_{i,j,0}) + w_{i,j,k}(\Delta\boldsymbol{p}_{i,j,L} - \Delta\boldsymbol{p}^2_{i,j,L}). \tag{27}$$

Generally it does not suffice to take $u_{i,j,k} = \frac{i}{N}$, $v_{i,j,k} = \frac{j}{M}$, $w_{i,j,k} = \frac{k}{M}$ (this would be the original linear blend from [40]), because this method may lead to grid folding when it is applied to anisotropic grids. Instead one should take the arclengths into account. Since the distances of the control points do generally not provide good approximation of the arclengths, we compute

$$u_{i,j,k} = \frac{\int_0^{\overline{U}_i} ||\boldsymbol{x}_u(u, \overline{V}_j, \overline{W}_k)||\mathrm{d}u}{\int_0^1 ||\boldsymbol{x}_u(u, \overline{V}_j, \overline{W}_k)||\mathrm{d}u}, \tag{28}$$

where $\overline{U}_i = \frac{1}{p-1}\sum_{n=i+1}^{i+p-1} u_n$ denotes the so-called *Greville abscissae* with respect the knot vector U, cf. [39]. The vectors v_{ijk}, w_{ijk} and the abscissae $\overline{V}_j$, $\overline{W}_k$ are defined analogously. The integrals are approximated by quadrature rules.

Angle preserving method. An alternative method is applicable in the case when only one boundary of a block has prescribed deformation (here we take the surface defined by the control points $\boldsymbol{p}_{i,j,0}$) and the other boundaries can move freely. For planar discrete grids this has been described in [57]. For each control point of the surface grid we first compute the translation $\Delta\boldsymbol{p}_{i,j,0}$ of the control point and the rotation of the normal vector at the corresponding Greville abscissae. Let α be the angle between the surface normals $\boldsymbol{n}^n$ and $\boldsymbol{n}^{n+1}$ at timelevels n and $n+1$. Then the rotation can be described by an orthogonal matrix Q

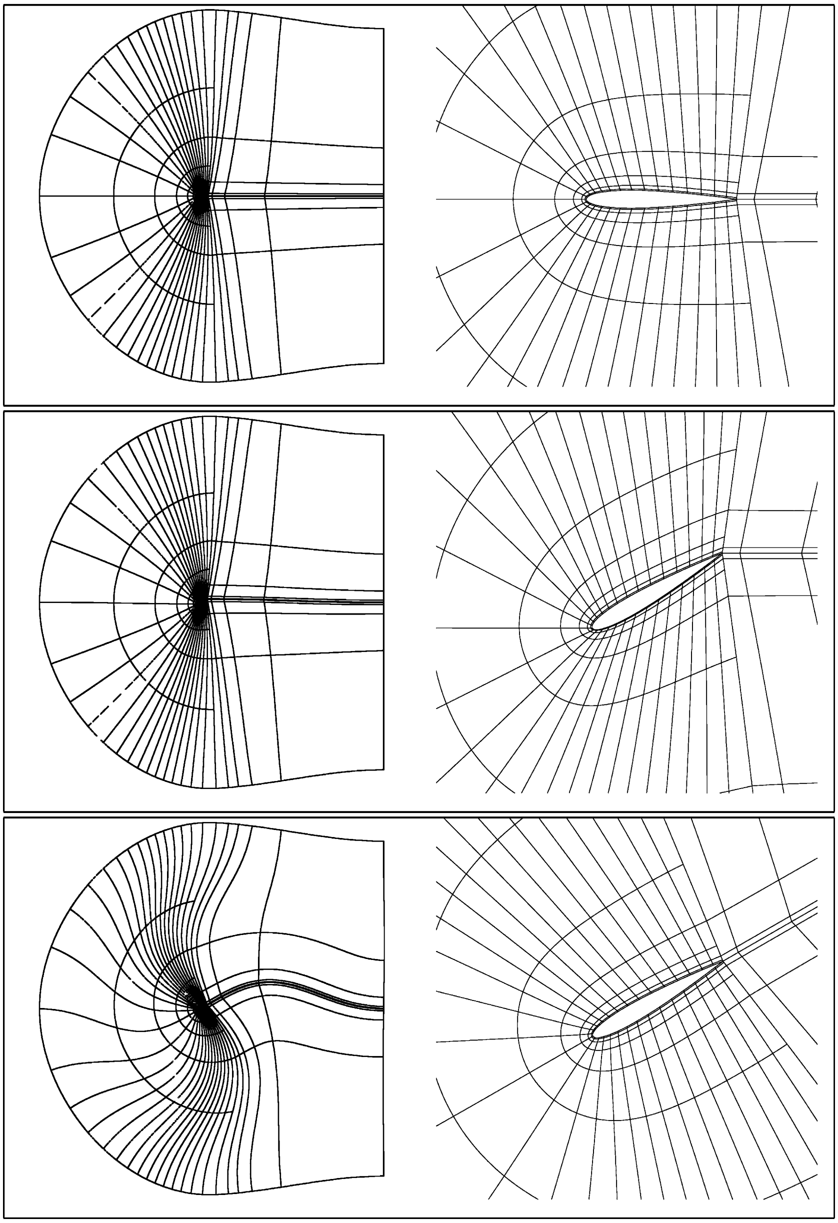


Fig. 9. Comparison of different grid deformation strategies. Top: original four-block grid for flow around NACA0012 airfoil; Center: profile rotated by 30° and grid adjusted with TFI-method; Bottom: grid adjusted using the angle preserving method.

$$Q = \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}} + \cos(\alpha)(I - \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}) + \sin(\alpha)\begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix}, \tag{29}$$

where $\boldsymbol{v} = \boldsymbol{n}^{n+1} \times \boldsymbol{n}^{n}$. α need not be computed but it suffices to evaluate $\sin(\alpha) = ||v||$ and $\cos(\alpha) = \boldsymbol{n}^{n+1} \cdot \boldsymbol{n}^{n}$. With this displacement and rotation we move the corresponding control grid line in a rigid way:

$$\boldsymbol{p}^{\mathrm{ref}}_{i,j,k} = \boldsymbol{p}_{i,j,0} + \Delta\boldsymbol{p}_{i,j,0} + Q(\boldsymbol{p}_{i,j,k} - \boldsymbol{p}_{i,j,0}). \tag{30}$$

The new grid control points are constructed by blending the reference grid line and the old grid line such that the points at the opposite surface remain fixed:

$$\boldsymbol{p}^{n+1}_{i,j,k} = (1.0 - b(w_{i,j,k}))\boldsymbol{p}^{\mathrm{ref}}_{i,j,k} + b(w_{i,j,k})\boldsymbol{p}^{\mathrm{old}}_{i,j,k}. \tag{31}$$

The blending function b has zero slope at the endpoints to ensure that the grid properties at the boundary, e.g., orthogonality and wall distances, are maintained and that the grid transition is smooth in the farfield. A possible choice is the quintic polynomial

$$b(\xi) = 1 - 10\xi^3 + 15\xi^4 - 6\xi^5. \tag{32}$$

proposed in [47]. In Fig. 9 this method has been applied to the four-block configuration around the NACA0012-profile. The wing is rotated by 30° around a rotation center that divides the chord by a ratio of 1:2. First of all the two blocks adjacent to the profile are deformed. From this follows a deformation of the vertical internal block boundaries. This deformation is the input for the deformation of the other two blocks.

5. Finite volume method

The occurrence of hanging nodes due to local mesh adaptation poses particular difficulties concerning the discretization of the governing equations. In the following section we present a finite volume method, which is capable to operate on meshes of any arbitrary topology. This approach offers a unified way to incorporate hanging nodes. Its main ingredients will be discussed in detail, including data structures, realization of spatial second order accuracy, treatment of convective and viscous fluxes as well as time integration.

5.1. Data structure

The discretization of the governing Eq. (1) is based on a cell centered finite volume scheme. The locally adapted grid is treated as a fully unstructured mesh, composed of simply connected elements with otherwise arbitrary topology. This flexibility is crucial for supporting the adaptive concept. Different element types are processed in a unified manner, rather than being distinguished. In particular, hanging nodes do not require any special treatment. The data structure of the flow solver is primarily based on the faces of the grid. A face based data structure has the advantage that there are no limitations on the number of faces, which can be connected to a cell, see Fig. 10 for an illustration in two space dimensions. The evaluation of fluxes and their contribution to cells can be efficiently implemented by sweeps over the faces.

5.2. Discretization of inviscid fluxes

The discretization of the inviscid fluxes is based on upwind methods. In the present study, the HLLC flux-difference splitting due to Batten et al. [12] is employed. For inviscid flows with strong discontinuities, e.g., hypersonic flows, we give preference to the more robust flux-vector splitting proposed by Hänel and Schwane [44].

The higher-order extension of the scheme is crucial to obtain accurate solutions of the governing equations. We choose a polynomial ansatz function for the reconstruction operator $R^k_{V_i}$, which is a trun-

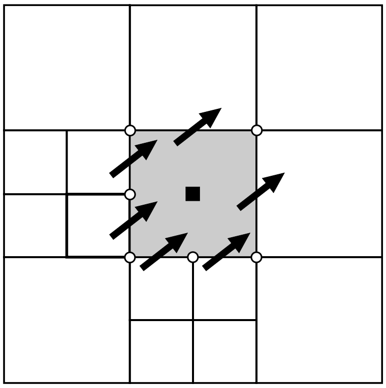


Fig. 10. Collection of fluxes for polygonally bounded control volume in two space dimensions.

cated Taylor series expansion around the centroid $\mathbf{x}_i$ of V_i. The reconstruction is said to be k-exact [9], when $R^k_{V_i}$ exactly reconstructs any polynomial $P(\mathbf{x})$ of degree k or less. To obtain second-order accuracy in space, a linear reconstruction ($k = 1$) of the primitive flow variables $w \in \{\varrho, \mathbf{v}, p\}$ is determined:

$$w(\mathbf{x})|_{V_i} := w_i + \varphi_i(\mathbf{x} - \mathbf{x}_i)^{\mathrm{T}} \cdot \nabla w_i, \quad \mathbf{x} \in V_i, \tag{33}$$

where w_i represents the solution at the centroid $\mathbf{x}_i$ of V_i and φ_i denotes a limiter function. The local support N_i of the reconstruction is based on a fixed set of cells, that share a face with V_i (face neighbors). For unstructured meshes, the least-squares technique [8] and the Green–Gauss method [10] are commonly employed techniques to approximate the gradient ∇w_i of the quantity in question. The principle of the least-squares reconstruction is to minimize the error in reconstructing the integral cell averages $w_j, \forall j \in N_i$ of the neighboring cells, that locally support the higher order method:

$$\min \sum_{j \in N_i} ||A_j w(\mathbf{x}) - w_j||_2, \tag{34}$$

where A_j represents the cell-averaging operator

$$A_j w(\mathbf{x}) := \frac{1}{|V_j|} \int_{V_j} w(\mathbf{x}) \, \mathrm{d}V. \tag{35}$$

