Adaptive Multiscale Methods for Flow Problems: Recent Developments

Wolfgang Dahmen, Nune Hovhannisyan and Siegfried Müller

Abstract The concept of the new fully adaptive flow solver Quadflow has been developed within the SFB 401 over the past 12 years. Its primary novelty lies in the integration of new and advanced mathematical tools in a unified environment. This means that the core ingredients of the finite volume solver, the grid adaptation and grid generation are adapted to each others needs rather than putting them together as independent black boxes. In this paper we shall present recent developments and demonstrate their efficiency by numerical experiments for some representative basic configurations.

1 Introduction

The work performed in the SFB 401 was motivated by two central questions arising from engineering applications, namely, (i) how to influence wake vortices generated by a lift-producing aircraft in order to reduce takeoff and landing frequencies at airports, and (ii) of better understanding the interaction of structural dynamics and aerodynamics to design new concepts for supporting wing structures. The accurate and reliable simulation of such processes pose challenging questions near or even beyond current simulation capabilities. The development of concepts that reduce computational complexity already on the level of mathematical algorithmic design appears to be indispensible. This has been the core objective of the new adaptive and parallel solver Quadflow [6, 7]. In order to exploit synergy effects, this solver has been designed as an integrated tool where the core ingredients, namely, (i) the flow solver concept based on a finite volume discretization, (ii) the grid adaptation concept based on wavelet techniques, and (iii) the grid generator based on B-spline mappings are adapted to each others needs, see Figure 1. In particular, the three tools

Wolfgang Dahmen, Nune Hovhannisyan and Siegfried Müller
Institut für Geometrie und Praktische Mathematik, RWTH Aachen University, D-52056 Aachen,
e-mail: dahmen,hovhannisyan,mueller@igpm.rwth-aachen.de

are not just treated as independent black boxes communicating via interfaces but are highly intertwined on a conceptual level mainly linking (i) the multiresolution-based grid adaption that reliably detects and resolves all physical relevant effects, and (ii) the B-spline grid generator which reduces grid changes to just moving a "few" control points whose number is, in particular, independent of any local grid refinements.

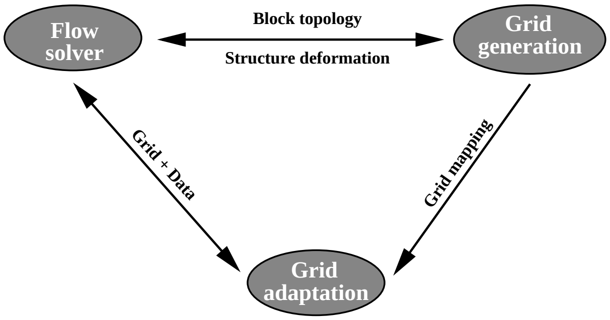


Fig. 1 Quadflow: An integrated concept

Over the past few years the above framework has been further enriched and extended by mathematical concepts, such as multilevel time stepping [20, 19, 23], multilevel solver strategies [24], time step control [29, 27, 28], adaptive local flux and source reconstruction [17], as well as techniques from computer science. In particular, parallelization of the multiresolution grid adaptation using space-filling curves [31], has been incorporated to further improve the efficiency of Quadflow.

Originally, the multiresolution-based grid adaptation technique was kept separate from the treatment of discrete evolution equations, cf. [7]. However, the multiresolution analysis offers a much higher potential when applying it directly to the discrete evolution equations arising from the finite volume discretization rather than just using it as a data compression tool for the set of discrete cell data. In the present work we explain how to integrate the multiresolution anaysis in the flow solver by presenting recent developments on (i) a local strategy to compute numerical fluxes and sources on locally refined grids with hanging nodes, cf. [17], (ii) a multilevel time stepping strategy where refinement is also performed in time, cf. [23], and (iii) an FAS-like strategy to solve efficiently the nonlinear problems arising from an implicit time discretization of the underlying finite volume scheme, cf. [24].

Of course, in spite of the significant reduction of the accomplished computational complexity (Cpu time and memory) in comparison to computations on uniform meshes, efficiently performing 3D computations for complex geometries requires complementing efforts concerning parallelization techniques. The performance of a parallelized code crucially depends on the load-balancing. Since the underlying adaptive grids are unstructured, this task cannot be considered trivial. This issue will be addressed in more detail in [12].

The aim of the present work is to give an overview on recent developments regarding multiscale-based grid adaptation. For this purpose, we first summarize the basic ingredients of the grid adaptation concept starting with the underlying equations and their discretization using finite volume schemes, see Section 2. This is followed by the multiscale analysis of the discrete cell averages resulting from the finite volume discretization, see Section 3 and the construction of locally refined grids using data compression techniques, see Section 4. Applying the multiscale analysis to the original finite volume discretization on the uniform grid we obtain multiscale evolution equations, see Section 5. The crucial point is then to perform the time evolution on the adaptive grid where the accuracy of the uniform discretization is maintained but the computational complexity is proportional only to the number of cells of the adaptive grid. For this purpose, the computation of the local flux balances and sources has to be performed judiciously, see Section 6. The resulting scheme is further accelerated using multilevel time stepping strategies in case of an explicit time discretization for instationary flow problems, see Section 7, and FAS-like multigrid techniques in case of an implicit time discretization for steady state flow problems, see Section 8. Finally, in Section 9, we present several computations that confirm the efficiency of the aforementioned concepts.

2 Governing equations and finite volume schemes

The fluid equations are determined by balance equations

$$\frac{\partial}{\partial t}\int_V \mathbf{u}\ dV + \oint_{\partial V} \mathbf{f}(\mathbf{u})\cdot\mathbf{n}\ dS = \int_V \mathbf{s}(\mathbf{u})\ dV\,, \tag{1}$$

where $\mathbf{u}$ is the array of the mean conserved quantities, e.g., density of mass, momentum, specific total energy, $\mathbf{f}$ is the array of the corresponding convective and diffusive fluxes, and $\mathbf{s}$ denotes a source term that may occur, for instance, in turbulence modelling.

The balance equations (1) are approximated by a finite volume scheme. For this purpose the finite fluid domain $\Omega \subset \mathbf{R}^d$ is split into a finite set of subdomains, the cells V_i, such that all V_i are disjoint at each instant of time and that their union covers Ω. To simplify notation, we will always assume that the grid does not move with time. Furthermore let $N(i)$ be the set of cells that have a common edge with the cell i, and for $j \in N(i)$ let $\Gamma_{ij} := \partial V_i \cap \partial V_j$ be the interface between the cells i and j and $\mathbf{n}_{ij}$ the outer normal of Γ_{ij} corresponding to cell i. For the time discretization we may use either explicit (forward Euler ($\theta = 0$)) or implicit schemes (Crank-Nicholson ($\theta = 0.5$), backward Euler ($\theta = 1$)) where the time step may vary i.e., $t_i^{n+1} = t_i^n + \tau_i^{n+1}$. These can be written in the form

$$v_i^{n+1} + \theta\,\frac{\tau_i^{n+1}}{|V_i|}\,(B_i^{n+1} - |V_i|\,S_i^{n+1}) = v_i^n - (1-\theta)\,\frac{\tau_i^{n+1}}{|V_i|}\,(B_i^n + |V_i|\,S_i^n) \tag{2}$$

to compute the approximated cell averages v_i of the conserved variables on the new time level. Here the fluxes and the source terms are approximated by

$$B_i^n := \sum_{j\in N(i)} |\Gamma_{ij}| F(v_{ij}^n, v_{ji}^n, \mathbf{n}_{ij}), \quad S_i^n := S(v_i^n), \tag{3}$$

where the numerical flux function $F(\mathbf{u},\mathbf{w},\mathbf{n})$ is an approximation for the flux $f(\mathbf{u},\mathbf{n}) := \mathbf{f}\cdot\mathbf{n}$ in outer normal direction $\mathbf{n}_{ij}$ on the edge Γ_{ij}. The numerical flux is assumed to be *consistent*, i.e., $F(\mathbf{u},\mathbf{u},\mathbf{n}) = f(\mathbf{u},\mathbf{n})$. For simplicity of presentation we neglect the fact that, due to higher order reconstruction, it usually depends on an enlarged stencil of cell averages. Moreover, to preserve a constant flow field we assume that the geometric consistency condition $\sum_{j\in N(i)} |\Gamma_{ij}|\mathbf{n}_{ij} = \mathbf{0}$ holds.

3 Multiscale Analysis

A finite volume discretization typically works on an array of cell averages. In order to realize a certain target accuracy at the expense of a possibly low number of degrees of freedom, viz. a possibly low computational effort, one should keep the size of the cells large whereever the data exhibit little variation, reflecting a high regularity of the searched solution components. Our analysis of the local regularity behavior of the data is based on the concept of biorthogonal wavelets [10]. This approach may be seen as a natural generalization of Harten's discrete framework [15]. The core ingredients are (i) a hierarchy of nested grids, (ii) biorthogonal wavelets and (iii) the multiscale decomposition. In what follows we will only summarize the basic ideas. For the realization and implementation see [22].

Grid hierarchy. Let be $\Omega_l := \{V_\lambda\}_{\lambda\in I_l}$ a sequence of different meshes corresponding to different resolution levels $l \in \mathbf{N}_0$ where the mesh size decreases with increasing refinement level. The grid hierarchy is assumed to be *nested*, i.e., each cell $\lambda \in I_l$ on level l is the union of cells $\mu \in M_\lambda^0 \subset I_{l+1}$ on the next higher refinement level $l+1$, i.e.,

$$V_\lambda = \bigcup_{\mu\in M_\lambda^0\subset I_{l+1}} V_\mu, \quad \lambda \in I_{l+1}, \tag{4}$$

where $M_\lambda^0 \subset I_{l+1}$ is the refinement set and, hence, $\Omega_l \subset \Omega_{l+1}$. A simple example is shown in Figure 2 for a dyadic grid refinement of Cartesian meshes. 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, cf. [22].

