A MULTILEVEL FINITE VOLUME METHOD WITH MULTISCALE-BASED GRID ADAPTATION FOR STEADY COMPRESSIBLE FLOWS

S. Müller* and Y. Stiriba†

* University of Technology RWTH Aachen, Institut für Geometrie und Praktische Mathematik
Templergraben 55, D-52056 Aachen, Germany
e-mail: mueller@igpm.rwth-aachen.de

† Universitat Rovira i Virgili, ETSEQ-DEM,
Av. Països Catalans 26, 43007 Tarragona, Spain
e-mail: youssef.stiriba@urv.cat

Key words: Finite volume method, multiresolution analysis, grid adaptation, steady state computations

Abstract. An implicit multilevel finite volume solver on adaptively refined quadtree meshes is presented for the solution of steady state flow problems. The nonlinear problem arising from the implicit time discretization is solved by an adaptive FAS multigrid method. Local grid adaptation is performed by means of a multiscale-based strategy. For this purpose data of the flow field are decomposed in coarse grid information and a sequence of detail coefficients that describe the difference between two refinement levels and reveal insight in the local regularity behavior of the solution. Here wavelet techniques are employed for the multiscale analysis. The key idea of the present work is to use the transfer operators of the multiscale analysis for the prolongation and restriction operator in the FAS cycle. The efficiency of the solver is investigated by means of an inviscid 2D flow over a bump.

1 INTRODUCTION

The numerical solution of compressible flow equations by finite volume methods requires a highly resolved mesh to simulate accurately the different scales of the flow field and its boundaries. For multidimensional problems, in particular, uniform grids are not feasible. However, due to the inhomogeneity of the flow field, a grid which is sufficiently fine near high gradient regions may cause unnecessary refinement in other flow regions. Therefore, adaptive grid methods can significantly improve the efficiency by concentrating cells only where they are most required, while reducing storage requirements as well as the computational time.
For this purpose, numerical schemes have been discussed or are under current investigation that aim at adapting the spatial grid to the local behavior of the flow field. In the early 90's Harten [11] proposed to use multiresolution techniques in the context of finite volume schemes applied to hyperbolic conservation laws. He employed these techniques 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.

In Harten's original approach the multiresolution analysis is used to control a hybrid flux computation by which computational time for the flux computation can be saved whereas the overall computational complexity is not reduced but still stays proportional to the number of cells on the uniformly fine reference mesh. Opposite to this strategy, threshold techniques are applied to the multiresolution decomposition in [18, 9] where detail coefficients below a threshold value are discarded. By means of the remaining significant details a locally refined mesh is determined whose complexity is significantly reduced in comparison to the underlying reference mesh.

The multiresolution-based grid adaptation technique can be used as a black box only, cf. [5]. However, the multiresolution analysis provides even more potential when it is applied directly to the discrete evolution equations arising from the finite volume discretization and not alone to the set of discrete cell data. For instance, it was recently used to develop a multilevel time stepping strategy where refinement is also performed in time, cf. [16].

In the present work, we are interested to combine the multiscale-based grid adaptation with multigrid techniques to solve efficiently the nonlinear problems arising from the implicit time discretization of the underlying finite volume scheme. First work on adaptive multigrid techniques has been reported by Brandt [6, 7] 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) [10, 17] can be regarded as an alternative to the MLAT approach. An overview on multigrid methods can be found in the review book [21].

Opposite to classical adaptive multigrid schemes we employ the multiresolution analysis using biorthogonal wavelets to define the restriction and prolongation operators. Since the underlying problem is nonlinear we choose the FAS [6] for the coarse grid correction.

Note that similar investigations have been published recently in [14] 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.

The paper is organized as follows. We start with a brief description of the fluid equations and their discretization by implicit finite volume schemes, see Section 2. To improve
the efficiency of the scheme without loss of accuracy we employ multiresolution techniques. For this purpose, we recall the basic ideas of the underlying multiscale concept, see Section 3. By means of the multiscale analysis we then describe how to construct locally refined grid, see Section 4. The nonlinear problems arising from the implicit time discretization are finally solved by an FAS multigrid method where we employ the linear transfer operators from the multiscale analysis for prolongation and restriction, see Section 5. In Section 6, results of a 2D Euler transonic flow around a circular arc bump in a channel are presented to illustrate the efficiency of the scheme.