For the linear reconstruction within a cell V_i the system of equations can be written as

$$\mathbf{B} \nabla w_i = \Delta w_i, \tag{36}$$

where

$$\mathbf{B} = \begin{bmatrix} \Delta x_1 & \Delta y_1 & \Delta z_1 \\ \vdots & \vdots & \vdots \\ \Delta x_N & \Delta y_N & \Delta z_N \end{bmatrix}, \quad \Delta w_i = \begin{pmatrix} \Delta w_1 \\ \vdots \\ \Delta w_N \end{pmatrix}, \tag{37}$$

with $\Delta(\bullet)_j = (\bullet)_j - (\bullet)_i, \forall j \in N_i$. The usually over-determined system of Eq. (36) is solved in a least squares sense by using normal equations. Numerical experiments have shown, that the use of normal equations is sufficient for the considered reconstruction technique.

Alternatively to the least squares method, the gradient ∇w_i can be determined using the divergence (Green–Gauss) theorem

$$\nabla w_i = \frac{1}{|\Gamma_i|} \oint_{\partial \Gamma_i} w \, \mathbf{n} \, \mathrm{d}S, \tag{38}$$

where Γ_i denotes an auxiliary control volume, which is defined by the centroids of the neighboring cells of V_i, which support the reconstruction. The Green–Gauss technique is currently applied in two space dimensions only.

At local extrema and discontinuities, the reconstruction polynomial may generate new extrema and therefore cause oscillations in the numerical solution. In order to circumvent this problem, the slope limiter by Venkatakrishnan [68] is employed.

5.3. Discretization of diffusive fluxes

For the discretization of the diffusive fluxes, the gradients of the velocity vector, ∇v_i, and the temperature, ∇T, are required at the cell interfaces. The simplest procedure is to compute the gradients of the quantity in question ∇w within each cell and to average ∇w between the two cells that share a face on its left hand side (∇w_{L}) and on its right hand side (∇w_{R}), respectively,

$$\nabla w|_{\text{face}} := \frac{1}{2}(\nabla w_{\mathrm{L}} + \nabla w_{\mathrm{R}}). \tag{39}$$

The gradients ∇w_{L} and ∇w_{R} are provided by the (unlimited) reconstruction procedure. This kind of discretization supports undamped oscillatory modes, that result from an odd–even point decoupling. A tighter coupling of the solution is obtained by approximating the gradient in the direction $\mathbf{l}_{\mathrm{LR}} = \mathbf{x}_{\mathrm{R}} - \mathbf{x}_{\mathrm{L}}$, that connects the centroids of the left and right cell of the face, by the divided difference

$$\left.\frac{\partial w}{\partial \mathbf{l}_{\mathrm{LR}}}\right|_{\text{face}} = \frac{w_{\mathrm{R}} - w_{\mathrm{L}}}{|\mathbf{l}_{\mathrm{LR}}|}. \tag{40}$$

Finally, the gradient is expressed by combining Eqs. (39) and (40)

$$\nabla w|_{\text{face}} = \overline{\nabla w}|_{\text{face}} - \left(\overline{\nabla w}|_{\text{face}} \cdot \frac{\mathbf{l}_{\mathrm{LR}}}{|\mathbf{l}_{\mathrm{LR}}|} - \frac{w_{\mathrm{R}} - w_{\mathrm{L}}}{|\mathbf{l}_{\mathrm{LR}}|} \right) \frac{\mathbf{l}_{\mathrm{LR}}}{|\mathbf{l}_{\mathrm{LR}}|}, \tag{41}$$

where $\overline{\nabla w}|_{\text{face}}$ is the averaged gradient, according to (39). In [35], Deister provides a detailed analysis for this kind of gradient estimation, with respect to its properties regarding accuracy and positivity on locally adapted grids.

5.4. Time integration

After having completed the spatial discretization, we obtain a system of ordinary differential equations

$$\frac{\partial}{\partial t} \int_{V(t)} \mathbf{u} \, \mathrm{d}V + \mathbf{R}(\mathbf{u}) = \mathbf{0}, \tag{42}$$

where $\mathbf{R}(\mathbf{u})$ denotes the residual vector defined by the sum of the discretized fluxes.

The implicit time integration scheme is suitable for steady as well as unsteady flow simulations. The governing equations are expressed in semi-discrete form as follows:

$$\widehat{\mathbf{R}} := \frac{\mathbf{u}^{n+1}|V^{n+1}| - \mathbf{u}^{n}|V^{n}|}{\Delta t} + \mathbf{R}(\mathbf{u}^{*}, \mathbf{x}^{*}) = \mathbf{0}. \tag{43}$$

The vector of state $(\bullet)^{*}$ is defined as $(\bullet)^{*} = \theta \cdot (\bullet)^{n+1} + (1-\theta) \cdot (\bullet)^{n}$. For steady fluid flow, a first order accurate implicit Euler time integration method is chosen, with $\theta = 1$. For unsteady flows, a second order time accurate implicit midpoint rule is employed, with $\theta = 0.5$. The solution $\mathbf{u}^{n+1}$ of the non-linear system (43) is determined by a Newton iteration within each physical time step:

$$\widehat{\mathbf{J}}(\mathbf{u}^{(l)})\Delta\mathbf{u}^{(l)} = -\frac{\mathbf{u}^{(l)}|V^{n+1}| - \mathbf{u}^{(n)}|V^{n}|}{\Delta t} - \mathbf{R}(\tilde{\mathbf{u}}, \mathbf{x}^{*}), \tag{44}$$

with

$$\lim_{l\to\infty} \mathbf{u}^{(l)} = \mathbf{u}^{n+1}. \tag{45}$$

The vector of state $\tilde{\mathbf{u}}$ is defined as $\tilde{\mathbf{u}} = \theta\mathbf{u}^{(l)} + (1-\theta)\mathbf{u}^{n}$. $\Delta u^{(l)} := \mathbf{u}^{(l+1)} - \mathbf{u}^{(l)}$ denotes the change of the solution within each Newton step, indicated by the superscript (l). The initial guess is $\mathbf{u}^{(0)} = \mathbf{u}^{n}$. For stationary flows we take one Newton iteration per physical time step. The Jacobian of the system of equations contains contributions of the temporal discretization and of the spatial discretization:

$$\widehat{\mathbf{J}}(\mathbf{u}^{(l)}) = \frac{\partial\widehat{\mathbf{R}}(\mathbf{u}^{(l)})}{\partial\mathbf{u}^{(l)}} = \frac{|V^{n+1}|}{\Delta t}\,\mathbf{I} + \theta\,\frac{\partial\mathbf{R}(\tilde{\mathbf{u}})}{\partial\tilde{\mathbf{u}}}. \tag{46}$$

The linearization of the convective fluxes is based on a first order accurate method in space. The convective flux functions are linearized by using tools of automatic differentiation, namely by the ADIFOR software [18]. The exact linearization of the diffusive fluxes requires a larger stencil than the linearization of the convective fluxes, based on a first order accurate method in space. Consequently, a higher memory requirement would be necessary. To maintain the matrix graph of the convective flux Jacobians, we introduce a simplified approximation of the face-gradients for purpose of linearization. The projection of the gradient of interest, ∇w, in the direction $\mathbf{l}_{\mathrm{LR}} = \mathbf{x}_{\mathrm{R}} - \mathbf{x}_{\mathrm{L}}$, that connects the centroids of the left and right cell of the face, is approximated by the finite difference

$$\nabla w|_{\mathrm{face}} \cdot \frac{\mathbf{l}_{\mathrm{LR}}}{|\mathbf{l}_{\mathrm{LR}}|} \approx \frac{w_{\mathrm{R}} - w_{\mathrm{L}}}{|\mathbf{l}_{\mathrm{LR}}|}. \tag{47}$$

The gradient components perpendicular to $\mathbf{l}_{\mathrm{LR}}$ are neglected. The transport coefficients μ and κ are frozen for the process of linearization.

The linear system of Eq. (44) is solved by an iterative Krylov subspace method. In the present study, we employ the GMRES algorithm [64], pre-conditioned by an incomplete LU-factorization. The implementation of the Newton–Krylov method is based on the PETSc [6] library of Argonne National Laboratory.

6. Numerical results

In the following, various test cases are considered to demonstrate the benefits of the adaptive concept for a wide range of applications. First, we investigate the inviscid, stationary flow about the NACA0012 airfoil in transonic Mach number regime, the inviscid hypersonic flow over a double ellipse and the transonic flow over a swept wing in a channel. To evaluate the method for viscous flows, the laminar flow about the NACA0012 airfoil and the laminar flow over a flat plate are investigated. Finally, we conclude with the unsteady, inviscid flow over the NACA0012 airfoil undergoing forced oscillation in pitch.

6.1. Transonic NACA0012 airfoil

We consider the inviscid, transonic flow about the NACA0012 airfoil at $M_\infty = 0.95$, $\alpha = 0°$ (AGARD reference test case 03 [1]). The far-field boundary is located about 20 chord lengths away from the airfoil. Standard characteristic boundary conditions are applied at the far-field, where the incoming Riemann invariant is set by free stream quantities. Computations are initialized on a structured grid consisting of four blocks with a resolution of 20 × 20 cells each. Thirteen cycles of adaptation have been performed with a maximum refinement level of $L_{\max} = 8$. Adaptation has been carried, each time the density residual decreased four orders of magnitude. The threshold value for the multiscale analysis is $\varepsilon = 4 \times 10^{-2}$. The flow pattern downstream of the trailing edge is characterized by a complex shock configuration. Two oblique shocks are formed at the trailing edge. The remaining supersonic region behind the oblique shocks is closed by a further normal shock. This configuration is often related to as so-called *fish-tail*. Fig. 11 presents the locally adapted grid and the corresponding Mach distribution in the vicinity of the airfoil, after 13 cycles of adaptation. The grid consists of 55,084 cells. All three shocks are highly resolved by the adaptive grid. The position of the normal shock is located at $x \approx 2.1721$ chord length behind the trailing edge of the airfoil.