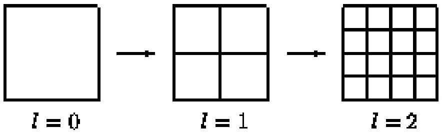


Fig. 2 Sequence of nested grids

Box function and cell averages. With each cell V_λ in the partitions Ω_l we associate the so–called *box function*

$$\tilde{\phi}_\lambda(x) := \frac{1}{|V_\lambda|}\chi_{V_\lambda}(x) = \begin{cases} 1/|V_\lambda| \,, & x \in V_\lambda \\ 0 \,, & x \notin V_\lambda \end{cases} \,, \quad \lambda \in I_l \tag{5}$$

defined as the L^1–normalized characteristic function of V_λ. By $|V|$ we denote the volume of a cell V. Then the averages of a scalar, integrable function $u \in L^1(\Omega)$ can be interpreted as an inner product, i.e.,

$$\hat{u}_\lambda := \langle u, \tilde{\phi}_\lambda \rangle_\Omega \quad \text{with} \quad \langle u, v \rangle_\Omega := \int_\Omega u\, v\, dx. \tag{6}$$

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

$$\tilde{\phi}_\lambda = \sum_{\mu \in M^0_\lambda \subset I_l} m^{l,0}_{\mu,\lambda}\, \tilde{\phi}_\mu \quad \text{and} \quad \hat{u}_\lambda = \sum_{\mu \in M^0_\lambda \subset I_l} m^{l,0}_{\mu,\lambda}\, \hat{u}_\mu, \quad \lambda \in I_{l-1}, \tag{7}$$

where the mask coefficients turn out to be $m^{l,0}_{\mu,\lambda} := |V_\mu|/|V_\lambda|$ for each cell $\mu \in M^0_\lambda$ in the refinement set.

Wavelets and details. In order to detect singularities of the solution we consider the difference of the cell averages corresponding to different resolution levels. For this purpose we introduce the wavelet functions $\tilde{\psi}_\lambda$ as linear combinations of the box functions, i.e.,

$$\tilde{\psi}_\lambda := \sum_{\mu \in M^1_\lambda \subset I_{l+1}} m^{l,1}_{\mu,\lambda}\, \tilde{\phi}_\mu, \quad \lambda \in J_l, \tag{8}$$

with mask coefficients $m^{l,1}_{\mu,\lambda}$ that only depend on the grids. The construction of the wavelets is subject to certain constraints, namely, (i) the wavelet functions $\tilde{\Psi}_l := (\tilde{\psi}_\lambda)_{\lambda \in J_l}$ build an appropriate completion of the basis system $\tilde{\Phi}_l := (\tilde{\phi}_\lambda)_{\lambda \in I_l}$. By this we mean (ii) they are locally supported, (iii) provide vanishing moments of a certain order and (iv) there exists a biorthogonal system Φ_l and Ψ_l of primal functions satisfying analogous two-scale relations. The last requirement is typically the hardest to satisfy. It is closely related to the Riesz basis property of the infinite collection $\tilde{\Phi}_0 \cup \bigcup_{l=0}^{\infty} \tilde{\Psi}_l$ of $L_2(\Omega)$, say. For details we refer to the *concept of stable completions*, see [10].

Aside from these stability aspects the biorthogonal framework facilitates an efficient change of bases. While the relations (7), (8) provide expressions of the coarse scale box functions and detail functions as linear combinations of fine scale box functions, the mask coefficients in the analogous two-scale relations for the dual system Φ_l, Ψ_l give rise to the reverse change of bases between $\tilde{\Phi}_l \cup \tilde{\Psi}_l$ and $\tilde{\Phi}_{l+1}$, i.e.,

$$\tilde{\phi}_\lambda = \sum_{\mu \in G^0_\lambda \subset I_l} g^{l,0}_{\mu,\lambda}\, \tilde{\phi}_\mu + \sum_{\mu \in G^1_\lambda \subset J_l} g^{l,1}_{\mu,\lambda}\, \tilde{\psi}_\mu, \quad \lambda \in I_{l+1}, \tag{9}$$

where we rewrite the basis function $\tilde{\phi}_\lambda$ on level $l+1$ by the scaling functions $\tilde{\phi}_\mu$ and the wavelet functions $\tilde{\psi}_\mu$ on the next coarser scale l. Here again the mask coefficients $g^{l,0}_{\mu,\lambda}$ and $g^{l,1}_{\mu,\lambda}$ depend only on the grid geometry.

Biorthogonality also yields a data representation in terms of the primal system. The corresponding *detail coefficients* are given by

$$d_\lambda := \langle u, \tilde{\psi}_\lambda \rangle_\Omega = \sum_{\mu \in M_\lambda^1} m_{\mu,\lambda}^{l,1}\, \hat{u}_\mu, \quad \lambda \in J_l, \tag{10}$$

whose two-scale format follows from its functional counterpart (8).

Note that, by biorthogonality, the d_λ are the expansion coefficients with respect to the basis Ψ, obtained by testing u by the elements from $\tilde{\Psi}$. Therefore, Ψ is called primal system, while $\tilde{\Psi}$ is used to expand the cell averages which are *functionals* of the solution u whose propagation in time gives rise to the finite volume scheme. The primal basis itself will actually never be used to represent the solution u. Instead the enhanced accuray of the approximate cell averages can be used for higher order reconstructions commonly used in finite volume schemes.

Cancellation Property. It can be shown that the details become small with increasing refinement level when the underlying function is smooth

$$|d_\lambda| \le C\, 2^{-lM}\, \|u^{(M)}\|_{L^\infty(V_\lambda)}. \tag{11}$$

in the support of the wavelet $\tilde{\psi}_\lambda$. More precisely, the details decay at a rate of at least 2^{-lM} provided the function u is differentiable and the wavelets have vanishing moments of order M, i.e., $\langle p, \tilde{\psi}_\lambda \rangle_\Omega = 0$ for all polynomials p of degree less than M. Here we assume that the grid hierarchy is quasi-uniform in the sense that the diameters of the cells on each level l is proportional to 2^{-l}.

If coefficient and function norms behave essentially the same, as asserted by the Riesz basis property, (11) suggests to neglect all sufficiently small details in order to compress the original data. In fact, the higher M the more details may be discarded in smooth regions.

Multiscale Transformation. In order to exploit the above compression potential, the idea is to transform the array of cell averages $u_L := (\hat{u}_\lambda)_{\lambda \in I_L}$ corresponding to a finest uniform discretization level into a sequence of coarse grid data $u_0 := (\hat{u}_\lambda)_{\lambda \in I_0}$ and details $d_l := (d_\lambda)_{\lambda \in J_l}$, $l = 0, \ldots, L-1$, representing the successive update from a coarser resolution to a higher resolution.

In summary, according to (7) and (10), the change of bases provides two–scale relations for the coefficients inherited from the two–scale relations of the box functions and the wavelet functions

$$\hat{u}_\lambda = \sum_{\mu \in M_\lambda^0 \subset I_{l+1}} m_{\mu,\lambda}^{l,0}\, \hat{u}_\mu,\ \lambda \in I_l, \qquad d_\lambda = \sum_{\mu \in M_\lambda^1 \subset I_{l+1}} m_{\mu,\lambda}^{l,1}\, \hat{u}_\mu,\ \lambda \in J_l, \tag{12}$$

and, conversely,

$$\hat{u}_\lambda = \sum_{\mu \in G_\lambda^0 \subset I_l} g_{\mu,\lambda}^{l,0}\, \hat{u}_\mu + \sum_{\mu \in G_\lambda^1 \subset J_l} g_{\mu,\lambda}^{l,1}\, d_\mu, \qquad \lambda \in I_{l+1}, \tag{13}$$

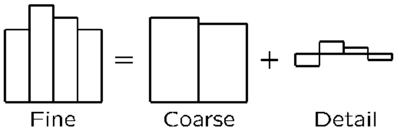


Fig. 3 Two-scale Transformation

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

Fig. 4 Multiscale transformation

which reflects the typical cascadic format of a wavelet transform. The two-scale relations are illustrated for the 1D case in Figure 3.

A successive application of the relations (12), see Figure 4, decomposes the array $\hat{u}_L$ into coarse scale averages and higher level fluctuations. We refer to this transformation as *multiscale transformation*. It is inverted by the *inverse multiscale transformation* (13).

4 Multiscale-based spatial grid adaptation

To determine a locally refined grid we employ the above multiscale decomposition. The basic idea is to perform data compression on the vector of detail coefficients using hard thresholding as suggested by the cancellation property. This will significantly reduce the complexity of the data. Based on the thresholded array we then perform local grid adaptation where we refine a cell whenever there exists a significant detail, i.e. a detail coefficient with absolute value above the given threshold.. The main steps in this procedure are summarized as follows:

Step 1: Multiscale analysis. Let be v_L^n the cell averages representing the discretized flow field at some fixed time level t^n on a given locally refined grid with highest level of resolution $l = L$. This sequence is encoded in arrays of *detail coefficients* d_l^n, $l = 0,\dots,L-1$ of ascending resolution, see Figure 4, and cell averages on some coarsest level $l = 0$. For this purpose the multiscale transformation (12) need to be performed *locally* which is possible due to the locality of the mask coefficients.

Step 2: Thresholding. In order to compress the original data we discard all detail coefficients d_λ whose absolute values fall below a level-dependent threshold value $\varepsilon_l = 2^{l-L}\varepsilon$. Let

$$D_{L,\varepsilon}^n := \left\{\lambda \; ; \; |d_\lambda^n| > \varepsilon_l, \; \lambda \in I_l, \; l \in \{0,\dots,L-1\}\right\}$$

be the set of *significant details*. The ideal strategy would be to determine the threshold value ε such that the *discretization error* of the reference scheme, i.e., difference between exact solution and reference scheme, and the *perturbation error*, i.e., the difference between the reference scheme and the adaptive scheme, are balanced. For a detailed treatment of this issue we refer to [11].