2 GOVERNING EQUATIONS AND FINITE VOLUME SCHEME

For the numerical simulation of steady state inviscid compressible fluids we solve the time-dependent 2D Euler equations. Here the time can be regarded as an iteration parameter rather than the physical time. These lead to a system of conservation equations

\[
\frac{\partial}{\partial t} \int_V \tilde{U} \, dV + \int_{\partial V} \tilde{f} \cdot \tilde{n} \, dS = 0.
\]

Here, \( \tilde{U} = (\rho, \rho \tilde{v}, \rho E)^T \) is the array of the mean conserved quantities: density of mass, momentum, specific total energy and \( \tilde{f} = (\rho \tilde{v}, \rho \tilde{v} \cdot \tilde{v} + p \tilde{v}, \tilde{v} (\rho E + p))^T \) the array of the corresponding convective fluxes. \( p \) is the pressure and \( \tilde{v} \) the fluid velocity. The system of equations is closed by the perfect gas equation of state \( p = \rho (E - 0.5 \tilde{v}^2)(\gamma - 1) \) with \( \gamma = 1.4 \) (air).

The equations (1) are approximated by a finite volume scheme where the time is discretized implicitly using the backward Euler scheme. For this purpose the finite fluid domain \( \Omega \subset \mathbb{R}^d \) 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 gives \( \Omega \). 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 \( \tilde{n}_{ij} \) the outer normal of \( \Gamma_{ij} \) corresponding to cell \( i \). To improve convergence to steady state we use local time stepping. Hence the time interval might be discretized differently for each cell, i.e., \( t_i^{n+1} = t_i^n + \tau_i^{n+1} \). On this particular discretization the implicit finite volume scheme can be written as

\[
v_i^{n+1} + \frac{\tau_i^{n+1}}{|V_i^{n+1}|} B_i^{n+1} = v_i^n \quad \text{with} \quad B_i^{n+1} := \sum_{j \in N(i)} |\Gamma_{ij}| F(v_{ij}^{n+1}, v_{ji}^{n+1}, \tilde{n}_{ij}) \quad (2)
\]

to compute the approximated cell averages \( v_i^{n+1} \) of the conserved variables on the new time level. Note that the time levels are not synchronized. Here the numerical flux function \( F(\tilde{u}, \tilde{v}, \tilde{n}) \) is an approximation for the flux \( f(\tilde{u}, \tilde{n}) := \tilde{f} \cdot \tilde{n} \) in outer normal direction \( \tilde{n}_{ij} \) on the edge \( \Gamma_{ij} \). The numerical flux is assumed to be consistent, i.e., \( F(\tilde{u}, \tilde{u}, \tilde{n}) = f(\tilde{u}, \tilde{n}) \). For simplicity of presentation we neglect 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 \( \sum_{j \in N(i)} |\Gamma_{ij}| \tilde{n}_{ij} = \tilde{0} \) holds.
This scheme is to be accelerated using multiscale techniques. First of all we reduce the number of discrete evolution equations in (2) by employing local grid adaptation. Then the reduced nonlinear problem is solved by an FAS multigrid method where again we employ the multiscale analysis to define the grid transfer operators.

3 MULTISCALE ANALYSIS

A finite volume discretization is typically working on a sequence of cell averages. In order to analyze the local regularity behavior of the data we employ the concept of biorthogonal wavelets [8]. This approach may be seen as a natural generalization of Harten’s discrete framework [12]. The core ingredients are (i) a hierarchy of nested grids, (ii) biorthogonal wavelets and (iii) the multiscale decomposition. Here we will only summarize the basic ideas. For the realization and implementation see [18].