Fig. 12 presents a total view of the shock configuration. The oblique shocks extend about 10–12 chord lengths into the flow domain. The adaptive grid provides high resolution over the complete extent of the shocks. Such a high shock resolution is not feasible using standard structured grids. Discretization of the shock region between $x \in [1, 5]$, $y \in [-10, 10]$, by a uniform structured mesh according to a refinement level $L = 8$ equals about 29.5×10^6 grid cells. A uniform discretization of the complete flow domain according to $L = 8$ would result in about 10^8 cells.

6.2. Hypersonic flow over a double-ellipse

Next, the inviscid, two-dimensional hypersonic flow about a double-ellipse at $M_\infty = 8.15$, $\alpha = 30°$ is considered. It represents a standard test case for flow simulations of reentry vehicles [37]. For the presented

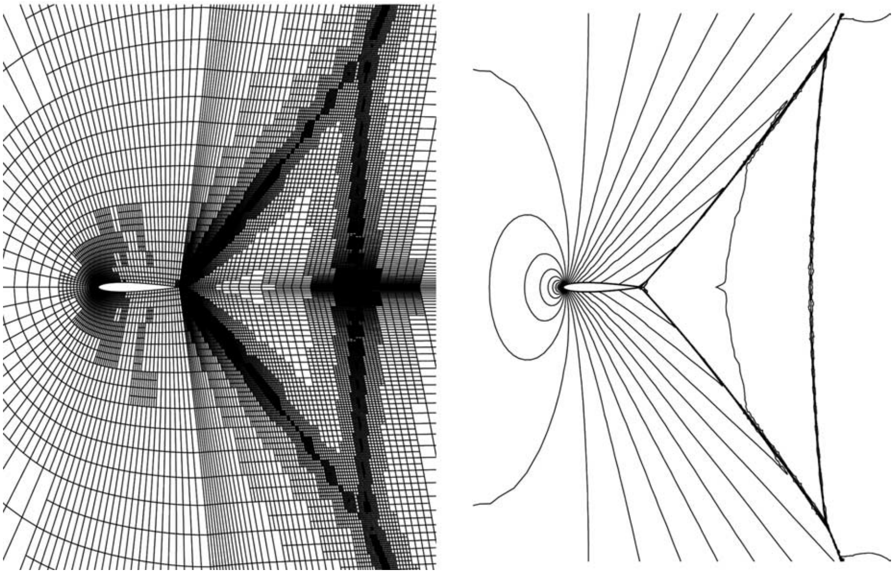


Fig. 11. Partial view of NACA0012 airfoil, $M = 0.95$, $\alpha = 0.0°$. Left figure: Computational grid. Right figure: Mach distribution, $M_{\min} = 0.0$, $M_{\max} = 1.45$, $\Delta M = 0.05$.

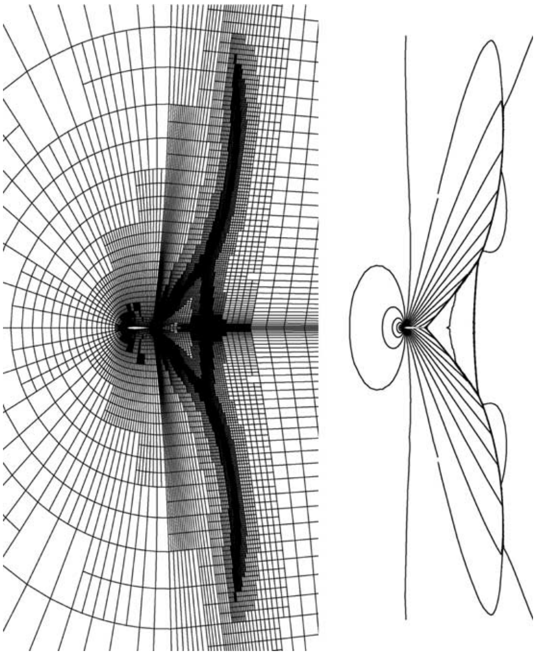


Fig. 12. Total view of NACA0012 airfoil, $M = 0.95$, $\alpha = 0.0°$. Left figure: Computational grid. Right figure: Mach distribution, $M_{\min} = 0.0$, $M_{\max} = 1.45$, $\Delta M = 0.05$.

computations, the Green–Gauss reconstruction and the Hänel/Schwane flux vector splitting [44] are employed. The solution is implicitly advanced in time. Five cycles of adaptation are performed with a maximum refinement level of $L_{\max} = 5$. The mesh is adapted each time the density residual decreased five orders of magnitude. The threshold value of the adaptation is $\varepsilon = 2.5 \times 10^{-2}$. Fig. 13 presents a partial view of the computational grid and the corresponding distribution of the Mach number for three different stages of adaptation, namely for the initial grid, after three and five adaptations, respectively. On the initial mesh, the detached bow shock is extremely smeared. After five adaptations, the detached bow shock and the canopy shock are both sharply resolved. Fig. 14 shows a good agreement of the surface pressure distribution after five adaptations and the results obtained by Gustafsson et al. [37].

6.3. Swept wing in channel

The inviscid, three-dimensional flow over a swept, non-tapered wing in an open channel is investigated. The aerodynamic profile of the wing is based on the BAC-3-11/RES/30/21 airfoil [56], which has been selected for the transonic cruise configuration of the collaborative research center SFB401: Modulation of Flow and Fluid/Structure Interaction at Airplane Wings, RWTH Aachen, Germany [7]. In the computations presented here, the span b equals the chord length c. The flow conditions are $M_\infty = 0.77, \alpha = 0°$, the sweep angle is $\gamma = 34.0°$.

Fig. 15 illustrates the present configuration. The domain is bounded by two end-plates, mounted to the wing, while the remaining domain is open, extending about 20 chord lengths away from the wing in the x, y-plane. The initial grid consists of four blocks, each with a resolution of 10 cells in the according curvilinear coordinate directions. Five cycles of adaptation are conducted. The threshold value of the adaptation is $\varepsilon = 10^{-2}$ and the maximum refinement level is $L_{\max} = 6$. Fig. 16 presents a perspective view of the Mach

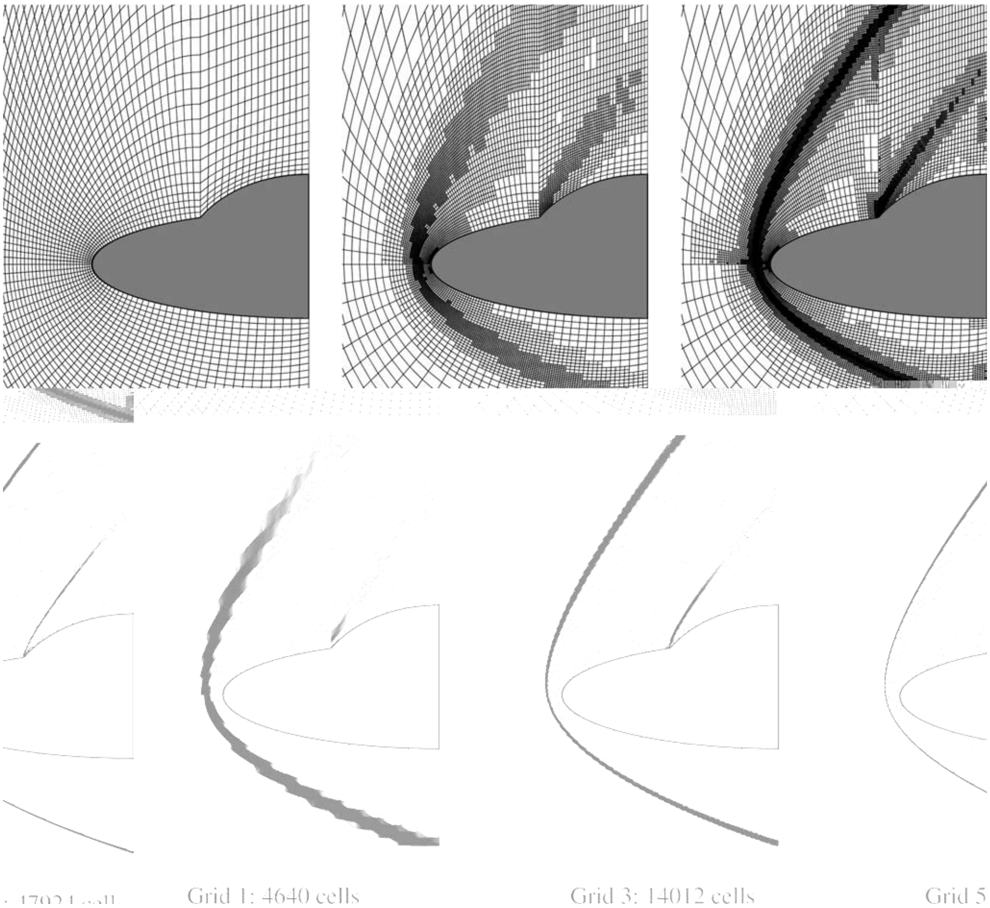


Fig. 13. Inviscid flow about double-ellipse, $M_\infty = 8.15$, $\alpha = 30°$. Upper figures: Computational grids (partial view); Lower figures: Mach number distribution, $M_{\min} = 0$, $M_{\max} = 8.15$, $\Delta M = 0.15$.

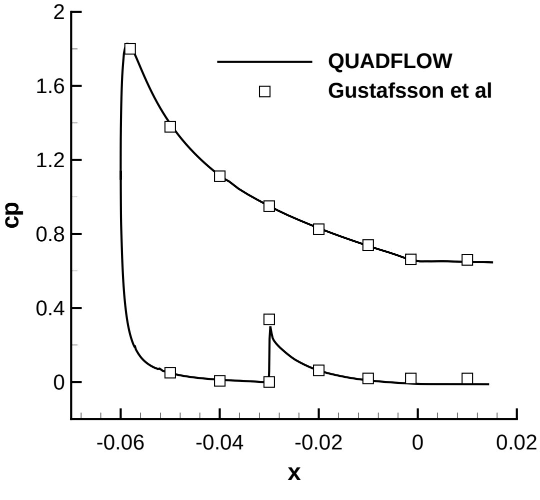


Fig. 14. Surface pressure distribution for inviscid flow about double-ellipse, $M_\infty = 8.15$, $\alpha = 30°$.