Step 3: Prediction and grading. Since the flow field evolves in time, grid adaptation is performed after each evolution step to provide the adaptive grid at the *new* time level. In order to guarantee the adaptive scheme to be *reliable* in the sense that no significant future feature of the solution is missed, we have to *predict* all significant details at the new time level $n+1$ by means of the details at the *old* time level n. Let $\tilde{D}^{n+1}_{L,\varepsilon}$ be the prediction set satisfying the *reliability condition*

$$D^{n}_{L,\varepsilon} \cup D^{n+1}_{L,\varepsilon} \subset \tilde{D}^{n+1}_{L,\varepsilon}. \tag{14}$$

Basically there are two prediction strategies (i.e. ways of choosing $\tilde{D}^{n+1}_{L,\varepsilon}$) discussed in the literature, see [14, 11]. Moreover, in order to manage grid adaptation this set is additionally inflated somewhat so that the grid refinement history, i.e. the parent-child relations of subdivided cells correspond to a *graded* tree, i.e. transitions between cells of different levels are sufficiently gradual. The connection with trees whose leaves form the grid partition and whose interior nodes are the refined cells will be addressed later again in some more detail.

Step 4: Grid adaptation. By means of the set $\tilde{D}^{n+1}_{L,\varepsilon}$ a locally refined grid is determined along the following lines. We check the transformed flow data represented on $\tilde{D}^{n+1}_{L,\varepsilon}$ proceeding levelwise from coarse to fine whether the detail associated with any cell marked by the prediction set is significant. If it is we refine the respective cell. We finally obtain the locally refined grid with hanging nodes represented by the index set $\tilde{G}^{n+1}_{L,\varepsilon}$. The flow data on the new grid can be computed from the detail coefficients in the same loop where we apply locally the inverse multiscale transformation (13).

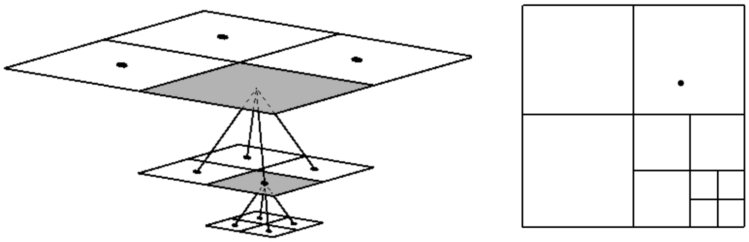


Fig. 5 Grid adaptation: refinement tree (left) and corresponding adaptive grid (right)

5 Adaptive multiresolution finite volume schemes

The rationale behind our design of adaptive multiresolution finite volume schemes (MR-FVS) is to accelerate a given finite volume scheme (reference scheme) on a uniformly refined mesh (reference mesh) through computing actually *only* on a locally refined adapted subgrid, while preserving (up to a fixed constant multiple) the accuracy of the discretization on the full uniform grid. We shall briefly indicate now how to realize this strategy with the aid of the ingredients discussed in the previous section.

The conceptual starting point is to rewrite the evolution equations (2) for the cell averages, i.e., $v_k = v_{L,k}$ for $k \in I_L$, of the reference scheme in terms of evolution equations for the multiscale coefficients. For this purpose we apply the multiscale transformation (12) to the set of evolution equations (2). Then we discard all equations that do not correspond to the prediction set $\tilde{D}_{L,\varepsilon}^{n+1}$ of significant details. Finally we apply locally the inverse multiscale transformation (13) and obtain the evolution equations for the cell averages on the adaptive grid $\tilde{G}_{L,\varepsilon}^{n+1}$ which is obtained from $\tilde{D}_{L,\varepsilon}^{n+1}$ as explained before:

$$v_\lambda^{n+1} + \Theta\,\lambda_\lambda\,(\overline{B}_\lambda^{n+1} - |V_\lambda|\,\overline{S}_\lambda^{n+1}) = v_\lambda^n - (1-\Theta)\,\lambda_\lambda\,(\overline{B}_\lambda^n + |V_\lambda|\,\overline{S}_\lambda^n), \tag{15}$$

for all $\lambda \in \tilde{G}_{L,\varepsilon}^{n+1}$ where $\lambda_\lambda := \Delta t^{n+1}/|V_\lambda|$. Here the flux balances $\overline{B}_\lambda^n$, the numerical fluxes $\overline{F}_\lambda^n$ and the source terms $\overline{S}_\lambda^n$ are recursively defined from fine to coarse scale via

$$\overline{B}_\lambda^n = \sum_{\Gamma_{\lambda,\mu}^l \subset \partial V_\lambda} |\Gamma_{\lambda,\mu}^l|\,\overline{F}_{\lambda,\mu}^{l,n}, \tag{16}$$

$$\overline{F}_{\lambda,\mu}^{l,n} = \sum_{\Gamma_{\mu,\nu}^{l+1} \subset \Gamma_{\lambda,\mu}^l} |\Gamma_{\mu,\nu}^{l+1}|\,\overline{F}_{\mu,\nu}^{l+1,n} = \ldots = \sum_{\Gamma_{\mu,\nu}^L \subset \Gamma_{\lambda,\mu}^l} |\Gamma_{\mu,\nu}^L|\,F(v_{L,\mu\nu}^n, v_{L,\nu\mu}^n, \mathbf{n}_{L,\mu\nu}), \tag{17}$$

$$\overline{S}_\lambda^n = \sum_{V_\mu \subset V_\lambda, \mu \in I_{l+1}} \frac{|V_\mu|}{|V_\lambda|}\overline{S}_\mu^n = \ldots = \sum_{V_\mu \subset V_\lambda, \mu \in I_L} \frac{|V_\mu|}{|V_\lambda|} S(v_\mu^n). \tag{18}$$

We refer to (17) and (18) as *exact flux and source reconstruction*, respectively. Since in (18) we have to compute *all* sources on the finest scale, there is no complexity reduction, i.e., we still have the complexity of the reference grid. In order to gain in efficiency we therefore have to replace the exact flux and source reconstruction by some approximation such that the overall accuracy is maintained. This will be discussed in detail in Section 6.

The complete adaptive scheme consists now of the following three steps:

Step 1. (*Refinement*) Determine the prediction set $\tilde{D}_{L,\varepsilon}^{n+1}$ from the data of the old time time step t_n and project the data of the old time step to the pre-refined grid $\tilde{G}_{L,\varepsilon}^{n+1}$ of the new time level, i.e.,

$$\{v_\lambda^n\}_{\lambda \in G^n} \to \{v_\lambda^n\}_{\lambda \in \tilde{G}^{n+1}}.$$

Step 2. (*Evolution*) Evolve the cell averages associated to the pre-refined grid $\tilde{G}_{L,\varepsilon}^{n+1}$ according to (15) where the numerical fluxes and sources are not necessarily determined by (17) and (18), respectively, i.e.,

$$\{v_\lambda^n\}_{\lambda \in \tilde{G}_{L,\varepsilon}^{n+1}} \to \{v_\lambda^{n+1}\}_{\lambda \in \tilde{G}_{L,\varepsilon}^{n+1}}.$$

Step 3. (*Coarsening*) Compress the data of the new time level by thresholding the corresponding detail coefficients and project the data to the (somewhat coarsened new) adaptive grid $G_{L,\varepsilon}^{n+1}$, i.e.,

$$\{v_\lambda^{n+1}\}_{\lambda\in\tilde{G}_{L,\varepsilon}^{n+1}} \to \{v_\lambda^{n+1}\}_{\lambda\in G_{L,\varepsilon}^{n+1}}.$$

The performance of the adaptive MR-FVS crucially depends on the threshold parameter ε. With decreasing value the adaptive grid becomes richer and, finally, if ε tends to zero, we obtain the uniform reference mesh, i.e., the adaptive scheme coincides with the reference scheme. On the other hand, the adaptive grid becomes coarser with increasing threshold value, i.e., the computation becomes faster but provides a less accuracte solution. An ideal choice would maintain the accuracy of the reference scheme at reduced computational cost. For a detailed analysis we refer to [11, 17] and explain here only the main ideas..

In order to estimate the error, we introduce the averages $\hat{u}_L^n$ of the exact solution, the averages v_L^n determined by the reference FVS and the averages $\overline{v}_L^n$ of the adaptive scheme prolongated to the reference mesh by means of the inverse multiscale transformation where non-significant details are simply set to zero. Ideally one would like to choose the threshold ε so as to guarantee that $\|\hat{u}_L^n - \overline{v}_L^n\| \le tol$ where tol is a given target accuracy and $\|\cdot\|$ denotes the standard weighted l^1-norm. Since $\overline{v}_L^n$ can be regarded as a perturbation of v_L^n this is only possible if L is chosen so as to ensure that the reference scheme is sufficiently accurate, i.e. one also has $\|\hat{u}_L^n - v_L^n\| \le tol$. Again ideally, a possibly low number of refinement levels L should be determined during the computation such that the error meets the desired tolerance $\|\hat{u}_L^n - \overline{v}_L^n\| \le tol$. Since no explicit error estimator is available for the adaptive scheme, we try to assess the error by splitting the error into two parts corresponding to the *discretization error* $\tau_L^n := \hat{u}_L^n - v_L^n$ of the reference FVS and the *perturbation error* $e_L^n := v_L^n - \overline{v}_L^n$. We now assume that there is an a priori error estimate of the discretization error, i.e., $\tau_L^n \sim h_L^\alpha$ where h_L denotes the spatial step size and α the convergence order. Then, ideally we would determine the number of refinement levels L such that $h_L^\alpha \sim tol$. In order to preserve the accuracy of the reference FVS we may now admit a perturbation error which is proportional to the discretization error, i.e., $\|e_L^n\| \sim \|\tau_L^n\|$. From this, one can derive a suitable level $L = L(tol,\alpha)$ and $\varepsilon = \varepsilon(L)$.