Grid hierarchy. Let be \( \Omega_l := \{ V_\lambda \}_{\lambda \in I_l} \) a sequence of different meshes corresponding to different resolution levels \( l \in \mathbb{N}_0 \) where the discretization length is decreasing 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 \( \mu \in \mathcal{M}_\lambda^0 \) on the next higher refinement level \( l + 1 \), i.e.,

\[
V_\lambda = \bigcup_{\mu \in \mathcal{M}_\lambda^0} V_\mu
\]

where \( \mathcal{M}_\lambda^0 \subset I_{l+1} \) is the refinement set and, hence, \( \Omega_l \subset \Omega_{l+1} \). A simple example is shown in Figure 1 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. [18].

Box function and cell averages. Relative to the partitions \( \Omega_l \) we introduce the so-called box function

\[
\tilde{\phi}_\lambda(x) := \frac{1}{|V_\lambda|} \chi_{V_\lambda}(x) = \begin{cases} 1/|V_\lambda|, & x \in V_\lambda \\ 0, & x \notin V_\lambda \end{cases}
\]

defined as the \( L^1 \)-scaled characteristic function with respect to \( V_\lambda \). 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.,

\[
\hat{u}_\lambda := \langle u, \tilde{\phi}_\lambda \rangle_\Omega \quad \text{with} \quad \langle u, v \rangle_\Omega := \int_\Omega u v \, dx.
\]
Obviously the nestedness of the grids as well as the linearity of integration imply the two-scale relations

\[
\tilde{\phi}_\lambda = \sum_{\mu \in \mathcal{M}_\lambda^0} m_{\mu,\lambda}^{l,0} \tilde{\phi}_\mu \quad \text{and} \quad \tilde{u}_\lambda = \sum_{\mu \in \mathcal{M}_\lambda^0} m_{\mu,\lambda}^{l,0} \tilde{u}_\mu
\]

where the mask coefficients turn out to be \( m_{\mu,\lambda}^{l,0} := |V_\mu|/|V_\lambda| \) for all cells \( \mu \in \mathcal{M}_\lambda^0 \) 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.,

\[
\tilde{\psi}_\lambda := \sum_{\mu \in \mathcal{M}_\lambda^1 \subset \mathcal{M}_{l+1}} m_{\mu,\lambda}^{l,1} \tilde{\phi}_\mu
\]

with mask coefficients \( m_{\mu,\lambda}^{l,1} \) that only depend on the grids. The construction of the wavelets is subject to certain constraints, namely, (i) the wavelet functions \( \tilde{\psi}_l := (\tilde{\psi}_\lambda)_{\lambda \in \mathcal{J}_l} \) build a completion of the basis system \( \tilde{\Phi}_l := (\tilde{\phi}_\lambda)_{\lambda \in \mathcal{J}_l} \), (ii) they are locally supported, (iii) provide vanishing moments and (iv) there exists a biorthogonal system \( \Phi_l \) and \( \Psi_l \) of primal functions. An example is shown in Figure 9 of Appendix A for the univariate case. For details we refer to the concept of stable completions, see [8]. Then we can perform a change of basis between \( \tilde{\Phi}_l \cup \tilde{\Psi}_l \) and \( \tilde{\Phi}_{l+1} \), i.e.,

\[
\tilde{\phi}_\lambda = \sum_{\mu \in \mathcal{M}_\lambda^1 \subset \mathcal{J}_l} g_{\mu,\lambda}^{l,0} \tilde{\phi}_\mu + \sum_{\mu \in \mathcal{M}_\lambda^1 \subset \mathcal{J}_l} g_{\mu,\lambda}^{l,1} \tilde{\psi}_\mu
\]

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 \( g_{\mu,\lambda}^{l,0} \) and \( g_{\mu,\lambda}^{l,1} \) depend only on the grid geometry. By means of the wavelet functions we introduce the detail coefficients

\[
d_\lambda := \langle u, \tilde{\psi}_\lambda \rangle_\Omega.
\]

These coefficients inherit the two-scale relation

\[
d_\lambda = \sum_{\mu \in \mathcal{M}_\lambda^1} m_{\mu,\lambda}^{l,1} \tilde{u}_\mu
\]

from its functional counterpart (7).

