Application of Multiscale Techniques to Hyperbolic Conservation Laws

Birgit Gottschlich–Müller and Siegfried Müller

Abstract. The solution of hyperbolic conservation laws is characterized by the finite speed of propagation and the development of discontinuities. Stable and robust numerical schemes, e.g. finite volume schemes, are based on upwind techniques in order to handle shocks and contact discontinuities. In general, schemes of this type are very expensive. In 1993, Harten derived a concept to reduce the computational costs. It is based on a multiscale decomposition such that regions with singularities can be located. Near discontinuities, an expensive upwind scheme is applied; otherwise a much cheaper finite difference scheme of high order is used.
Harten's multiscale decomposition is based on the primitive function of the solution. This approach is inherently restricted to structured grids. In this paper we present an alternative multiscale decomposition using general reconstruction methods. Here, we consider the one–dimensional case and verify that our concept coincides with those of Harten's decomposition method. Furthermore, we apply our algorithm to dimensionally reduced hypersonic stagnation point flows around spheres. All these techniques can be extended to multidimensional problems even for unstructured grids (see [7]).

§1. Introduction

In the present paper, we consider partial differential equations of the form

$$\frac{\partial\, u(t,x)}{\partial\, t} + \frac{\partial\, f(u(t,x))}{\partial\, x} = s(x,u(t,x)), \qquad 0 < t < T,\ a < x < b \tag{1.1}$$

with $T > 0$ and $a, b \in \mathbb{R}$ fixed, so–called conservation laws. Herein, $u : [a,b] \to \mathbb{R}$ is the conservative quantity, $f : \mathbb{R} \to \mathbb{R}$ the flux and $s : \mathbb{R} \to \mathbb{R}$ the source term. In order to ensure a unique solution, an initial condition

$$u(0,x) = u_0(x), \qquad a < x < b \tag{1.2}$$

has to be imposed. In practical applications, boundary conditions of the form

$$g_a(u(t,a), u_x(t,a)) = 0, \qquad g_b(u(t,b), u_x(t,b)) = 0, \qquad 0 \le t \le T \tag{1.3}$$

are also needed.

It is well known that a smooth solution u of (1.1, 1.2) in general only exists for small times T even if u_0 is smooth. Due to the nonlinearity of the flux, discontinuities might develop and the solution is no longer differentiable. From a physical point of view, functions with such discontinuities are reasonable and should be admissible as solution of (1.1, 1.2). Therefore, an extended class of solutions has to be introduced, the so–called *weak solutions*. In one space dimension the weak solution is characterized by the evolution equation

$$\begin{aligned} &\int_I u(t+\tau,x)\,dx - \int_I u(t,x)\,dx = \\ &\int_t^{t+\tau} f(u(s,x_a))\,ds - \int_t^{t+\tau} f(u(s,x_b))\,ds + \int_I \int_t^{t+\tau} s(x,u(s,x))\,ds\,dx \end{aligned} \tag{1.4}$$

for all $I = [x_a, x_b] \subset \mathbb{R}$ and $[t, t+\tau] \subset \mathbb{R}_0^+$ (see [13]).

From equation (1.4), the class of finite volume schemes can be derived as follows. We discretize the continuum $[a,b] \times [0,T]$ in space and time by a spatial grid $a = x_0 < \ldots < x_N = b$, $N \in \mathbb{N}$, and a uniform temporal grid $t_n = \tau\, n$, $\tau := T/\overline{N}$, $\overline{N} \in \mathbb{N}$. Then we can write equation (1.4) in the form

$$\hat{u}_j^{n+1} = \hat{u}_j^n - \lambda_j \left(\hat{f}_j^n - \hat{f}_{j-1}^n \right) + \tau \hat{s}_j^n, \tag{1.5}$$

where $\hat{u}_j^n$ is the average of the solution in the cell $I_j = [x_{j-1}, x_j]$ at time level t_n, $\hat{s}_j^n$ the spatial and temporal average of the source, $\hat{f}_j^n$ the temporal average at the interface x_j defined by

$$\hat{u}_j^n := \frac{1}{|I_j|} \int_{I_j} u(t_n, x)\,dx, \tag{1.6}$$

$$\hat{s}_j^n := \frac{1}{\tau} \frac{1}{|I_j|} \int_{t_n}^{t_{n+1}} \int_{I_j} s(x, u(t_n, x))\,dx dt, \tag{1.7}$$

$$\hat{f}_j^n := \frac{1}{\tau} \int_{t_n}^{t_{n+1}} f(u(t, x_j))\,dt, \tag{1.8}$$

and $\lambda_j := \tau / |I_j|$ the ratio of the spatial and temporal step size with $|I_j| := x_j - x_{j-1}$. This means that the average of the solution on the new time level can be calculated from the one on the old time level, the flow over the cell interfaces and the average of the source term. In order to evaluate the flux and the source, the solution u itself has to be known explicitly. But, except for the linear case, this is not the case. Therefore, the averages $\hat{f}_j^n$ and $\hat{s}_j^n$

have to be approximated which results in an approximation $\hat{v}_j^n$ of $\hat{u}_j^n$. Then the discrete evolution equation reads according to equation (1.5)

$$\hat{v}_j^{n+1} = \hat{v}_j^n - \lambda_j \left(\hat{F}_j^n - \hat{F}_{j-1}^n \right) + \tau \hat{S}_j^n, \qquad j = 1, \ldots, N \tag{1.9}$$

with

$$\hat{F}_j^n := \hat{F}_j^n(\hat{v}_{j-p+1}^n, \ldots, \hat{v}_{j+p}^n), \quad p \in \mathbb{N} \tag{1.10}$$

$$\hat{S}_j^n := s(\hat{x}_j, \hat{v}_j^n), \quad \hat{x}_j := (x_{j-1} + x_j)/2. \tag{1.11}$$