Therefore it remains to verify that the perturbation error can be controlled. To this end, note that in each time step we introduce an error due to the thresholding procedure. Obviously, this error accumulates in each step, i.e., the best we can hope for is an estimate of the form $\|e_L^n\| \le Cn\varepsilon$. However, the threshold error may be amplified in addition by the evolution step. In order to control the cumulative perturbation error, we have to prove that the constant C is independent of L, n, τ and ε.

For this purpose, we have to choose a prediction strategy satisfying the reliability condition (14). In [14], a heuristic approach was suggested, taking into account the finite speed of propagation and the steepening of gradients that are characteristic for hyperbolic problems. So far the reliability condition (14) could not be rigorously verified for this approach. However, in [11] a slight modification of Harten's prediction strategy was shown to lead to a reliable prediction strategy in the sense of (14) for homogeneous conservation laws using exact flux reconstruction. Recently, in [17] this could be extended to inhomogeneous conservation laws where an approximate flux and source reconstruction strategy was used.

6 Approximate flux and source approximation strategies

As already mentioned above, the adaptive MR-FVS with exact flux and source reconstruction (17) and (18) will have the same complexity as the reference scheme performed on the reference mesh. If there is no inhomogeneity, i.e., $s \equiv 0$, then the complexity of the resulting algorithm might be significantly reduced from the cardinality of the reference mesh to the cardinality of the refined mesh. To see this we note that, due to the nestedness of the grid hierarchy and the conservation property of the numerical fluxes, the coarse-scale flux balances are only computed by the fine-scale fluxes corresponding to the edges of the coarse cell, see (17). Those in turn, have to be determined by the fine scale data. However, the internal fluxes cancel and, hence, the overall complexity is reduced. For instance, for a d-dimensional Cartesian grid hierarchy we would have to compute $2d\,2^{(L-l)(d-1)}$ fluxes corresponding to all fine-scale interfaces $\mu \in I_L$ with $\partial V_\mu \subset \partial V_\lambda$ where $\lambda \in I_l$, due to the subdivision of the cell faces. Note that in both cases missing data on the finest scale have to be determined where we locally apply the inverse two-scale transformation. This is illustrated in Figure 6. On the other hand, the coarse scale sources can be computed similarly with the aid of the recursive formulae (18). Here, however, we have to compute *all* sources on the finest scale which at the first glance prevents the desired complexity reduction.

Hence the adaptive scheme with both exact flux and source reconstruction is useless for practical purposes. However, in the reliability analysis one may perform the adaptive scheme with some approximate flux and source recosntruction to be considered as a further perturbation of the "exact" adaptive scheme.

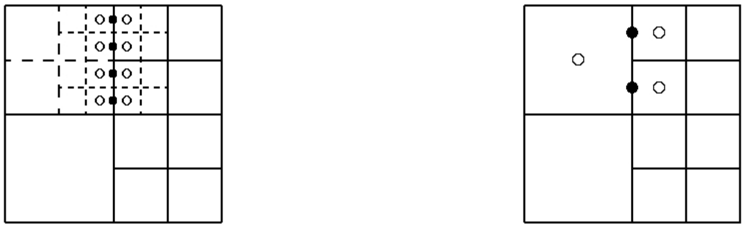


Fig. 6 Exact (left) versus local (right) flux and source computation

In order to retain efficiency we therefore have to replace the exact flux and source reconstruction by some approximation such that the overall accuracy is maintained. A naive approach would be to use the local data provided by the adaptive grid, i.e.,

$$\overline{F}^{l,n}_{\lambda,\mu} = F(v^n_{l,\lambda\mu}, v^n_{l,\mu\lambda}, \mathbf{n}_{l,\lambda\mu}), \quad \overline{S}^n_\lambda = S(v^n_\lambda) \tag{19}$$

for $\lambda, \mu \in I_l$.

So far, this approach is applied in Quadflow. Obviously, the complexity of the resulting adaptive MR-FVS is reduced to the cardinality of the adaptive grid. Unfortunately, this approach may suffer from serious loss in accuracy in comparison with the reference scheme. This is demonstrated by a performance study in Section 9.3 for a model problem.

Recently, in [17] a new approach was suggested using an approximate flux and source reconstruction strategy along the following lines:

Step 1. Determine for each cell V_λ, $\lambda \in \tilde{G}_{L,\varepsilon}^{n+1}$, a higher order reconstruction polynomial R_λ^N of degree N using only local data corresponding to the adaptive grid.
Step 2. Approximate the boundary and volume integrals in (17) and (18) by some appropriate quadrature rules.
Step 3. Compute fluxes and source terms in quadrature knodes by determining point-values or cell averages on level L of the local reconstruction polynomial R_λ^N, respectively.

In [17], this concept has been analyzed in detail for the 1D case. In particular, it was proven that the accuracy of the reference scheme can be maintained when using the prediction strategy in [11] and judisciously tuning the parameters such as the reconstruction order and the quadrature rules. In Section 9.3, we will demonstrate that this new approach is superior to the naive approach with respect to accuracy and efficiency.

7 Multilevel Time Stepping

For an explicit time discretization the time step size is bounded due to the CFL condition by the smallest cell in the grid. Hence Δt is determined by the highest refinement level L, i.e., $\Delta t = \tau_L$. However, for cells on the coarser scales $l = 0, \dots, L-1$ we may use $\Delta t = \tau_l = 2^{L-l}\,\tau_L$ to satisfy locally the CFL condition. In [23] a multilevel time stepping strategy has been incorporated into the adaptive multiscale finite volume scheme. This strategy has been extended to multidimensional problems in [20, 19]. Here ideas similar to the predictor-corrector scheme [25] and the Adaptive Mesh Refinement (AMR) technique [5, 4] are used. The differences between the classical approaches and the multilevel strategy are discussed in detail in [23].

The basic idea is to save flux evaluations where the local CFL condition allows a large time step. The precise time evolution algorithm is schematically described by Fig. 7: In a global time stepping, i.e., using $\Delta t = \tau_L$ for all cells, each vertical line section appearing in Fig. 7 (left) represents a flux evaluation and each horizontal line (dashed or solid) represents a cell update of u due to the fluxes. In the multilevel time stepping a flux evaluation is only performed at vertical line sections that emanate from a point where at least one solid horizontal line section is attached. If a vertical line section emanates from a point, where two dashed horizontal sections are attached, then we do not recompute the flux, but keep the flux value from the preceeding vertical line section. Hence fluxes are only computed for the vertical edges in Fig. 7 (right).

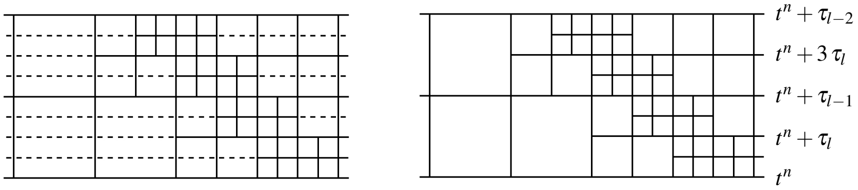


Fig. 7 Synchronized time evolution on space-time grid

In case of the multilevel time stepping we perform 2^L intermediate time steps with step size τ_L, i.e., one macro time step corresponds to the time interval $\tau_0 = 2^L\,\tau_L$. On each intermediate time level (horizontal lines) u is updated for *all* cells. Since not all fluxes have to be recomputed, we can save significantly in CPU time. Furthermore, for each even intermediate time level, i.e., at $t^n + k\,\tau_L$ for $k \in \{2,4,\ldots,2^L\}$ we perfom the multiscale-based grid adaption but only for the levels $l = l_k,\ldots,L$ where $l_k = \min\{l\,:\,0 \le l \le L,\ k \bmod 2^{L-l} = 0\}$ is the smallest synchronization level. This partial grid adaptation procedure ensures that a discontinuity can be tracked on the intermediate time levels instead of a–priori refining the whole range of influence, see Fig. 7 (right).

8 FAS-like multilevel scheme

In the present work, we are interested in combining the multiscale-based grid adaptation with multigrid techniques to solve efficiently the nonlinear system (16) arising from the implicit time discretization of the underlying finite volume scheme. First work on adaptive multigrid techniques has been reported by Brandt [8, 9] who introduced the so-called multilevel adaptive technique (MLAT) that is an adaptive generalization of the full approximation scheme (FAS). The fast adaptive composite grid method (FAC) [13, 21] can be regarded as an alternative to the MLAT approach. An overview on multigrid methods can be found in the review book [30]. In contrast to classical adaptive multigrid schemes we employ the multiscale transformation (12) and (13) using biorthogonal wavelets to define the restriction and prolongation operators, respectively. Since the underlying problem is nonlinear we choose the FAS [8] for the coarse grid correction.

Note that similar investigations have been published in [18] where classical AMR techniques are used for grid adaptation and the standard FAS method is extended to locally refined grids. The definition of *composite residuals* turned out to be crucial in this concept whereas they are easily determined from the multiscale analysis in our strategy.

In order to solve the nonlinear system (16) arising from the implicit time discretization on locally refined grids in one evolution step we combine the FAS strategy [8] with the multiresolution analysis. The main ingredients are (i) the smoother to damp high frequencies, (ii) the restriction and prolongation operator to transfer data from coarse to fine and vice versa and (iii) the coarse grid problem to perform the coarse grid correction. All of them are operating on adaptively refined grids that are composed of cells in the underlying grid hierarchy. To describe them properly we have to distinguish between (i) the cells of the *adaptive grid* that are levelwise characterized by the index sets $G_l \subset I_l$, $l = 0,\dots,L$, and (ii) the cells in the grid hierarchy that are being refined during the adaptation procedure; these are characterized by the *significant details* that are levelwise determined by the index sets D_l, $l = 0,\dots,L-1$, and $D_L := \emptyset$. Then the *adaptive grid* G is the union $G = \bigcup_{l=0,\dots,L} G_l$ of the index sets G_l, $l = 0,\dots,L$. Furthermore the *composite set* T is composed of all cells in the adaptive grid *and* the cells characterized by significant details, i.e., it is the union $T := \bigcup_{l=0,\dots,L} T_l$ of the composite index sets $T_l := G_l \cup D_l$ on level $l = 0,\dots,L$ with $G_l \cap D_l = \emptyset$. The above collection of cells and index sets, respectively, can be interpreted in terms of a graded tree where the adaptive grid G corresponds to the leaves of this tree and the non-leaves (interior nodes) correspond to the significant details D. The composite collection T is the union of both, i.e., the tree itself. For an illustration see Figure 5 (left). Note that we suppress the time index n for simplification of representation.