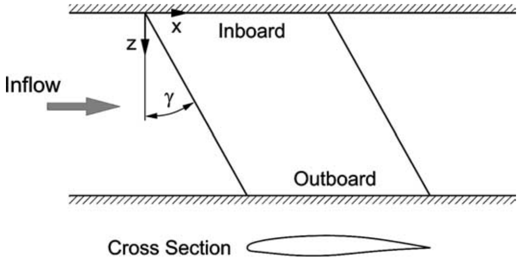


Fig. 15. Geometry of swept wing in channel. Cross-section: BAC-3-11/RES/30/21 airfoil.

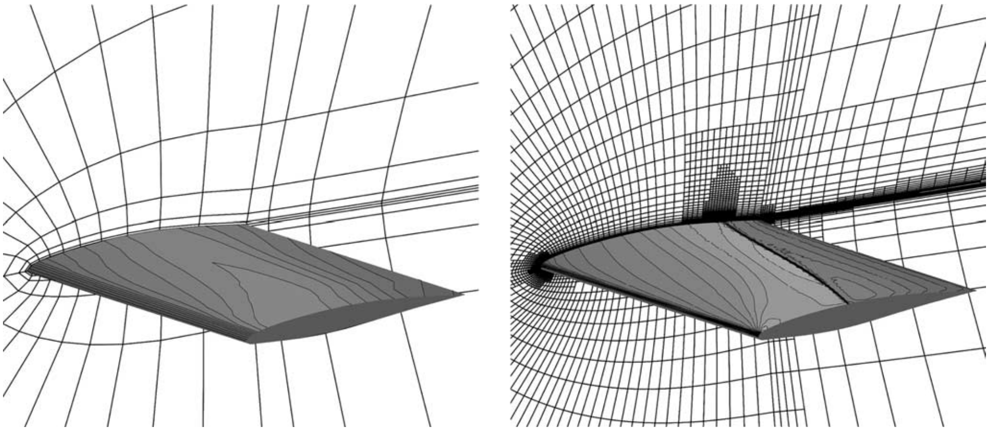


Fig. 16. Swept wing in channel ($M = 0.77$, $\alpha = 0°$, $\gamma = 34.0°$). Mach number distribution on wing surface and computational grid at inboard wall of channel. Left figure: Solution for initial grid. Right figure: Solution after five adaptations.

number distribution on the upper surface of the wing and the computational grid at the inboard wall of the channel for the initial configuration and after five adaptations. The adapted mesh consists of 804,877 cells. The pressure distributions on the upper surface of the wing for two cross-sections, namely at $z/b = 0.2$ and 0.8 are depicted in Fig. 17. On the initial grid, the solution is extremely smeared, so that no shock can be identified. The adaptive scheme detects the shock and provides automatically a high resolution of the shock and of the stagnation areas at the leading edge and at the trailing edge. In spanwise direction, the position of the shock is shifted towards the leading edge and the after-expansion is more pronounced.

6.4. Laminar flow over NACA0012 airfoil

We consider the laminar flow over the NACA0012 airfoil at $M_\infty = 0.8$, $\alpha = 10°$ and a Reynolds number of $Re_\infty = 500$. This testcase was specified as part of a GAMM workshop [24] to assess the accuracy of Navier–Stokes solvers. The flow field is characterized by a large separation region on the upper surface of the airfoil.

The initial grid consists of 1600 cells. The far-field boundaries are located approximately 20 chord lengths away from the airfoil. The first grid spacing off the wall is about 1.5×10^{-3}. Five cycles of adaptation are performed with a maximum refinement level of $L_{\max} = 6$. The threshold value of the adaptation is $\varepsilon = 1.0 \times 10^{-2}$. Fig. 18(a) and 18(b) show the initial grid and the locally adapted mesh after five cycles of

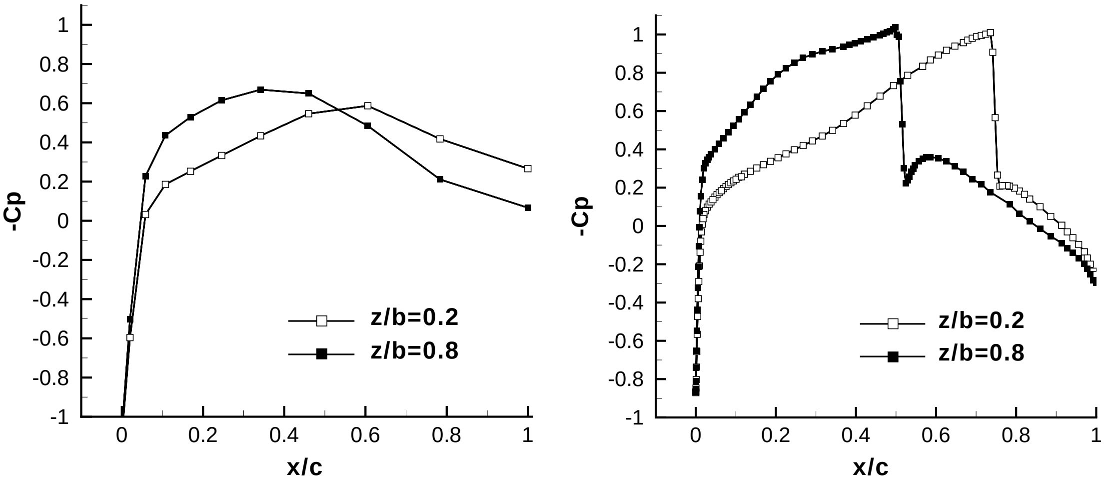


Fig. 17. Swept wing in channel ($M_\infty = 0.77, \alpha = 0°, \gamma = 34.0°$): Pressure distribution on upper surface for two cross-sections, $z/b = 0.2$ and 0.8. Left figure: Solution for initial grid. Right figure: Solution after five adaptations.

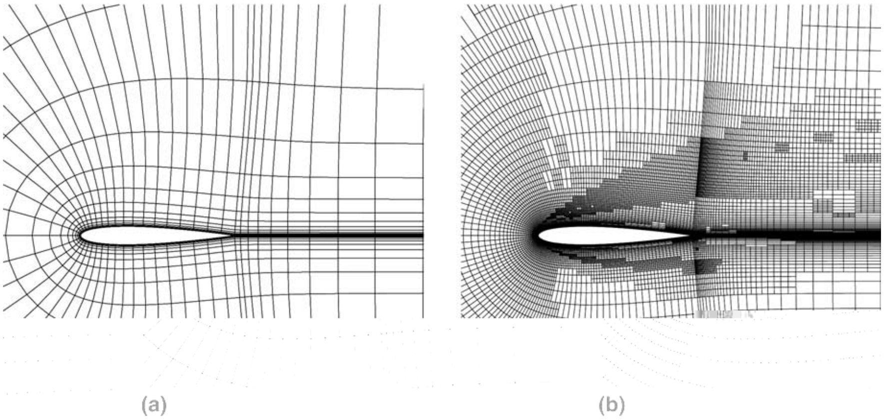


Fig. 18. Laminar flow about the NACA0012 airfoil, $M_\infty = 0.8$, $\alpha = 10°$, $Re_\infty = 500$. (a) Initial grid, resolution: 1600 cells; (b) Grid after five adaptations, resolution: 24,658 cells.

adaptation. The wake is automatically detected by the adaptation criteria and the resolution of the grid is adjusted accordingly. The high resolution of the complete wake, extending to the far-field boundary, leads to a relative large number of 24,658 grid cells. Fig. 19(a) and 19(b) present the Mach number distribution and the streamlines in vicinity of the airfoil. The recirculation region and the vortex system are clearly to identify. Table 1 summarizes the aerodynamic coefficients and the location of the separation point x_{sep} on the upper surface of the airfoil. C_{L} denotes the lift coefficient due to the pressure force, C_{D}^{p} the drag due to pressure, C_{D}^{f} the drag due to friction and $C_{\text{D}}^{\text{tot}}$ represents the total drag, that consists of the sum of the drag due to pressure and friction. The current results obtained after five stages of adaptation (grid 6), as well as results obtained by Delanaye [36] and Cambier [24], which serve as a reference, are shown. The location of the separation point agrees well in all cases. The aerodynamic coefficients predicted by the current scheme (on grid 6) are slightly larger than the ones determined by Delanaye and Cambier.

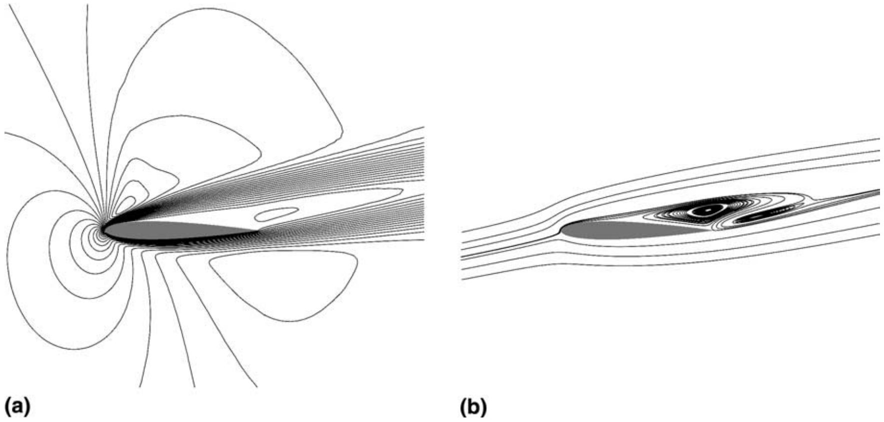


Fig. 19. Laminar flow around the NACA0012 airfoil after five adaptations, $M_\infty = 0.8$, $\alpha = 10°$, $Re_\infty = 500$. (a) Mach number distribution, $M_{\min} = 0.0$, $M_{\max} = 1.1$, $\Delta M = 0.1$. (b) Streamlines.