**Multiscale Transformation.** The ultimate goal is to transform the array of cell averages \( u_L := (\hat{u}_{L,k})_{k \in \mathcal{J}_L} \) corresponding to a finest uniform discretization level into a sequence of coarse grid data \( u_0 := (\hat{u}_k)_{k \in \mathcal{J}_0} \) and details \( d_l := (d_\lambda)_{\lambda \in \mathcal{J}_l}, \ l = 0, \ldots, L - 1, \) representing the successive update from a coarse resolution to a high resolution. According
to (6) and (7) we obtain two–scale relations for the coefficients inherited from the two–scale relations of the box functions and the wavelet functions

\[ \hat{u}_\lambda = \sum_{\mu \in \mathcal{M}_\lambda^0} m_{\mu,\lambda}^{l,0} \hat{u}_\mu, \quad d_\lambda = \sum_{\mu \in \mathcal{M}_\lambda^1} m_{\mu,\lambda}^{l,1} \hat{u}_\mu \]  

and

\[ \hat{u}_\lambda = \sum_{\mu \in \mathcal{G}_\lambda^0} d_{\mu,\lambda}^{l,0} \hat{u}_\mu + \sum_{\mu \in \mathcal{G}_\lambda^1} d_{\mu,\lambda}^{l,1} d_\mu. \]  

Applying the relations (11) iteratively, see Figure 2, the array \( \hat{u}_L \) is successively decomposed. We refer to this transformation as multiscale transformation. It is reversed by the inverse multiscale transformation (12).

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

\[ |d_\lambda| \leq C 2^{l} \| u^{(M)} \|_{L^\infty(V_\lambda)}. \]  

Obviously, the details decay with a rate at least of \( 2^{-lM} \) provided the function \( u \) is differentiable and the wavelets have \( M \) vanishing moments, 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 \( 2^{-l} \).

This motivates to neglect all sufficiently small details in order to compress the original data.

4 MULTISCALE-BASED SPATIAL GRID ADAPTATION

To determine a locally refined grid we now employ the above multiscale decomposition. The basic idea is to perform data compression on the vector of detail coefficients using hard thresholding as is motivated by the cancellation property. This will significantly reduce the complexity of the data. By means of the thresholded sequence we then perform local grid adaptation where we refine a cell whenever there exists a significant detail. The main steps in this procedure are summarized in the following:

Step 1: Multiscale analysis. Let be \( v^u_L \) 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 \( d_i^n \), \( l = 0, \ldots, L - 1 \) of ascending resolution, see Figure 2, and cell averages on some coarsest level \( l = 0 \). For this purpose we perform locally the multiscale transformation (11).

Step 2: Thresholding. In order to compress the original data we discard all detail coefficients \( d_\lambda \) whose absolute values fall below a level-dependent threshold value \( \epsilon_l = 2^{l-L} \epsilon \). Let \( D_{L,\epsilon} \) be the set of significant details. The ideal strategy would be to determine
the threshold value \( \epsilon \) 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, see [9].

**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 \( D_{L, \epsilon}^{n+1} \supset D_{L, \epsilon}^n \cup D_{L, \epsilon}^{n+1} \) be the prediction set. The prediction strategy is detailed in [9]. In view of the grid adaptation step this set is additionally inflated such that it corresponds to a graded tree.

**Step 4: Grid adaptation.** By means of the set \( D_{L, \epsilon}^{n+1} \) a locally refined grid is determined. For this purpose, we recursively check proceeding levelwise from coarse to fine whether there exists a significant detail to a cell. If there is one, then we refine the respective cell. We finally obtain the locally refined grid with hanging nodes represented by the index set \( G_{L, \epsilon}^{n+1} \). The data on the new grid can be computed in the same loop where we apply locally the inverse multiscale transformation (12).