Smoothing. To smooth the data on level l we perform μ Newton steps, i.e.,

$$N_l'(\overline{v}^{(i)})\,\Delta\,\overline{v}^{(i)} = -N_l(\overline{v}^{(i)})\,v_l^n, \quad \overline{v}^{(i+1)} = \overline{v}^{(i)} + \Delta\,\overline{v}^{(i)}, \quad i = 0,\dots,\mu-1, \tag{20}$$

with initial data $\overline{v}^{(0)} = v_l^m$ given by the mth FAS cycle . Here the nonlinear operator N_l is determined by the discrete evolution equations (16) of the implicit finite volume scheme for the data corresponding to the composite grid T_l on level l, i.e.,

$$(N_l\,v)_\lambda = v_\lambda + \frac{\tau_l}{|V_\lambda|}\,(B_\lambda - |V_\lambda|\,S_\lambda), \quad \lambda \in T_l,$$

where we usually use the naive flux and source reconstruction strategy (19). The linear systems is solved iteratively using GMRES with ILU(2) pre-conditioner. For this purpose we employ the PETSc software library of Argonne National Laboratory [3, 1, 2]. The iteration terminates when the residual is dropping below the tolerance $tol = 1.e-8$ or the maximum number of 100 relaxation steps is exceeded.

Restriction. Due to the nestedness of the underlying grid hierarchy the restriction operator $I_{l+1}^{l} : T_{l+1} \to D_l$ is naturally defined by

$$v_\lambda = \sum_{\mu \in M_\lambda^0} \frac{|V_\mu|}{|V_\lambda|} v_\mu \tag{21}$$

according to (7). This relation holds for all cells. However, the restriction is performed on level l only for those cells that have been refined since we are working on locally adapted grids. These are characterized by the set D_l of significant details. Furthermore we note that by the restriction the adaptive grid I_l on level l is inflated by the new data corresponding to D_l. This is the *composite grid* T_l on level l.

Prolongation. For the prolongation of data $\mu \in I_{l+1}$ from level l to level $l+1$ we employ the inverse two-scale transformation (13) where we put the details to zero, i.e.,

$$v_\mu = \sum_{\mu \in G^0_\mu} g^{l,0}_{\lambda,\mu} v_\mu. \tag{22}$$

This prolongation can be considered a higher order polynomial reconstruction of fine grid data by coarse grid data provided the underlying wavelets have sufficiently high vanishing moments. Note that the prolongation operator $I^{l+1}_l : D_l \to T_{l+1}$ is only applied to cells of the composite grid T_l on level l that are refined according to the significant details D_l.

Coarse grid problem. Let us assume that we have some approximation $v_l = (v_\lambda)_{\lambda \in T_l}$ and $v_{l-1} = (v_\lambda)_{\lambda \in I_{l-1}}$ on level $k = l-1, l$ and some right hand side $f_l = (f_\lambda)_{\lambda \in T_l}$. To set up the nonlinear problem on the coarser level $l-1$ we first have to determine the residual of the nonlinear problem on level l, i.e., the defect. For this purpose we compute the nonlinear operator N_l by means of the given data v_l, i.e.,

$$(N_l\, v_l)_\lambda = v_\lambda + \frac{\tau_l}{|V_\lambda|}(B_\lambda - |V_\lambda|\, S_\lambda), \quad \lambda \in T_l.$$

Note that for the computation of the flux balances B_λ we resort also to data of the adaptive grid on coarser scales. Then the defect on level l is determined by

$$\overline{d}_\lambda = f_\lambda - (N_l\, v_l)_\lambda, \quad \lambda \in T_l.$$

The defect data should not be confused with the detail coefficients of the multiscale decomposition.

Next we apply the restriction operator I^{l-1}_l to the defect (array) $\overline{d}_l$ and to the data v_l, i.e.,

$$\overline{d}_\lambda = (I^{l-1}_l\, d_l)_\lambda \quad \text{and} \quad v_\lambda = (I^{l-1}_l\, v_l)_\lambda, \quad \lambda \in D_{l-1}.$$

Note that the restriction of the latter will not interfere with the given data v_{l-1} because $D_{l-1} \cap G_{l-1} = \emptyset$. Therefore, we may concatenate the data on level $l-1$, i.e., $\overline{v}_{l-1} = (v_\lambda)_{\lambda \in G_{l-1} \cup D_{l-1}}$. Furthermore we employ the same restriction operator for both the defect and the data. In other approaches, it is suggested to use different operators.

We then determine the right hand side f_{l-1} on the coarse scale $l-1$ by means of the coarse grid data $\overline{v}_{l-1}$. For this purpose we first compute the nonlinear operator N_{l-1}

$$(N_{l-1}\, v_{l-1})_\lambda = v_\lambda + \frac{\tau_{l-1}}{|V_\lambda|}(B_\lambda - |V_\lambda|\, S_\lambda), \quad \lambda \in T_{l-1},$$

where again we may access to data of the adaptive grid on coarser scales to compute the flux balances. Then the right hand side f_{l-1} is determined by

$$f_\lambda = \overline{d}_\lambda + (N_{l-1}\, v_{l-1})_\lambda, \quad \lambda \in D_{l-1}, \tag{23}$$

for the cells on level $l-1$ that are being refined and

$$f_\lambda = r^n_\lambda, \quad \lambda \in I_{l-1}, \tag{24}$$

for the non-refined cells in the adaptive grid. Here r^n_λ is the residual corresponding to the old time level t^n, i.e.,

$$r^n_\lambda = v^n_\lambda, \quad \lambda \in G_{l-1}. \tag{25}$$

Then the coarse grid problem is given by

$$(N_{l-1}\, w_{l-1})_\lambda \equiv w_\lambda + \frac{\tau_{l-1}}{|V_\lambda|}\,(B_\lambda - |V_\lambda|\, S_\lambda) = f_\lambda, \quad \lambda \in T_{l-1}. \tag{26}$$

Adaptive FAS cycle. Finally, we describe one iteration step $m \longrightarrow m+1$ of the adaptive multilevel cycle

$$v^{m+1}_l = \text{ADAPCYCLE}(l, \gamma, v^m_{l-1}, v^m_l, N_l, f_l, \mu_1, \mu_2)$$

in terms of the above ingredients. Here we restrict ourselves to the adaptive two-scale case with given data v^m_k on level $k = l$ (fine grid) and on level $k = l-1$ (coarse grid) corresponding to G_l and G_{l-1}, respectively. The iteration cycle is initialized by the data on the adaptive grid at time level t^n. From these we compute the residuals r^n_λ, $\lambda \in G_k$, according to (25) that are stored in the right hand side terms f_k: We start with performing μ_1 smoothing steps (20) on the data v^m_l of level l. Next we perform the coarse-grid correction. For this purpose, we first compute the defect $\overline{d}^m_l$ from the relaxed data $\overline{v}^m_l$. The defect as well as the relaxed data are restricted from T_l to D_{l-1} according to (21). Note that there exist data v^m_{l-1} of the adaptive grid on level $l-1$ that are complemented by the restricted data on T_{l-1}. From this we compute the right hand side f^m_{l-1} where we have to distinguish between cells of the adaptive grid G_{l-1} and the refined cells D_{l-1} on level $l-1$, see (23) and (24). The coarse grid problem (26) is then iteratively solved by the Newton scheme (20) or, if there are additional scales, we recursively apply the algorithm again to the coarser scale $l-1$. The current solution on the adaptive grid on level $l-1$ is then replaced by the coarse grid solution $\hat{w}^m_{l-1}$ whereas for the refined cells D_{l-1} the correction $\hat{v}^m_{l-1} = \hat{w}^m_{l-1} - \overline{v}^m_{l-1}$ is computed. The latter is interpolated to T_l using (22) and the relaxed data are updated by the interpolation $\hat{v}^m_l$. On the corrected approximation $v^{m,cgc}_l = \overline{v}^m_l + \hat{v}^m_l$ we again perform μ_2 smoothing steps. The algorithm is sketched in Figure 8.

$$\begin{tikzcd}
v^m_l \arrow[r, "\text{SMOOTH}"'] & \hat{v}^m_l \arrow[r] \arrow[d, "I^{l-1}_l"] & \overline{d}^m_l = f_l - N_l \overline{v}^m_l \arrow[d, "I^{l-1}_l"] & \hat{v}^m_l \arrow[r] & \overline{v}^m_l + \hat{v}^m_l \arrow[r, "\text{SMOOTH}"'] & v^{m+1}_l \\
 & \hat{v}^m_{l-1} & \overline{d}^m_{l-1} \arrow[r] & N_{l-1}\,(\overline{v}^m_{l-1} + \hat{v}^m_{l-1}) = f^m_{l-1} \arrow[u, "I^l_{l-1}"] & &
\end{tikzcd}$$

Fig. 8 FAS $(l, l-1)$ two-grid method

9 Numerical results

The performance resulting from the above concepts, namely, the multilevel time stepping, the multilevel solver and the flux and source reconstruction strategies, will be investigated now by means of different test configurations in 1D and 2D: an instationary flow over an oscillating plate, a steady state computation over a bump and the solution of the inhomogeneous, inviscid Burgers' equation, respectively. For all computations the multiscale analysis is based on biorthogonal wavelets with $M = 3$ vanishing moments. For the prediction we apply Harten's original strategy, cf. [14].