Table 1
Comparison of aerodynamic coefficients and the location of the separation point for NACA0012 profile, $M_\infty = 0.8$, $\alpha = 10°$, $Re_\infty = 500$

	Number of cells	x_{sep}	C_{L}	C_{D}^{p}	C_{D}^{f}	$C_{\text{D}}^{\text{tot}}$
QUADFLOW (Grid 6)	24,658	0.358	0.4458	0.1458	0.1267	0.2726
Delanaye [36]	13,114	0.358	0.4383	0.1452	0.1211	0.2663
Cambier [24]	17,225	0.359	0.4342	–	–	0.2656

6.5. Laminar boundary layer

In the scope of this section, we apply anisotropic adaptation to resolve the boundary layer along a flat plate. The flow conditions are determined by the Mach number $M_\infty = 0.2$ and the Reynolds number $Re_\infty = 10^4$, based on unit length. The wall is considered as isothermal, with $T_{\text{wall}} = T_\infty = 273.0$ K. For purpose of validation, the similarity solution according to Blasius [53] for an incompressible laminar fluid flow serves as a reference.

The plate extends along the x-axis between $x = 0.0$ and 6.0, with 100 cells located on the plate itself, see Fig. 20(a) for a partial view of the initial grid. Upstream of the leading edge, the lower boundary of the domain is modeled as an inviscid impermeable wall. The grid is clustered about the leading edge in streamwise direction, measuring a first grid spacing of 10^{-3}. In the y-direction, the initial resolution comprises only 10 grid points. In the following, nine cycles of adaptation in the y-coordinate direction are performed, with a highest refinement level permitted of $L_{\max} = 8$. The threshold value for the multiscale analysis is $\varepsilon = 10^{-2}$. The locally adapted grid after nine cycles of adaptation is depicted in Fig. 20(b). The highest refinement level reached during the computation is $L = 7$, which is located in the vicinity of the leading edge of the plate. This fact indicates that even in the presence of steep gradients within the boundary layer, the multiscale analysis converges within the range of the prescribed threshold value i.e., the adaptation procedure was not truncated by reaching the maximum permissible refinement level. Fig. 21 shows a good agreement between the computed skin friction coefficient along the plate after nine adaptations and the theoretical solution, according to Blasius.

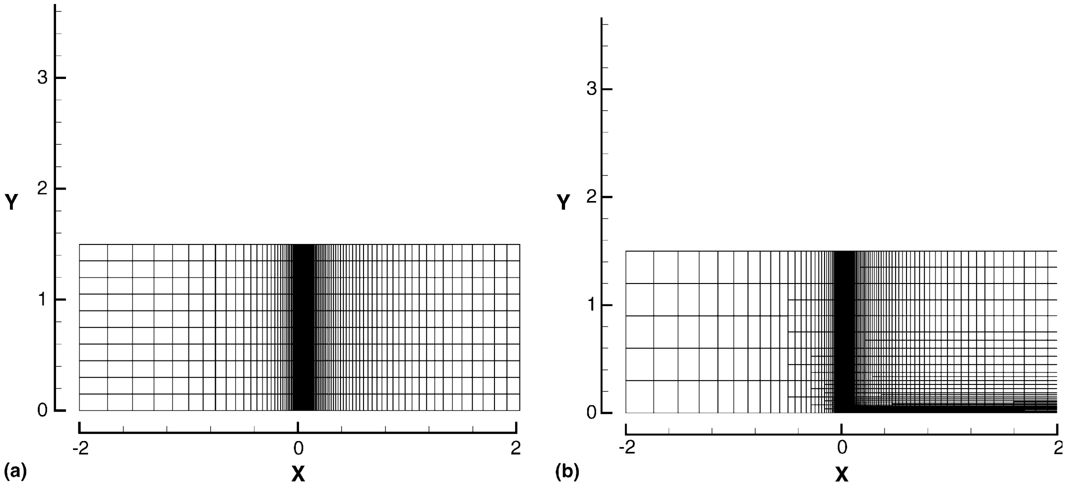


Fig. 20. Computational grids for laminar flow over flat plate, $M_\infty = 0.2$, $Re_\infty = 10^4$.

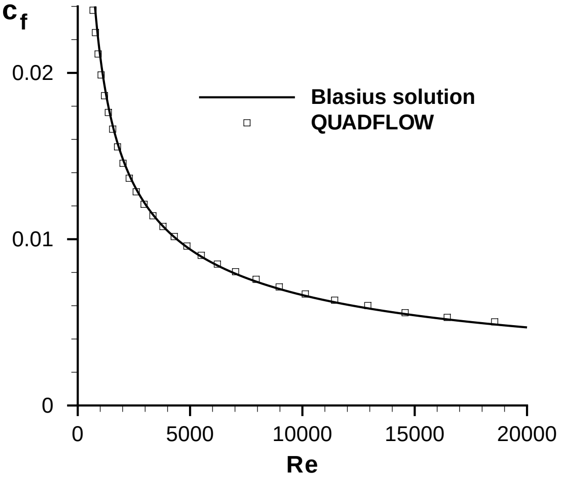


Fig. 21. Distribution of the skin friction coefficient for laminar flow over flat plate after nine adaptations, $M_\infty = 0.2$, $Re_\infty = 10^4$.

6.6. Oscillating NACA0012 airfoil

As a last example we show the inviscid, unsteady transonic flow at $M_\infty = 0.755$ about the NACA0012 airfoil undergoing forced oscillation in pitch

$$\alpha = 0.016^\circ + 2.51^\circ \sin(\omega t + \varphi), \tag{48}$$

about the quarter-chord. The reduced frequency is $k = \omega c/|\mathbf{v}_\infty| = 0.1628$. This test case was experimentally investigated by Landon [51]. For the numerical simulation, a phase shift of $\varphi = -90^\circ$ is introduced, so that for $t = 0$ the motion of the airfoil is accelerated but has no rotational velocity. We start the computation with a steady state solution about the initial airfoil position. For time integration the implicit mid-point rule

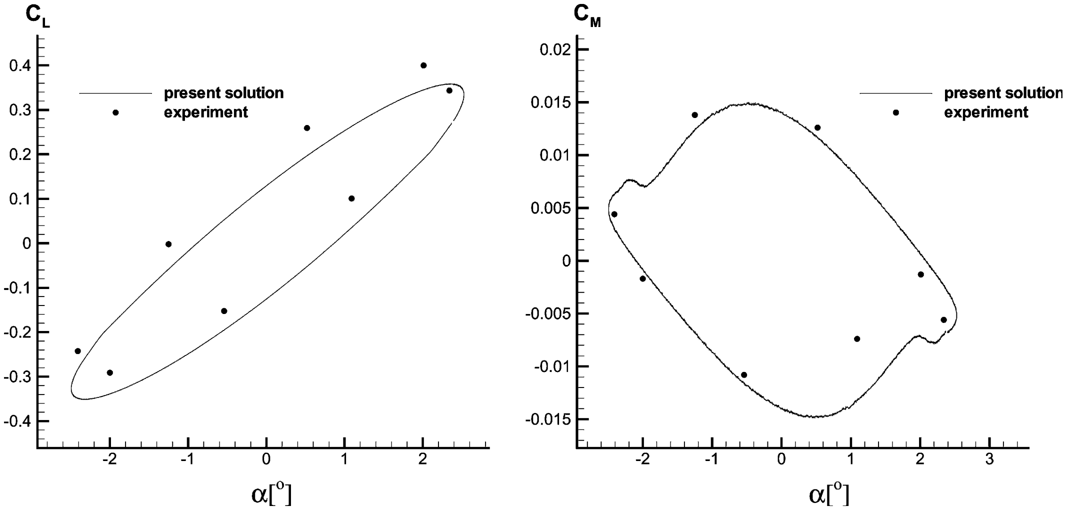


Fig. 22. Hysteresis curves for oscillating NACA0012 airfoil. Left: lift coefficient, right: moment coefficient.

is used. The physical timestep is chosen to correspond to a maximum CFL-number within the computational domain of about CFL = 50. As a result each oscillation cycle is resolved by 5100 time steps. The unsteady residual of the Newton iteration is reduced four orders of magnitude within each time step. Grid adaption is performed after each timestep using a threshold value of $\varepsilon = 10^{-3}$. Maximum refinement level is $L = 6$ resulting in about 15,000–18,000 grid cells depending on the phase of the oscillation. For the grid deformation the angle preserving method described in Section 4.4 is employed. Fig. 23 shows the isobars of the flow field and the adaptive grid for three different stages of the motion. The flow undergoes strong variations in time. During the upward motion of the airfoil ($\dot{\alpha} > 0$), a supersonic region is formed on the upper surface, which is closed by a shock. During the downward motion, the supersonic region on the upper surface breaks down and a supersonic region on the lower surface develops. Fig. 22 presents the hysteresis curves of the lift coefficient C_{L} and the moment coefficient C_{M} for the fourth cycle of oscillation. The discrepancies with respect to the experiments of Landon [51] are often observed in the literature [3,11,49,54].

6.7. Remarks on CPU times

In the previous examples we have decided to use mesh size estimates as measure for the efficiency of the adaptive method exclusively. This is just due to the fact that in case of steady state computations the computational overhead caused by the adaptation can be neglected. After each adaptation the flow solver is called to compute a nearly converged solution on the new grid. This typically requires several hundred timesteps and more than 98% of the overall computational time. Matters are different in case of instationary problems. Here the grid has to be adapted after each time step. In Table 2 we compare the execution times spent in the main modules of the solver for the pitching airfoil testcase, that has been described in the previous section.