![Figure 3: Grid adaptation: refinement tree (left) and corresponding adaptive grid (right)](image)

## 5 FAS MULTIGRID SCHEME

In order to solve the nonlinear problem arising from the implicit time discretization on locally refined grids in one evolution step we combine the FAS strategy [6] 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 \( G_l \subset I_l, \ l = 0, \ldots, L \) 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 $\mathcal{D}_l$, $l = 0, \ldots, L - 1$ and $\mathcal{D}_L := \emptyset$. Then the \textit{adaptive grid} $\mathcal{G}$ is the union $\mathcal{G} = \bigcup_{l=0}^{L} \mathcal{G}_l$ of the index sets $\mathcal{G}_l$, $l = 0, \ldots, L$. Furthermore the \textit{composite grid} $\mathcal{T}$ is composed of all cells in the adaptive grid \textit{and} the cells characterized by significant details, i.e., it is the union $\mathcal{T} := \bigcup_{l=0}^{L} \mathcal{T}_l$ of the composite index sets $\mathcal{T}_l := \mathcal{G}_l \cup \mathcal{D}_l$ on level $l = 0, \ldots, L$ with $\mathcal{G}_l \cap \mathcal{D}_l = \emptyset$. The above grids and index sets, respectively, can be interpreted by a graded tree where the adaptive grid $\mathcal{G}$ corresponds to the leaves of this tree and the non-leaves correspond to the significant details $\mathcal{D}$. The composite grid $\mathcal{T}$ is the union of both, i.e., the tree itself. For an illustration see Figure 3 (left). Note that we suppress the time index $n$ for simplification of representation.

\textbf{Smoothing.} To smooth the data on level $l$ we perform $\mu$ Newton steps, i.e.,

$$
N_l^{i+1} = -N_l^{i} \Delta \mathbf{v}^{(i)}; \quad \mathbf{v}^{(i+1)} = \mathbf{v}^{(i)} + \Delta \mathbf{v}^{(i)}, \quad i = 0, \ldots, \mu - 1
$$

(14)

with initial data $\mathbf{v}^{(0)} = v^n_m$ given by the $m$th FAS cycle. Here the nonlinear operator $N_l$ is determined by the discrete evolution equations (2) of the implicit finite volume scheme for the data corresponding to the composite grid $\mathcal{T}_l$ on level $l$, i.e.,

$$
(N_l \mathbf{v})_{\lambda} = v_{\lambda} + \frac{\tau_l}{|V_{\lambda}|} B_{\lambda}, \quad \lambda \in \mathcal{T}_l.
$$

The linear systems is iteratively solved using GMRES with ILU(2) pre-conditioner. For this purpose we employ the PETSc software library of Argonne National Laboratory \cite{1, 2, 3}. The iteration terminates when the residual is dropping below the tolerance $\text{tol} = 1.e - 8$ or the maximum number of 100 relaxation steps is exceeded.

\textbf{Restriction.} Due to the nestedness of the underlying grid hierarchy the restriction operator $I_{l+1}^{l} : \mathcal{T}_{l+1} \rightarrow \mathcal{D}_l$ is naturally defined by

$$
v_{\lambda} = \sum_{\mu \in \mathcal{M}_{\lambda}^l} \frac{|V_{\mu}|}{|V_{\lambda}|} v_{\mu}
$$

(15)

according to (6). 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 $\mathcal{D}_l$ of significant details. Furthermore we note that by the restriction the adaptive grid $\mathcal{I}_l$ on level $l$ is inflated by the new data corresponding to $\mathcal{D}_l$. This is the \textit{composite grid} $\mathcal{T}_l$ on level $l$.

\textbf{Prolongation.} For the prolongation of data $\mu \in \mathcal{I}_{l+1}$ from level $l$ to level $l + 1$ we employ the inverse two-scale transformation (12) where we put the details to zero, i.e.,

$$
v_{\mu} = \sum_{\mu \in \mathcal{M}_{\mu}^{l+1}} g_{\lambda, \mu}^{l,0} v_{\mu}.
$$

(16)
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 $I_{l+1}^l : \mathcal{D}_l \rightarrow \mathcal{T}_{l+1}$ is only applied to cells of the composite grid $\mathcal{T}_l$ on level $l$ that are refined according to the significant details $\mathcal{D}_l$.