9.1 Multilevel time stepping: Oscillating plate

The multilevel time stepping strategy in combination with the naive flux reconstruction strategy (19) is investigated for an inviscid flow over an oscillating plate with prescribed deformation in time. The deformation is determined by

$$w(t,x) = \frac{\alpha}{l} sin(2\pi t \beta) * sin(\pi x/l)$$

with amplitude $\alpha = 0.2$, panel length $l = 1$ and frequency $\beta = 1/2\pi$. The flow domain extends from -5 to 5 in x-direction and from 0 to 5 in y-direction. At time $t = 0$ a periodic oscillation in the interval [0,1] is initiated at the the lower boundary. The simplicity of the geometry allows us to employ transfinite interpolation techniques for deforming the grid. Although the multiscale-based grid adaptation and the multilevel time stepping strategies have been outlined here only for stationary flow domains, these can be extended to moving grids by using an ALE formulation of the Euler equations, cf. [7, 19].

The flow enters the domain from the left hand side with free-stream conditions $\rho_\infty = 1.2929$ [kg/m^3], $p_\infty = 101325$ [Pa], $v_\infty = (165.619, 0)$ [m/s]. The reference time is determined by $t_{ref} = 1./\sqrt{p_\infty/\rho_\infty} = 279.947$ [m/s]. At the boundaries we impose slip conditions at the lower boundary and characteristic boundary conditions elsewhere because of the subsonic free-stream conditions ($M_\infty = 0.5$). The grid is adapted after every timestep. The maximum refinement level is $L_{max} = 5$, the threshold $\varepsilon = 0.002$, the coarsest grid consists of 1375 cells. After two cycles of the boundary oscillation the number of grid cells varies around 40.000 grid points depending on the phase of the boundary movement.

The bump is moving periodically up and down. When the bump is moving upwards a shock occurs at the leeward side because of the acceleration of the flow. The shock weakens and moves in upstream direction when the bump moves downward. This can be deduced from Figure 9 where the Mach number at the midpoint of the bump is plotted versus the dimensionless time t/t_{ref}. When the shock is passing a steep gradient can be seen.

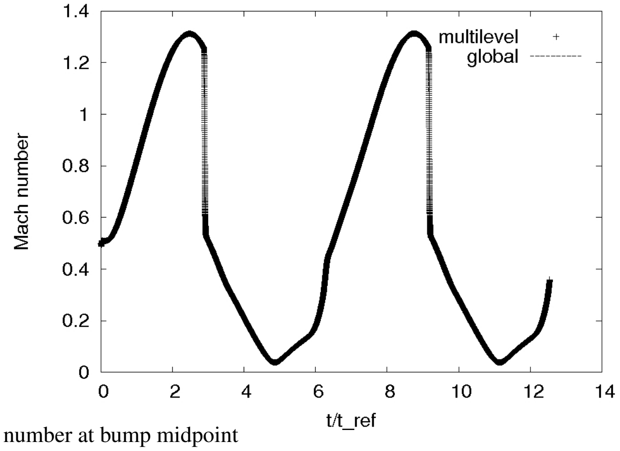


Fig. 9 Mach number at bump midpoint

The computation was carried out with both the global and the multilevel time stepping strategy. Although we performed no grid deformation step for the intermediate time levels in the latter case the accuracy of the solution is not affected as can be concluded from Figure 9. On the other hand, in comparison to a global time stepping strategy, we gain a factor of 3.7 in efficiency.

9.2 FAS-like multilevel scheme: Bump

Results of a 2D Euler transonic flow, considered in [26], are presented next in order to illustrate the convergence and efficiency of the multilevel strategy. The computational domain is defined by a circular arc bump in a channel with a secant of length $l = 1[\mathrm{m}]$ and a thickness of $h = 0.024[\mathrm{m}]$, see Figure 10. At the inlet boundary, the Mach number is 0.85 and a homogeneous flow field characterized by the free-stream quantities is imposed. At the outlet boundary, characteristic boundary conditions are used. We apply slip boundary conditions across the solid wall.

Again, the multiscale analysis employs biorthogonal wavelets with the order $M = 3$ of vanishing moments. The threshold value in the grid adaptation step is $\varepsilon = 2.5 \times 10^{-3}$ and $L = 5$. Since we are dealing with a steady state problem, the time stepsize is determined locally for each cell by a time-dependent CFL number. For the local flux computation we use the naive strategy (19). In each time step, we perform one FAS cycle to approximate the solution of the nonlinear problem.

Our computation started on a structured grid corresponding to refinement level 1 that is determined by uniformly refining once the cells of the coarsest resolution of 24×8 cells, which span the entire computational domain, and we run to steady state. Additional refinement levels are added in response to time residual dropping or after a fixed number of time steps. In the present computation we enforced grid adaptation

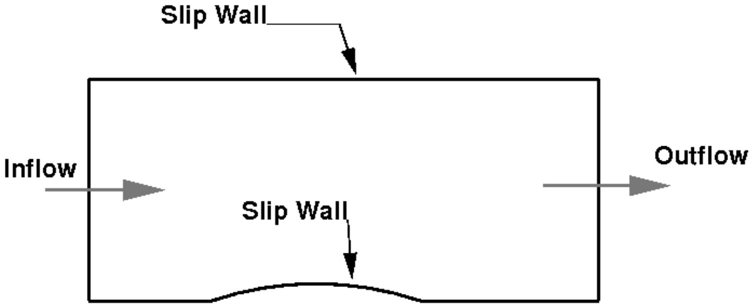


Fig. 10 Circular arc bump configuration.

after 20 iterations for the first five adaptation steps. Then additional adaptations are performed as soon as the averaged residual of the density has dropped by a factor of 10^{-5}.

Figure 13 (right) shows the computed pressure distributions after each adaptation step. At Mach 0.85 there is a compression shock separating a supersonic and a subsonic domain. The shock wave is sharply captured and the stagnation areas are highly resolved as can be concluded from the adaptive meshes shown in Figure 13 (left).

Figure 11 shows the corresponding convergence history of the computation. The measure of convergence to steady state is the averaged residual for the density, i.e.,

$$\sum_{\lambda \in G_{L,\varepsilon}^{n+1}} \frac{|V_\lambda|}{|\Omega|}(\rho_\lambda^{n+1} - \rho_\lambda^n).$$

At the beginning the residual oscillates and decreases almost monotonically between two adaptation steps. After each grid adaptation it increases by several orders of magnitudes. This is caused by the thresholding that is performed within the multiscale analysis. After the 5th adaptation, the flow pattern is already established and the residual decreases more strongly. In total, the residual was reduced from 10^{-4} to 10^{-16}.

For steady flows, the CFL number is controlled and varied between a minimum and maximum value during the computation. In the presence of shock waves, it is not possible to start the computation with a large CFL number directly due to the instationary behavior of the shock development. In the present work, computations were initiated with $CFL_{min} = 10$ and increased after each time step by a constant factor $\beta = 1.05$ until a maximum CFL_{max} is reached, i.e.,

$$CFL(t^n) = \beta\, CFL(t^{n-1}).$$

Figure 11(right) shows the history of the CFL number.

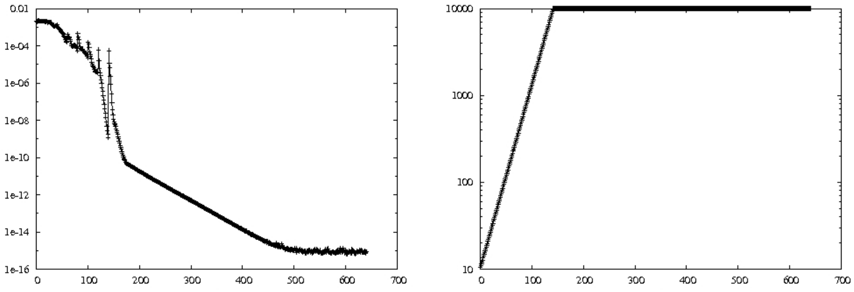


Fig. 11 Temporal variation of the residual (left) and CFL number.

9.3 Local versus exact flux and source reconstruction: Burger's equation

In order to investigate the performance of the different flux and source reconstruction strategies discussed in Section 6, we conduct some parameter studies for a simplified 1D configuration. For this purpose, we consider the inhomogeneous, inviscid Burgers' equation with flux $f(u) = 0.5u^2$, source $s(u) = u(u-0.5)(u-1)$ and initial data $u_0(x) = \sin(2\pi x)$.

The computational domain $\Omega = [0,1]$ is discretized by $N_0 = 10$ cells on the coarsest level, i.e., $h_0 = 0.1$. Hence the resolution for higher refinement levels is $N_l = 2^l N_0$ and $h_l = 2^{-l} h_0$. At the boundaries we use periodic boundary conditions. For the time discretization, we have to respect the CFL condition. Here we choose $\tau_0 = 0.016$. The final integration time is $T = 0.24$. Since we use global time stepping, the CFL condition has to hold for the smallest cells corresponding to the highest refinement level L, i.e., $\tau = 2^{-L}\tau_0$.

The explicit reference FVS (2) is determined by the Godunov flux. In order to improve spatial and temporal accuracy, we employ a piecewise linear ENO reconstruction, cf. [16]. For the source term, we apply the first order approximation (3).

Computations have been carried out for several threshold values ε and different flux and source reconstruction strategies: (i) exact reconstruction strategy according to (17) and (18), (ii) flux and source computation on unstructured meshes using only local data corresponding to the adaptive grid according to (19) as is frequently used in applications, cf. [7], and (iii) approximate reconstruction strategy using the midpoint rule and reconstruction polynomials of degree $N = 2$ with central stencil that are detailled in [17]. In the following, these are referred to as the original, naive and modified adaptive MR-FVS.

The solution develops a shock at time $t = 1/\pi$ at position $x = 0.5$ which is moving at negative speed due to the inhomogeneity. In Figure 14 we present the solution for the modified adaptive scheme for $L = 10$, $\varepsilon = 10^{-3}$ by points at the cell center of the adaptive grid and the *exact* solution computed by the reference scheme on a uniform grid corresponding to $L = 14$.