The multiscale transform itself can be realized efficiently using an appropriate memory management, that in particular supports the inserting and removing of cells. For this reason specialized hashing strategies have been developed in the C++ template library `igpm_t_lib`, see [60]. A significant contribution to the execution time of this testcase stems from the exact evaluation of curvilinear cell volumes and cell centers that is more expensive than the formulas for hexahedrons usually employed in standard

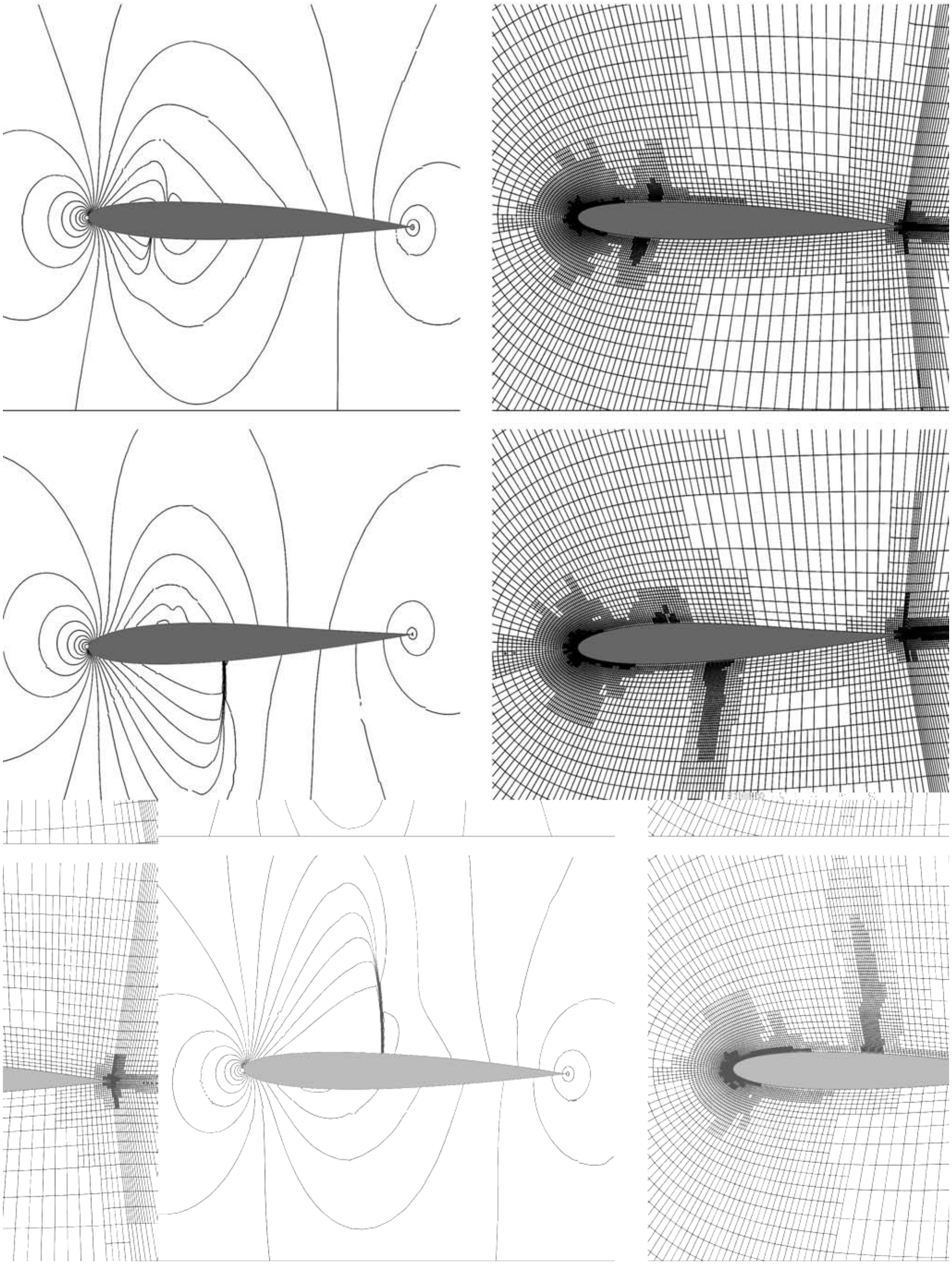


Fig. 23. Time evolution of pressure distribution and adaptive grid for flow around oscillating NACA0012 profile. Upper figures: $\alpha(t_1) = 0.82^\circ, \dot{\alpha}(t_1) < 0$; Center: $\alpha(t_2) = -2.24^\circ, \dot{\alpha}(t_2) < 0$; Lower figures: $\alpha(t_3) = 1.82^\circ, \dot{\alpha}(t_3) > 0$. $p_{\min} = 56,600$ Pa; $p_{\max} = 146,600$ Pa; $\Delta p = 4500$ Pa.

finite volume solvers. These quantities have to be recomputed before every timestep. In case of non-moving grids this contribution is less important, since only the data of newly inserted cells have to be computed.

Table 2
CPU-times for one cycle of oscillation, i.e. 5100 timesteps, on a Linux-PC with 2.4 GHz Intel Xeon processor

Module	Time (s)	Time (%)
Flow solver	37,791	69.2
Multiscale analysis	4982	9.1
Grid deformation	2341	4.2
Evaluation of geometry	9426	17.3

7. Conclusion and outlook

We have presented the still intermediate state of development of the new flow solver QUADFLOW that integrates dynamic adaptation, mesh generation and finite volume discretization. Its main features have been illustrated by numerous steady and unsteady computations governing a wide class of flow problems including transonic and hypersonic flows, laminar flows as well as moving boundaries. Without a-priori knowledge of the solution all physical relevant effects (shocks, boundary layers, etc.) are detected and appropriately resolved.

The present state of the implementation still lacks important features. We conclude with some remarks on further developments which are currently being performed within the Collaborative Research Center SFB 401 Flow Modulation and Fluid–Structure Interaction at Airplane Wings, RWTH Aachen.

Parallelization. So far the solver is not yet fully parallelized. This is a severe shortcoming, in particular, for 3D computations. The time-dependency of the problems considered here makes the parallelization inherently difficult, since local adaption may result in suboptimal load balancing, even if the initial domain partitioning has been chosen appropriately. Currently, our approach is to adapt the geometric block-structure of the grid dynamically to serve the needs of the parallelization.

Time adaptivity. Grid adaptation is currently performed only in space but not in time. However, we have been able to incorporate a local time stepping in the spirit of adaptive mesh refinement (AMR), see [13,14], into the adaptive multiscale concept. Recent results make us believe that we can significantly improve the efficiency of QUADFLOW in case of non-stationary problems.

Hybrid grid refinement. In the current version we can perform in each grid block either *isotropic* or *anisotropic* grid refinement. So far the refinement strategy has to be specified blockwise a-priori by the user. However there are applications such as an interaction between a shock and a boundary layer where in one grid block both refinement strategies are locally needed. For this purpose we are currently developing a grid adaptation tool based on a multiscale analysis that admits for isotropic *and* anisotropic grid refinement.

Physical Model. The ultimate goal of the project is the investigation of fluid–structure interactions at airplane wings. Up to now, we only investigated moving boundaries with a *prescribed* deformation. In order to account for aeroelastic effects we will incorporate a FEM code and couple it with QUADFLOW. Several coupling strategies have already been investigated, see [55].

Appendix A. Multivariate wavelets

Here we summarize the wavelet construction for a logically Cartesian grid hierarchy. In this case the extension from univariate wavelets to multivariate wavelets is straightforward. To illustrate this we consider a uniform dyadic partition of $[0,1]^{\mathrm{d}}$, i.e.,

$$V_{j,\mathbf{k}} = [2^{-j}\,k_1,\quad 2^{-j}\,(k_1+1)] \times \ldots \times [2^{-j}\,k_d,\quad 2^{-j}\,(k_d+1)],$$

for $\mathbf{k} \in I_j := \{0, \dots, 2^j - 1\}^{\mathrm{d}}$. Then the refinement sets are determined by $\mathcal{M}_{j,\mathbf{k}} = \{2\mathbf{k} + \mathbf{e}; \mathbf{e} \in E\}$ with $E := \{0, 1\}^{\mathrm{d}}$. In this shift–invariant case, multivariate wavelets can be constructed by tensor products. For this purpose, we introduce the convention $\tilde{\psi}_{j,k,0} := \tilde{\varphi}_{j,k}$ and $\tilde{\psi}_{j,k,1} = \tilde{\psi}_{j,k}$ where $\tilde{\varphi}_{j,k}$ and $\tilde{\psi}_{j,k}$ are the univariate box function and the univariate box wavelet, respectively, see Section 3.1.2. Then the multivariate analogue is determined by

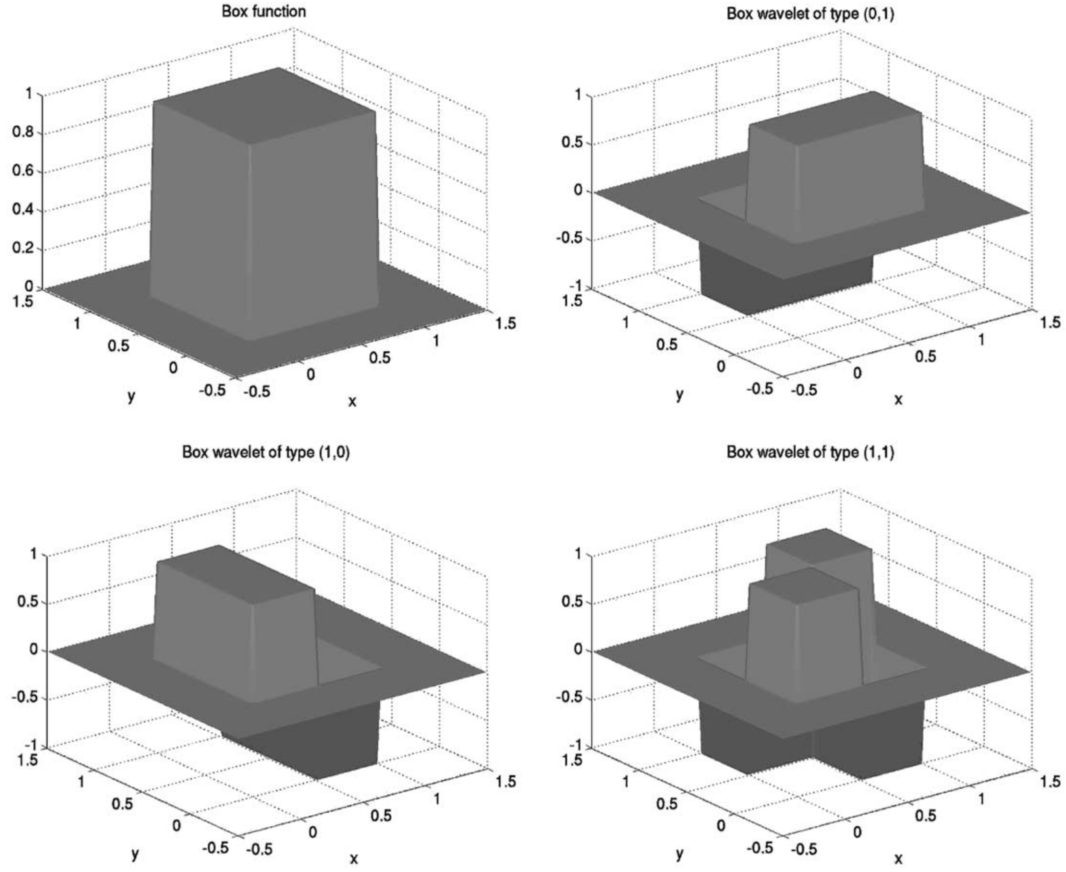
$$\tilde{\varphi}_{j,\mathbf{k}}(\mathbf{x}) := \prod_{i=1}^{\mathrm{d}} \tilde{\psi}_{j,k_i,0}(x_i), \quad \check{\psi}_{j,\mathbf{k},\mathbf{e}}(\mathbf{x}) := \prod_{i=1}^{\mathrm{d}} \tilde{\psi}_{j,k_i,e_i}(x_i), \tag{A.1}$$