**Coarse grid problem.** Let assume that we have some approximation $v_l = (v_{\lambda})_{\lambda \in \mathcal{T}_l}$ and $v_{l-1} = (v_{\lambda})_{\lambda \in \mathcal{G}_{l-1}}$ on level $k = l-1, l$ and some right hand side $f_l = (f_{\lambda})_{\lambda \in \mathcal{T}_l}$. To setup 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.,

$$(N_l v_l)_{\lambda} = v_{\lambda} + \frac{\tau_l}{|V_\lambda|} B_{\lambda}, \quad \lambda \in \mathcal{T}_l.$$  

Note that for the computation of the flux balances $B_{\lambda}$ we access also to data of the adaptive grid on coarser scales. Then the defect on level $l$ is determined by

$$d_l = f_l - (N_l v_l)_{\lambda}, \quad \lambda \in \mathcal{T}_l.$$  

It is not to be confused with the detail coefficients of the multiscale decomposition.

Next we apply the restriction operator $I_{l-1}^l$ to the defect $d_l$ and to the data $v_l$, i.e.,

$$\bar{d}_{\lambda} = f_{\lambda} - (N_l v_l)_{\lambda}, \quad \lambda \in \mathcal{T}_l.$$  

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

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

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

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

$$f_{\lambda} = d_{\lambda} + (N_{l-1} v_{l-1})_{\lambda}, \quad \lambda \in \mathcal{D}_{l-1}$$  

(17)

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

$$f_{\lambda} = r^n_{\lambda}, \quad \lambda \in \mathcal{I}_{l-1}$$  

(18)

for the non-refined cells in the adaptive grid. Here $r^n_{\lambda}$ is the residual corresponding to the old time level $t^n$, i.e.,

$$r^n_{\lambda} = v^n_{\lambda}, \quad \lambda \in \mathcal{G}_{l-1}.$$  

(19)

Then the coarse grid problem is given by

$$(N_{l-1} w_{l-1})_{\lambda} = w_{\lambda} + \frac{\tau_{l-1}}{|V_\lambda|} B_{\lambda} = f_{\lambda}, \quad \lambda \in \mathcal{T}_{l-1}$$  

(20)
Adaptive FAS cycle. Finally, we may describe one iteration step \( m \rightarrow m + 1 \) of the adaptive multilevel cycle

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

by means 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 \( t^m \). From these we compute the residuals \( r^m_\lambda, \lambda \in G_k \), according to (19) that are stored in the right hand side terms \( f_k \): We start with performing \( \mu_1 \) smoothing steps (14) on the data \( v^m_l \) of level \( l \). Next we perform the coarse-grid correction. For this purpose, we first compute the defect \( \tilde{d}^m_l \) from the relaxed data \( \tilde{v}^m_l \). The defect as well as the relaxed data are restricted from \( T_l \) to \( D_{l-1} \) according to (15). 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 (17) and (18). The coarse grid problem (20) is then iteratively solved by the Newton scheme (14) or, if there are additional scales, we recursively apply the algorithm again to the coarser scale \( l - 1 \). The current solution on adaptive grid on level \( l - 1 \) is then replaced by the coarse grid solution \( \tilde{w}^m_{l-1} \) whereas for the refined cells \( D_{l-1} \) the correction \( \tilde{v}^m_{l-1} = \tilde{w}^m_{l-1} - \tilde{v}^m_{l-1} \) is computed. The latter is interpolated to \( T_l \) using (16) and the relaxed data are updated by the interpolation \( \hat{v}^m_l \). On the corrected approximation \( v^m_{l,cyc} = \tilde{v}^m_l + \hat{v}^m_l \) we again perform \( \mu_2 \) smoothing steps. The algorithm is sketched in Figure 4 and summarized in Appendix B.

![Figure 4: FAS \((l, l - 1)\) two-grid method](image)

### 6 NUMERICAL RESULTS

The scheme is to be validated by means of a transonic flow problem. For this purpose we first have to specify some details on the grid generation and the numerical flux evaluation on locally refined grids with hanging nodes.
Grid generation. The computational domain in our test configuration at hand is bounded by curvilinear boundaries. For this domain we compute a parametric grid mapping \( \tilde{x} : [0, 1]^d \rightarrow \Omega \). Then a hierarchy of Cartesian grids for the parameter domain is mapped to a grid hierarchy of curvilinear meshes in the computational domain. The grid mapping is realized efficiently by a sparse B-Spline representation, cf. [5, 15]. Then the locally refined grids are determined by evaluation of this mapping.