To investigate the influence of the different flux and source reconstruction strategies on the efficiency of the adaptive schemes, we have to consider the computational effort (memory and CPU time) and the accuracy (discretization and perturbation error) for various different threshold values. All adaptive computations are performed with $L = 10$ refinement levels.

According to the ideal strategy in Section 5, the threshold value ε has to be chosen such that the discretization error $\tau_L = \hat{u}_L - v_L$ of the reference scheme and the perturbation error $e_L = v_L - \overline{v}_L$ are balanced. For $L = 10$ we obtain $\|\tau_L\| = 5.8 \times 10^{-4}$ where the "exact " solution is obtained by the FVS on a uniform mesh corresponding to $L = 14$ refinement levels.

First we consider the perturbation error due to thresholding plotted in Figure 17 for various threshold parameters. Obviously, the perturbation error is decreasing with smaller threshold values. In particular, $\|e_L\| \to 0$ for $\varepsilon \to 0^+$, i.e., the modified adaptive scheme converges to the reference solution obtained on the reference grid with L refinement levels. Of course, we do not gain in accuracy when choosing a very small threshold value because the discretization error is fixed by the number of refinement levels.

To determine the optimal threshold value, we plot the error $\|\hat{u}_L - \overline{v}_L\|$ of the adaptive scheme for different threshold values, see Figure 16. From this, we conclude that an optimal choice would be $\varepsilon_{opt} \in [10^{-5}, 10^{-4}]$ because the error of the adaptive schemes is decreasing with decreasing threshold value ε as long as $\varepsilon > \varepsilon_{opt}$ whereas it stalls for $\varepsilon < \varepsilon_{opt}$. Hence, for $\varepsilon > \varepsilon_{opt}$ the perturbation error due to thresholding dominates whereas for $\varepsilon < \varepsilon_{opt}$ the discretization error dominates.

The above observations concerning the discretization and perturbation error hold true independently of the adaptive scheme. However, for a threshold value ε_{opt} in the optimal range we see in Figures 17 and 16 that the highest accuracy is obtained with the original adaptive scheme. The modified adaptive scheme looses a bit in accuracy, but for the naive adaptive scheme the loss is much more severe.

To draw any conclusions concerning the efficiency of the different adaptive schemes, we have to take the computational cost into account. First we discuss the size of the adaptive grids that determine the memory requirements, see Figure 18. We observe that the minimal grid size is usually obtained for the original adaptive scheme whereas for the naive and the modified adaptive scheme we need more cells. This might be caused by small oscillations induced by the reconstruction error. This becomes more severe in the case of the naive adaptive scheme when the threshold value is chosen too small, i.e., $\varepsilon < \varepsilon_{opt}$.

Finally, we consider in Figure 19 the computational time. The CPU time needed for the original adaptive scheme is *much* higher as long as the threshold value is not too small. This is caused by the source term computation on the uniform reference grid dominating the overall cost for grid adaptation and time evolution. In case of the naive and the modified adaptive scheme, the adaptive grid becomes more dense with decreasing threshold values, i.e., more cells are refined, and the cost approaches the cost of the reference computation on the reference grid. Of course, this behavior is expected for any adaptive scheme.

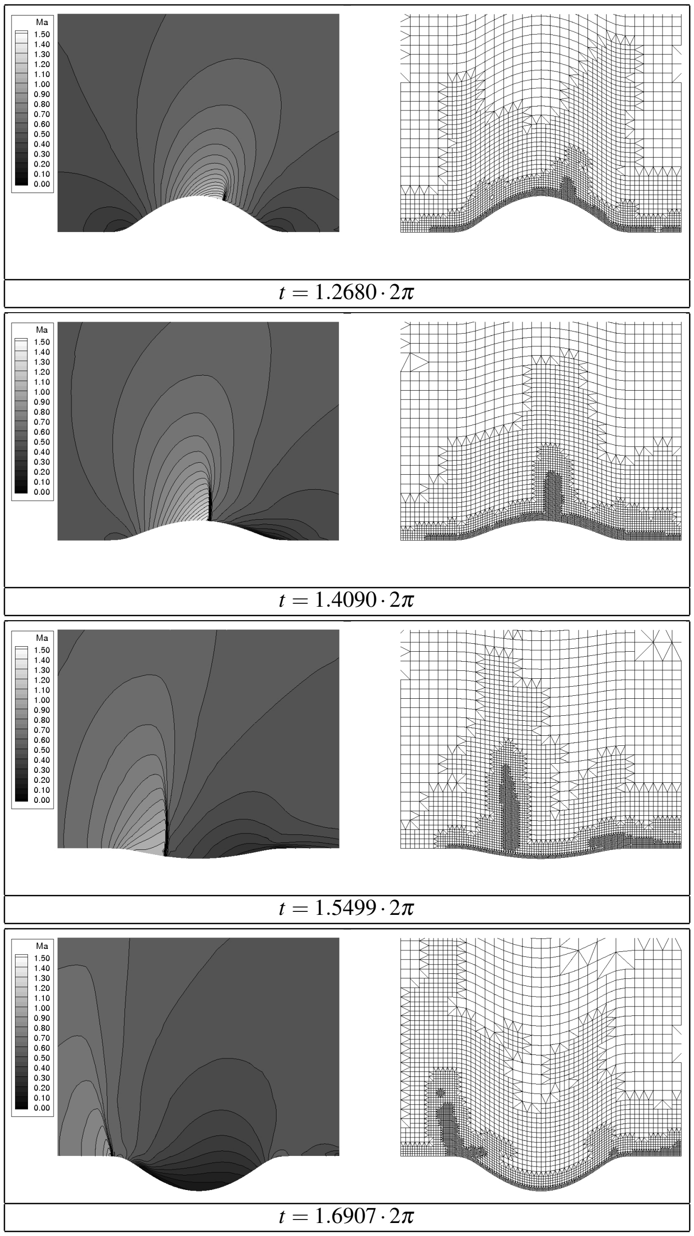
To summarize the above observations, we conclude that for an optimal threshold value ε_{opt} the exact strategy is most accurate but at the cost of the reference computation, i.e., there is no gain at all. For the naive adaptive scheme, we observe a severe loss in accuracy at lower computational cost in comparison to the the modified adaptive scheme. This loss can only be compensated by a smaller threshold value at higher computational cost. From this point of view, the approximate strategy is more efficient when fixing the target accuracy by the discretization error, i.e., $\log(\|\tau_L\|) = -3.24$, see Figure 15.

Finally, we wish to point out that in practice the optimal threshold value ε_{opt} can only be roughly estimated and, hence, the use of the local strategy cannot be recommended: we either (i) loose significant accuracy if $\varepsilon \gg \varepsilon_{opt}$, see Figures 16, or (ii) the computational cost (memory) is significantly higher due to instabilities triggered by the increasing influence of the reconstruction error if $\varepsilon \ll \varepsilon_{opt}$, see Figure 18.

References

1. S. Balay, K. Buschelmann, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhang. Petsc web page, http://www.mcs.anl.gov/petsc. Technical report, 2001.
2. S. Balay, K. Buschelmann, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhang. Petsc users manual. Technical report anl-95/11 - revision 2.1.5, Argonne National Laboratory, 2004.
3. S. Balay, W.D. Gropp, L.C. McInnes, and B.F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A.M. Bruaset, and H.P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.
4. M.J. Berger and R.J. LeVeque. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM J. Numer. Anal.*, 35(6):2298–2316, 1998.
5. M.J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comp. Physics*, 53:484–512, 1984.
6. F. Bramkamp, B. Gottschlich-Müller, M. Hesse, Ph. Lamby, S. Müller, J. Ballmann, K.-H. Brakhage, and W. Dahmen. *H*-adaptive Multiscale Schemes for the Compressible Navier–Stokes Equations — Polyhedral Discretization, Data Compression and Mesh Generation. In J. Ballmann, editor, *Flow Modulation and Fluid-Structure-Interaction at Airplane Wings*, volume 84 of *Numerical Notes on Fluid Mechanics*, pages 125–204. Springer, 2003.
7. F. Bramkamp, Ph. Lamby, and S. Müller. An adaptive multiscale finite volume solver for unsteady an steady state flow computations. *J. Comp. Phys.*, 197(2):460–490, 2004.
8. A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, 31:333–390, 1977.
9. A. Brandt. Multi-level adaptive techniques (mlat) for partial differential equations: Ideas and software. In *Mathematical software III, Proc. Symp., Madison 1977*, pages 277–318, 1977.
10. J.M. Carnicer, W. Dahmen, and J.M. Peña. Local decomposition of refinable spaces and wavelets. *Appl. Comput. Harmon. Anal.*, 3:127–153, 1996.
11. A. Cohen, S.M. Kaber, S. Müller, and M. Postel. Fully Adaptive Multiresolution Finite Volume Schemes for Conservation Laws. *Math. Comp.*, 72(241):183–225, 2003.
12. W. Dahmen, M. Behr, A. Reusken, Ch.Ĥohn J.B̃allmann, S. Mogosan, K. Brix, and S. Müller. Parallel and adaptive methods for fluid-structure-interactions. Numerical Notes on Fluid Mechanics, 2009. submitted.
13. L. Hart, S.F. McCormick, A. O'Gallagher, and J. Thomas. The fast adaptive composite-grid method (FAC): Algorithms for advanced computers. *Appl. Math. Comput.*, 19:103–125, 1986.