where $\mathbf{e} \in E^* := \{0, 1\}^{\mathrm{d}} \setminus \{\mathbf{0}\}$ denotes different wavelet types corresponding to the cell $V_{j,\mathbf{k}}$. These functions are shown in Fig. 24. By means of the two-scale relations for the Haar basis (7)–(9) we derive from the definition of the multivariate functions (A.1) similar relations

$$\tilde{\varphi}_{j,\mathbf{k}} = \sum_{\mathbf{i} \in E} 2^{-d}\, \tilde{\varphi}_{j+1,2\mathbf{k}+\mathbf{i}}, \quad \check{\psi}_{j,\mathbf{k},\mathbf{e}} = \sum_{\mathbf{i} \in E} 2^{-d}\, (-1)^{\mathbf{i}\cdot\mathbf{e}}\, \tilde{\varphi}_{j+1,2\mathbf{k}+\mathbf{i}}, \quad \mathbf{e} \in E^*, \tag{A.2}$$

and

$$\tilde{\varphi}_{j+1,2\mathbf{k}+\mathbf{i}} = \tilde{\varphi}_{j,\mathbf{k}} + \sum_{\mathbf{e} \in E^*} (-1)^{\mathbf{i}\cdot\mathbf{e}}\, \check{\psi}_{j,\mathbf{k},\mathbf{e}}, \quad \mathbf{i} \in E. \tag{A.3}$$


Box function

Box wavelet of type (0,1)

Box wavelet of type (1,0)

Box wavelet of type (1,1)

Fig. 24. Box function and box wavelets on $[0,1]^2$.

As has been proven in [41], p. 77, the relations (A.2) and (A.3) are equivalent.

As explained before, in order to get a better compression by exploiting higher order smoothness we have to raise the order of vanishing polynomial moments. The basic idea is to modify the box wavelet $\check{\psi}_{j,\mathbf{k},\mathbf{e}}$ by some coarse grid box functions, leading to the ansatz

$$\tilde{\psi}_{j,\mathbf{k},\mathbf{e}} := \check{\psi}_{j,\mathbf{k},\mathbf{e}} + \sum_{\mathbf{l}\in\mathscr{L}^{\mathbf{e}}_{j,\mathbf{k}}} l^{j,\mathbf{e}}_{\mathbf{l},\mathbf{k}}\, \tilde{\varphi}_{j,\mathbf{l}}, \quad \mathbf{e}\in E^*, \tag{A.4}$$

with parameters $l^{j,\mathbf{e}}_{\mathbf{l},\mathbf{k}}$ that are yet to be determined. Here the stencil $\mathscr{L}^{\mathbf{e}}_{j,\mathbf{k}} \subset I_j$ denotes the cells $V_{j,\mathbf{l}}$ in the neighborhood of the cell $V_{j,\mathbf{k}}$. Then the parameters $l^{j,\mathbf{e}}_{\mathbf{l},\mathbf{k}}$ are chosen such that

$$\langle p, \tilde{\psi}_{j,\mathbf{k},\mathbf{e}}\rangle_\Omega = 0, \tag{A.5}$$

holds for all polynomials p of degree less than an arbitrary but fixed number M. The details of the construction can be found in [41,59].

By means of (A.2) we can rewrite (A.4) as

$$\tilde{\psi}_{j,\mathbf{k},\mathbf{e}} = \sum_{\mathbf{r}\in\mathscr{M}^1_{j,\mathbf{k}}} m^{j,\mathbf{e}}_{\mathbf{r},\mathbf{k}}\, \tilde{\varphi}_{j+1,\mathbf{r}}, \quad \mathbf{e}\in E^*.$$

Introducing the details by functionals of a function $u \in L_1(\Omega)$, i.e., $d_{j,\mathbf{k},\mathbf{e}} := \langle u, \tilde{\psi}_{j,\mathbf{k},\mathbf{e}}\rangle_\Omega$ we derive two-scale relations similar to (7) and (8)

$$\hat{u}_{j,\mathbf{k}} = \sum_{\mathbf{r}\in\mathscr{M}_{j,\mathbf{k}}} m^{j,0}_{\mathbf{r},\mathbf{k}}\, \hat{u}_{j+1,\mathbf{r}}, \quad d_{j,\mathbf{k},\mathbf{e}} = \sum_{\mathbf{r}\in\mathscr{M}^1_{j,\mathbf{k}}} m^{j,\mathbf{e}}_{\mathbf{r},\mathbf{k}}\, \hat{u}_{j+1,\mathbf{r}}, \quad \mathbf{e}\in E^*, \tag{A.6}$$

for certain mask coefficients $m^{j,\mathbf{e}}_{\mathbf{r},\mathbf{k}}$ and index set $\mathscr{M}^1_{j,\mathbf{k}} \subset I_{j+1}$. By means of (A.3) and (A.4) the corresponding inverse two-scale transformation reads

$$\hat{u}_{j+1,\mathbf{k}} = \sum_{\mathbf{r}\in\mathscr{G}^0_{j,\mathbf{k}}} g^{j,0}_{\mathbf{r},\mathbf{k}}\, \hat{u}_{j,\mathbf{r}} + \sum_{\mathbf{e}\in E^*} \sum_{\mathbf{r}\in\mathscr{G}^1_{j,\mathbf{k}}} g^{j,\mathbf{e}}_{\mathbf{r},\mathbf{k}}\, d_{j,\mathbf{r},\mathbf{e}}, \tag{A.7}$$

which is similar to (10). Here $g^{j,\mathbf{e}}_{\mathbf{r},\mathbf{k}}$ denotes the filter coefficients and $\mathscr{G}^0_{j,\mathbf{k}}, \mathscr{G}^1_{j,\mathbf{k}} \subset I_j$ the index sets corresponding to the support of the box function and the wavelets, respectively.

Finally we would like to comment on general grid hierarchies. In this case it is also possible to construct box wavelets. Similar to (A.4) it is possible to modify these box wavelets such that we obtain wavelets with higher vanishing moments. For details we refer to [59].

References

[1] Test cases for inviscid flow field methods. AGARD-AR-211, 1985.

[2] R. Abgrall, Multiresolution analysis on unstructured meshes: Applications to CFD, in: Chetverushkin et al. (Eds.), Experimentation, Modelling and Computation in Flow, Turbulence and Combustion, John Wiley & Sons, 1997, pp. 147–156.

[3] W.K. Anderson, J.L. Thomas, C.L. Rumsey, Extension and applications of flux-vector splitting to unsteady calculations on dynamic meshes, AIAA Paper 87-1152 (1987).

[4] F. Arandiga, R. Donat, A. Harten, Multiresolution based on weighted averages of the hat function I: Linear reconstruction techniques, SIAM J. Numer. Anal. 36 (1) (1998) 160–203.

[5] F. Arandiga, R. Donat, A. Harten, Multiresolution based on weighted averages of the hat function II: Non-linear reconstruction techniques, SIAM J. Sci. Comput. 20 (3) (1999) 1053–1093.

[6] S. Balay, K. Buschelman, W. Gropp, D. Kaushik, M. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc users manual, Technical Report ANL-95/11-Revision 2.1.5, Argonne National Laboratory, 2002.