Numerical Flux. In order to fix the numerical flux we have to choose the Riemann solver and the reconstruction method. In our computations the underlying discretization is always a hierarchy of curvilinear grids. Therefore we employ a quasi one-dimensional second-order ENO reconstruction that is applied componentwise to the primitive variables \((\rho, \vec{v}, p)\) according to [13] to improve the spatial accuracy. In order to avoid unphysical oscillations near to discontinuities we use the limiter by Venkatakrishnan, cf. [22]. At the cell interfaces we then solve approximately a Riemann problem by the Roe Riemann solver [20]. In the smoothing steps of the FAS algorithm where we apply a fixed number of Newton steps we have to compute the Jacobian of the numerical fluxes. Here we use the approximate Roe Jacobian according to [4]. The Jacobian of the reconstruction is not taken into account.

Interfaces with hanging nodes. Due to local grid adaptation we have to deal with hanging nodes. To compute numerical fluxes at interfaces with hanging nodes in (2) we first compute the numerical fluxes on the higher scale by values on the same refinement level. If the neighboring cells correspond to higher or lower scales the data are projected to the respective level by means of the multiscale transformation (12). Then the numerical flux on the coarser scale is the sum of all fluxes on higher scale by which the coarse interface is composed. This is shown in Figure 5. This procedure is motivated by applying the

![Figure 5: Flux evaluation at interface with hanging node: adaptive grid (right), virtually refined grid (left)](image)

![Figure 6: Flux computation: fine grid (left), cancellation due to conservation (middle), coarse grid (right)](image)

multiscale decomposition to the evolution equations (2) on a uniformly refined reference grid. Then by the conservation property the fluxes corresponding to internal fluxes cancel and only the fine-grid fluxes contribute to the edges of the coarse-grid cells, see Figure 6.

Computations. Results of a 2D Euler transonic flow, considered in [19], are presented to illustrate the convergence and efficiency of the numerical method. The computational
domain is defined by a circular arc bump in a channel with a secant of length $l = 1\,[\text{m}]$ and a thickness of $h = 0.024\,[\text{m}]$, see Figure 7. 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, i.e., the normal velocity is set to zero.

![Figure 7: Circular arc bump configuration.](image)

Computations were started on a structured grid corresponding to refinement level 1 that is determined by uniformly refining once the cells of the coarsest resolution of $24 \times 8$ 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 after 20 iterations for the first five adaptation steps. Then additional adaptations are performed when the averaged residual of the density has dropped by a factor of $10^{-5}$.

The multiscale analysis is performed employing biorthogonal wavelets with $M = 3$ vanishing moments. The threshold value in the grid adaptation step is $\epsilon = 2.5 \times 10^{-3}$ and $L = 5$. In general, a small threshold value results in more grid refinement whereas a large value gives locally coarser grids, see the discussion in Section 4. In each time step, we perform one FAS cycle to approximate the solution of the nonlinear problem.

Figure 10 (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 10 (left).

Figure 8 shows the corresponding convergence history of the computation. The measure of convergence to steady state is the averaged residual for the density, i.e.,

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

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 $10^{-16}$. The total CPU time was 1647.081 s.

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 CFL number directly due to the instationary behavior of the shock development. In the present work, computations were initiated with $CFL_{\text{min}} = 10$ and increased after each time step by a constant factor $\beta = 1.05$ until a maximum $CFL_{\text{max}}$ is reached, i.e.,

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

Figure 8(right) shows the history of the CFL number.

![CFL history](image)

7 CONCLUSIONS

In this work, a new implicit multilevel finite volume solver on adaptively refined quadtree meshes has been formulated and implemented for the solution of steady state compressible flow problems. This approach is aiming at unified grid transfer operators for the grid adaptation and the FAS cycle. The result shows that the adaptive multigrid method is an efficient tool of accelerating convergence to steady Euler solutions. Reduction of computational costs in terms of both memory storages and the total CPU time are considerably achieved.