14. A. Harten. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Comm. Pure Appl. Math.*, 48(12):1305–1342, 1995.
15. A. Harten. Multiresolution representation of data: A general framework. *SIAM J. Numer. Anal.*, 33(3):1205–1256, 1996.
16. A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy. Uniformly high order accurate essentially non–oscillatory schemes III. *J. Comp. Phys.*, 71:231–303, 1987.
17. N. Hovhannisyan and S. Müller. On the stability of fully adaptive multiscale schemes for conservation laws using approximate flux and source reconstruction strategies. IGPM–Report 284, RWTH Aachen, 2008.
18. J.-C. Jouhaud, M. Montagnac, and L. Tourrette. A multigrid adaptive mesh refinement strategy for 3d aerodynamic design. *Int. Journal for Numerical Methods in Fluids*, 47:367–385, 2005.
19. Ph. Lamby, R. Massjung, S. Müller, and Y. Stiriba. Inviscid flow on moving grids with multiscale space and time adaptivity. In *Numerical Mathematics and Advanced Applications: Proceedings of Enumath 2005 the 6th European Conference on Numerical Mathematics and Advanced Mathematics*, pages 755–764. Springer, 2006.
20. Ph. Lamby, S. Müller, and Y. Stiriba. Solution of shallow water equations using fully adaptive multiscale schemes. *Int. Journal for Numerical Methods in Fluids*, 49(4):417–437, 2005.
21. S.F. McCormick. *Multilevel adaptive methods for partial differential equations*, volume 6 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia, 1989.
22. S. Müller. Adaptive multiresolution schemes. In B. Herbin and D. Kröner, editors, *Finite Volumes for Complex Applications*. Hermes Science, Paris, 2002.
23. S. Müller and Y. Stiriba. Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping. *Journal for Scientific Computing*, 30(3):493–531, 2007.
24. S. Müller and Y. Stiriba. A multilevel finite volume method with multiscale-based grid adaptation for steady compressible flow. *Journal of Computational and Applied Mathematics*, 2008. DOI 10.1016/j.cam.2008.03.035.
25. S. Osher and R. Sanders. Numerical approximations to nonlinear conservation laws with locally varying time and space grids. *Math. Comp.*, 41:321–336, 1983.
26. A. Rizzi and H. Viviand. *Numerical methods for the computation of inviscid transonic flows with socck waves*, volume 3 of *Notes on Numerical Fluid Mechanics*. Vieweg, Braunschweig, 1981.
27. Ch. Steiner. *Adaptive timestepping for conservation laws via adjoint error representation*. PhD thesis, RWTH Aachen, 2008.
28. Ch. Steiner, S. Müller, and S. Noelle. Adaptive timestep control for weakly instationary solutions of the Euler equations. IGPM–Report 292, RWTH Aachen, 2009.
29. Ch. Steiner and S. Noelle. On adaptive timestepping for weakly instationary solutions of hyperbolic conservation laws via adjoint error control. *Communications in Numerical Methods in Engineering*, 2008. accepted for publication.
30. U. Trottenberg, C.W. Oosterlee, and A. Schüller. *Multigrid. With guest contributions by A. Brandt, P. Oswald, K. Stüben*. Academic Press, Orlando, 2001.
31. G. Zumbusch. *Parallel multilevel methods. Adaptive mesh refinement and loadbalancing*. Advances in Numerical Mathematics. Teubner, Wiesbaden, 2003.



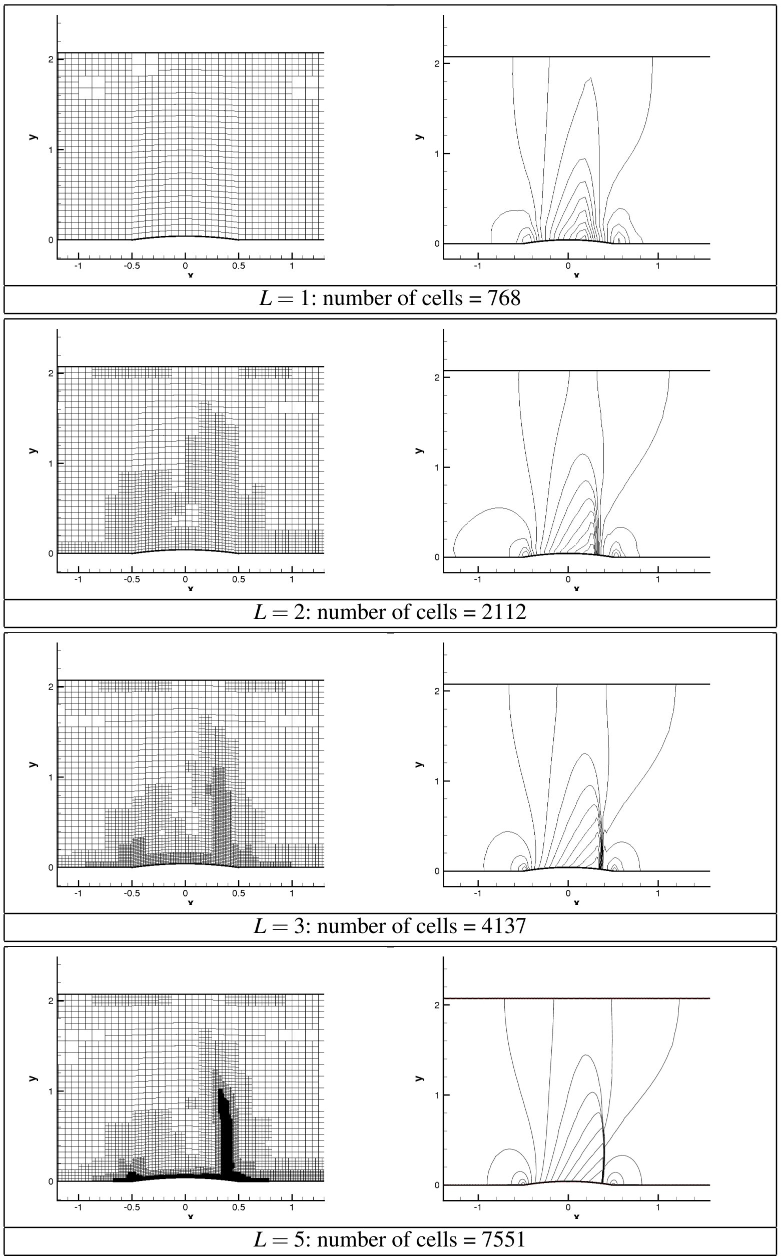
$t = 1.2680 \cdot 2\pi$

$t = 1.4090 \cdot 2\pi$

$t = 1.5499 \cdot 2\pi$

$t = 1.6907 \cdot 2\pi$

Fig. 12 Time evolution of Mach number distribution and adaptive grid for flow over moving bump (multilevel time stepping, $L = 5$)



$L = 1$: number of cells = 768

$L = 2$: number of cells = 2112

$L = 3$: number of cells = 4137

$L = 5$: number of cells = 7551

Fig. 13 Adaptive grid (left) and pressure contours over the bump (right) after each adaptation

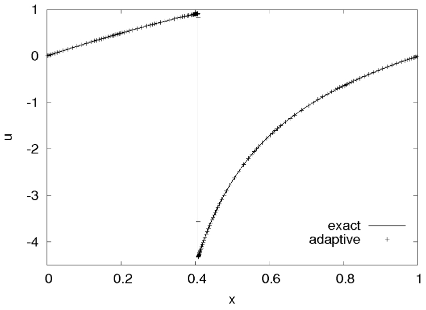


Fig. 14 Comparison of adaptive solution ($L = 10$, $\varepsilon = 10^{-3}$) and *exact* solution ($L = 14$, $\varepsilon = 0$)

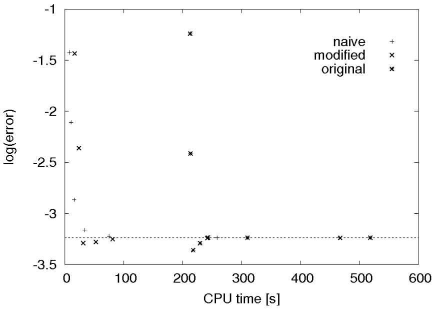


Fig. 15 Comparison of CPU time and error of adaptive scheme for different threshold values.

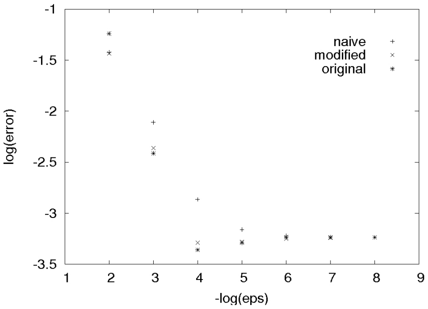


Fig. 16 Error of adaptive solution with $L = 10$ and varying threshold value ε.

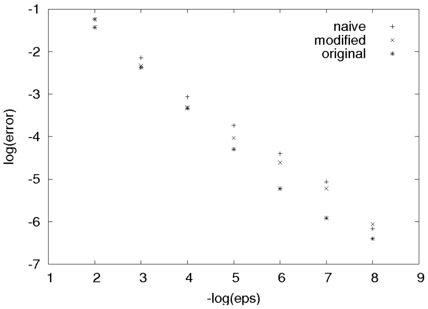


Fig. 17 Perturbation error of adaptive solution ($L = 10$, varying threshold value ε) and reference solution ($L = 10$, $\varepsilon = 0$) on reference grid ($L = 10$)

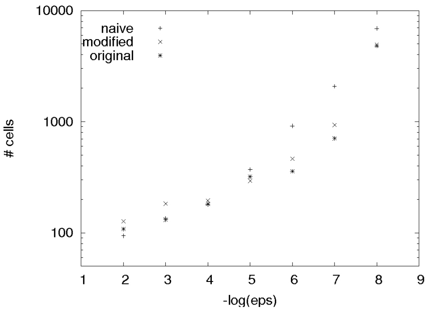


Fig. 18 Number of cells: Adaptive computations with $L = 10$ and varying threshold value ε.

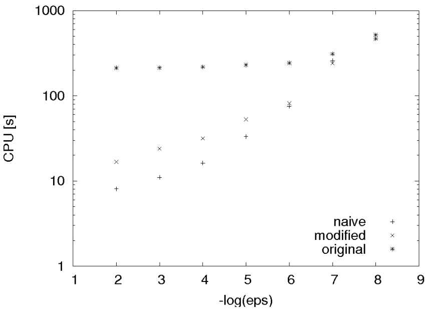


Fig. 19 Computational time: Adaptive computations with $L = 10$ and varying threshold value ε.