[7] J. Ballmann (Ed.), Flow modulation and fluid–structure-interaction at airplane wings, Numerical Notes on Fluid Mechanics, vol. 84, Springer Verlag, 2003.
[8] T.J. Barth. A 3-D Upwind Euler Solver for Unstructured Meshes, AIAA Paper 91-1548-CP, 1991.
[9] T.J. Barth, P.O. Frederickson, Higher order solution of the euler equations on unstructured grids using quadratic reconstruction, AIAA Paper 90-0013 (1990).
[10] T.J. Barth, D.C. Jaspersen, The design and application of upwind schemes on unstructured meshes, AIAA Paper 89-0366 (1989).
[11] J.T. Batina, Implicit flux-split Euler schemes for unsteady aerodynamic analysis involving unstructured dynamic meshes, AIAA J. 29 (11) (1991) 1836–1843.
[12] P. Batten, M.A. Leschziner, U.C. Goldberg, Average-state Jacobians and implicit methods for compressible viscous and turbulent flows, J. Comp. Phys. 137 (1997) 38–78.
[13] M. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comp. Phys. 82 (1989) 64–84.
[14] M. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comp. Phys. 53 (1984) 484–512.
[15] B. Bihari, A. Harten, Application of generalized wavelets: An adaptive multiresolution scheme, J. Comp. Appl. Math 61 (1995) 275–321.
[16] B. Bihari, A. Harten, Multiresolution schemes for the numerical solution of 2-D conservation laws I, SIAM J. Sci. Comput. 18 (2) (1997) 315–354.
[17] B.L. Bihari, D.K. Ota, Z. Liu, S.V. Ramakrishnan, The multiresolution method on general unstructured meshes, AIAA paper (AIAA 2001-2553), 2001.
[18] C. Bischof, A. Carle, P. Khademi, A. Mauer, ADIFOR 2.0: Automatic Differentiation of Fortran 77 Programs, IEEE Computat. Sci. Eng. 3 (3) (1996) 18–32.
[19] W. Böhm, G. Farin, J. Kahmann, A survey of curve in surface methods in CAGD, Computer Aided Geometric Design 1 (1984) 1–60.
[20] K.-H. Brakhage, High quality mesh generation and sparse representation using B-Splines, in: B.K. Soni, J.F. Thompson, J. Häuser, P.Eiseman (Eds.), Proceedings of the Seventh International Conference on Numerical Grid Generation in Computational Field Simulations, Chateau Whistler Resort, British Columbia, 2000, pp. 753–762.
[21] K.-H. Brakhage, Ph. Lamby, CAGD tools for high quality grid generation and sparse representation, in: B.K. Soni, J.F. Thompson, J. Häuser, P.Eiseman (Eds.), Proceddings of the Ninth International Conference on Numerical Grid Generation in Computational Field Simulations, Waikiki Beach, Hawaii, 2002, pp. 599–608.
[22] K.-H. Brakhage, S. Müller, Algebraic-hyperbolic grid generartion with precise control of intersection of angles, Int. J. Numer. Meth. Fluids 33 (2000) 89–123.
[23] F. Bramkamp, B. Gottschlich-Müller, M. Hesse, Ph. Lamby, S. Müller, J. Ballmann, K.-H. Brakhage, W. Dahmen, H-adaptive multiscale schemes for the compressible Navier–Stokes equations – polyhedral discretization, data compression and mesh generation, in: J. Ballmann (Ed.), Flow Modulation and Fluid–Structure-Interaction at Airplane Wings, Numerical Notes on Fluid Mechanics, vol. 84, Springer, 2003, pp. 125–204.
[24] M.O. Bristeau, R. Glowinski, J. Periaux, H. Viviand (Eds.), Notes on Numerical Fluid Mechanics, vol. 18, Vieweg, 1987.
[25] J.M. Carnicer, W. Dahmen, J.M. Pena, Local decomposition of refinable spaces and wavelets, Appl. Comput. Harmon. Anal. 3 (1996) 127–153.
[26] G. Chiavassa, R. Donat, Point value multiresolution for 2D compressible flows, SIAM J. Sci. Comput. 23 (3) (2001) 805–823.
[27] G. Chiavassa, R. Donat, A. Marquina, Fine–mesh numerical simulations for 2D Riemann problems with a multilevel scheme, in: G. Warnecke, H. Freistühler (Eds.), Hyperbolic Problems: Theory, Numerics, Applications, Birkhäuser, 2002, pp. 247–256.
[28] A. Cohen, Wavelets in Numerical Analysis, Handbook of Numerical Analysis, Elsevier, Amsterdam, 1998.
[29] A. Cohen, I. Daubechies, J. Feauveau, Bi-orthogonal bases of compactly supported wavelets, Comm. Pure Appl. Math. 45 (1992) 485–560.
[30] A. Cohen, N. Dyn, S.M. Kaber, M. Postel, Multiresolution finite volume schemes on triangles, J. Comp. Phys. 161 (2000) 264–286.
[31] A. Cohen, S.M. Kaber, S. Müller, M. Postel, Fully adaptive multiresolution finite volume schemes for conservation laws, Math. Comp. 72 (241) (2003) 183–225.
[32] A. Cohen, S.M. Kaber, M. Postel, Multiresolution analysis on triangles: application to gas dynamics, in: G. Warnecke, H. Freistühler (Eds.), Hyperbolic Problems: Theory, Numerics, Applications, Birkhäuser, 2002, pp. 257–266.
[33] W. Dahmen, B. Gottschlich–Müller, S. Müller, Multiresolution schemes for conservation laws, Numer. Math. 88 (3) (2000) 399–443.
[34] C. de Boor, A Practical Guide To Splines, Springer, 1978.
[35] F.J. Deister, Selbstorganisierendes hybrid-kartesisches Netzverfahren zur Berechnung von Strömungen um komplexe Konfigurationen, PhD thesis, VDI, 2002.
[36] M. Delanaye, Polynomial Reconstruction Finite Volume Schemes for the Compressible Euler and Navier–Stokes Equations on Unstructured Adaptive Grids, PhD thesis, Universitè de Liège, 1996.
[37] J.A. Désidéri, R. Glowinski, J. Périaux (Eds.), Hypersonic Flows for Reentry Problems, vol. 2, Springer Verlag, 1991.

[38] L. Dubuc, F. Cantariti, M. Woodgate, B. Gribben, K.J. Badcock, B.E. Richards, A grid deformation technique for unsteady flow computations, Int. J. Numer. Meth. Fluids 32 (2000) 285–311.

[39] G. Farin, Curves and Surfaces in Computer Aided Geometric Design – A practical guide, second ed., Academic Press, 1990.

[40] W.J. Gordon, C.A. Hall, Construction of curvilinear coordinate systems and applications to mesh generation, Int. J. Numer. Meth. Eng. 7 (1973) 461–477.

[41] B. Gottschlich–Müller, Multiscale Schemes for Conservation Laws, PhD thesis, RWTH Aachen, 1998.

[42] B. Gottschlich–Müller, S. Müller, Adaptive finite volume schemes for conservation laws based on local multiresolution techniques, in: M. Fey, R. Jeltsch (Eds.), Hyperbolic Problems: Theory, Numerics, Applications, Birkhäuser, 1999, pp. 385–394.

[43] B. Gottschlich–Müller, S. Müller, On multi-scale concepts for multi-dimensional conservation laws, in: W. Hackbusch, G. Wittum (Eds.), Numerical Treatment of Multi-scale Problems, Vieweg, 1999, pp. 119–133.

[44] D. Hänel, R. Schwane, An implicit flux-vector splitting scheme for the computation of viscous hypersonic flow, AIAA Paper 89-0274 (1989).

[45] A. Harten, Multiresolution algorithms for the numerical solution of hyperbolic conservation laws, Comm. Pure Appl. Math. 48 (12) (1995) 1305–1342.

[46] A. Harten, Multiresolution representation of data: A general framework, SIAM J. Numer. Anal. 33 (3) (1996) 1205–1256.

[47] M. Hesse, G. Britten, J. Ballmann. A multi-block grid deformation algorithm for aeroelastic analysis, in: B.K. Soni, J.F. Thompson, J. Häuser, P. Eiseman (Eds.), Proceedings of the Seventh International Conference on Numerical Grid Generation in Computational Field Simulations, Chateau Whistler Resort, British Columbia, 2000, pp. 161–170.

[48] P. Houston, J.A. Mackenzie, E. Süli, G. Warnecke, A posteriori error analysis for numerical approximations of friedrichs systems, Numer. Math. 82 (1999) 433–470.

[49] R.P. Koomullil, B.K. Soni, Flow simulation using generalized static and dynamic grids, AIAA J. 37 (12) (1999) 1551–1557.

[50] D. Kröner, M. Ohlberger, A posteriori error estimates for upwind finite volume schemes for nonlinear conservation laws in multi dimensions, Math. Comp. 69 (229) (1999) 25–39.

[51] R.H. Landon. NACA0012. Oscillatory and transient pitching, pages 3.1–3.25, Compendium of Unsteady Aerodynamic Measurements, Data Set 3, AGARD Report 702, 1983.

[52] V.D. Liseikin, Grid Generation Methods, Scientific Computation, Springer, 1999.

[53] L.G. Loitsianski, Laminare Grenzschichten, Akademie Verlag, Berlin, 1967.

[54] H. Luo, J.D. Baum, R. Löhner, An accurate, fast, matrix-free implicit method for computing unsteady flows on unstructured grids, AIAA Paper 99-0937 (1999).

[55] R. Massjung, J. Hurka, J. Ballmann, W. Dahmen, On well-posedness and modelling for nonlinear aeroelasticity, in: J. Ballmann (Ed.), Flow Modulation and Fluid–Structure-Interaction at Airplane Wings, Numerical Notes on Fluid Mechanics, vol. 84, Springer, 2003, pp. 227–248.

[56] I.R.M. Moir, Measurements on a Two-Dimensional Aerofoil with High-Lift Devices, AGARD-AR-303 2 (1994) 58–59.

[57] S.A. Morton, R.B. Melville, M.R. Visbal, Accuracy and coupling issues of aeroelastic Navier–Stokes solutions on deforming meshes, J. Aircraft 35 (5) (1998) 798–805.

[58] S. Müller, Adaptive multiresolution schemes, in: B. Herbin (Ed.), Finite Volumes for Complex Applications, Hermes Science, Paris, 2002, pp. 119–136.

[59] S. Müller, Adaptive multiscale schemes for conservation laws, in: Lecture Notes on Computational Science and Engineering, vol. 27, Springer, 2002.

[60] S. Müller, A. Voss, A Manual for the Template Class Library igpm_t_lib, IGPM–Report 197, RWTH Aachen, 2000.

[61] L. Piegl, W. Tiller, The NURBS book, second ed., Springer, 1997.

[62] A. Rault, G. Chiavassa, R. Donat, Shock-vortex interactions at high Mach numbers, J. Comp. Phys. 19 (2003) 347–371.

[63] O. Roussel, K. Schneider, A. Tsigulin, H. Bockhorn, A conservative fully adaptive multiresolution algorithm for parabolic PDEs, J. Comp. Phys. 188 (2) (2003) 493–523.

[64] Y. Saad (Ed.), Iterative Methods for Sparse Linear Systems, PWS, Boston, 1996.

[65] S.P. Spekreijse, B.B. Prananta, J.C. Kok, A Simple, Robust and Fast Algorithm to Compute Deformations of Multi-block Structured Grids, Technical Report NLR-TP-2002-105, National Aerospace Laboratory NLR, 2002.

[66] D. Hempel, T. Sonar, V. Hannemann, Dynamic adaptivity and residual control in unsteady compressible flow computation, Math. Comp. Modell. 20 (1994) 201–213.

[67] E. Süli, T. Sonar, A dual graph-norm refinement indicator for finite volume approximations of the Euler equations, Numer. Math. 78 (1998) 619–658.

[68] V. Venkatakrishnan, Convergence to steady state solutions of the Euler equations on unstructured grids with limiters, J. Comp. Phys. 118 (1995) 120–130.