In the future, we will perform comparisons using classical transfer operators in the FAS cycle as well as approximating the linear system in the Newton step by classical relaxation schemes. Further developments include extensions to three dimensions and the Navier-Stokes systems.
8 ACKNOWLEDGMENT

This work has been performed with funding by the Deutsche Forschungsgemeinschaft in the Collaborative Research Center SFB 401 ”Flow Modulation and Fluid-Structure Interaction at Airplane Wings” of the RWTH Aachen, Germany and the ”Ramón y Cajal” program of the Ministerio de Educacion y Ciencia, Spain. The second author also thanks the DAAD ”Deutscher Akademischer Austausch Dienst” for partial support.

REFERENCES


S. Müller and Y. Stiriba


A  SCALING FUNCTIONS AND WAVELETS

Figure 9: Biorthogonal wavelets and scaling functions for $M = 3$
B ADAPTIVE FAS CYCLE

(1) Presmoothing:
- Compute $\mathbf{v}_l^m$ by applying $\mu_1$ smoothing steps to $\mathbf{v}_l^m$ on $\mathcal{T}_l$, i.e.,
  \[
  \mathbf{v}_l^m = \text{SMOOTH}^{\mu_1}(l, \mathbf{v}_l^m, N_l, f_l)
  \]

(2) Coarse grid correction:
- Compute defect on $\mathcal{T}_l$:
  \[
  d_l^m = f_l - N_l \mathbf{v}_l^m
  \]
- Restrict defect from $\mathcal{T}_l$:
  \[
  d_{l-1}^m = I_{l-1}^{-1} d_l^m
  \]
- Compute data on $\mathcal{T}_{l-1}$:
  \[
  \mathbf{v}_{l-1}^m = \begin{cases}
  I_{l-1}^{-1} \mathbf{v}_l^m, & \text{on } D_{l-1} \\
  \mathbf{v}_l^m, & \text{on } G_{l-1}
  \end{cases}
  \]
- Compute right hand side on $\mathcal{T}_{l-1}$:
  \[
  f_{l-1} = \begin{cases}
  I_{l-1}^{-1} d_l^m + N_{l-1} \mathbf{v}_{l-1}^m, & \text{on } D_{l-1} \\
  r_{l-1}^m, & \text{on } G_{l-1}
  \end{cases}
  \]
- Compute an approximate solution $\hat{w}_{l-1}^m$ of the coarse grid problem (20) on $\mathcal{T}_{l-1}$:
  \[
  \hat{w}_{l-1}^m = \begin{cases}
  \text{NEWTON}(l - 1, \mathbf{v}_{l-1}^m, N_{l-1}, f_{l-1}, \mu_1), & l = 1 \\
  \text{ADAPCYCLE}(l - 1, \gamma, \mathbf{v}_{l-2}^m, \mathbf{v}_{l-1}^m, N_{l-1}, f_{l-1}, \mathbf{v}_{l-1}, \mu_1, \mu_2), & l > 1
  \end{cases}
  \]
- Compute correction on $D_{l-1}$:
  \[
  \hat{v}_{l-1}^m = \hat{w}_{l-1}^m - \mathbf{v}_{l-1}^m
  \]
- Set the solution on $G_{l-1}$:
  \[
  v_{l-1}^{m+1} = \hat{w}_{l-1}^m
  \]
- Interpolate correction to $\mathcal{T}_l$:
  \[
  \hat{v}_l^m = I_{l-1} \hat{v}_{l-1}^m
  \]
- Compute the corrected approximation on $\mathcal{T}_l$:
  \[
  v_l^{m,cgc} = \mathbf{v}_l^m + \hat{v}_l^m
  \]

(3) Postsmoothing:
- Compute $v_l^{m+1}$ by applying $\mu_2$ smoothing steps to $v_l^{m,cgc}$ on $\mathcal{T}_l$, i.e.,
  \[
  v_l^{m+1} = \text{SMOOTH}^{\mu_2}(l, v_l^{m,cgc}, N_l, f_l)
  \]
C GRID AND PRESSURE AFTER EACH ADAPTATION

$L = 1$: number of cells = 768

$L = 2$: number of cells = 2112

$L = 3$: number of cells = 4137
\[ L = 4: \text{number of cells} = 5010 \]

\[ L = 5: \text{number of cells} = 7551 \]

Figure 10: Adaptive grid (left) and pressure contours around the bump (right) after each adaptation