Near to the boundary, he numerical flux $\hat{F}_j^n$ can be calculated by (1.10) only for $p \le j \le N - p$. In the case of $j < p$ or $N - p < j$, we either need further information for $\hat{v}_j^n$ or we use numerical fluxes based on one–sided stencils, i.e.,

$$\hat{F}_j^n = \begin{cases} \hat{F}_j^n(\hat{v}_1^n, \ldots, \hat{v}_{2p+1}^n) & , \quad 1 \le j < p \\ \hat{F}_j^n(\hat{v}_{N-2p}^n, \ldots, \hat{v}_N^n) & , \quad N - p < j < N \end{cases} .$$

In both cases, we have to ensure that the boundary conditions (1.3) are fulfilled. In view of nonlinear boundary conditions, we prefer the definition of auxiliary data. At the boundary, we have to apply the boundary conditions (1.3) and the definition of the averaged flux (1.8) for the calculation of the numerical fluxes $\hat{F}_0^n$ and $\hat{F}_N^n$.

Schemes that can be written in the form (1.9) are called finite volume schemes. They are of great importance from an analytical as well as from a practical point of view. During the last decade they have been applied to numerous practical applications in complex flow fields. Two characteristic features that most of the finite volume schemes have in common are solving Riemann problems at the cell interfaces and the reconstruction of a function by it's discrete cell averages to ensure upwinding and higher order accuracy (see [11]). Thus, the evaluation of the numerical flux is, in general, very expensive.

In order to reduce these costs, Harten developed the strategy of hybrid schemes (see [9]), i.e., in smooth flow regions a cheap finite difference scheme is used, and in regions with singularities an expensive upwind scheme. But how to distinguish between these regions? For this purpose, Harten developed a multiresolution technique similar to multigrid techniques. It is based on a sequence of grids

$$G_k := \{x_{k,j}\}_{j=0,\ldots,N_k} \subset [a, b], \quad k = 0, \ldots, L,$$

where L is the number of refinement levels and N_k the number of cells on each level which increases with k. The cells are given by $I_{k,j} := [x_{k,j-1}, x_{k,j}]$ with the interval length $h_{k,j} := x_{k,j} - x_{k,j-1}$, $j = 1, \ldots, N_k$. The grids are assumed to be nested, i.e., $G_k \subset G_{k+1}$, $k = 0, \ldots, L-1$. For simplification of notation, we assume throughout this paper that the discretization is uniform on each level, i.e.,

$$h_k = (b - a)/N_k, \quad k = 0, \ldots, L \tag{1.12}$$

and the factor of refinement between two levels is just 0.5, i.e.,

$$N_{k+1} = 2\ N_k, \quad k = 0, \ldots, L-1. \tag{1.13}$$

We emphasis that this is no essential constraint. Consequently, one has

$$\begin{array}{l} h_{k+1} = h_k/2,\ \lambda_{k+1} = 2\ \lambda_k, \\ x_{k+1,2\,j} = x_{k,j} = a + j\ h_k,\ j = 0, \ldots, N_k, \\ G_{k+1} = G_k \cup \{x_{k+1,2\,j-1}\}_{j=1,\ldots,N_k}, \\ I_{k,j} = I_{k+1,2\,j-1} \cup I_{k+1,2\,j}. \end{array} \tag{1.14}$$

On each level a finite volume scheme is now defined analogously to (1.9) by

$$\hat{v}_{k,j}^{n+1} = \hat{v}_{k,j}^{n} - \lambda_k \left(\hat{F}_{k,j}^{n} - \hat{F}_{k,j-1}^{n} \right) + \tau \hat{S}_{k,j}^{n}, \qquad j = 1, \ldots, N_k \tag{1.15}$$

with

$$\hat{F}_{k,j}^{n} := \hat{F}_{k,j}^{n}(\hat{v}_{L,j_0-p+1}^{n}, \ldots, \hat{v}_{L,j_0+p}^{n}), \qquad j_0 = 2^{L-k}\ j \tag{1.16}$$

$$\hat{S}_{k,j}^{n} := \frac{h_L}{h_k} \sum_{i=1}^{2^{L-k}} s(\hat{x}_{L,2^{L-k}(j-1)+i}, \hat{v}_{L,2^{L-k}(j-1)+i}^{n}), \tag{1.17}$$

$$\hat{x}_{k,j} := \frac{1}{h_k} \int\limits_{I_{k,j}} x\ dx = (x_{k,j-1} + x_{k,j})/2 = a + \left(j - \frac{1}{2} \right)\ h_k. \tag{1.18}$$

This scheme reads in vector notation

$$\hat{\mathbf{v}}_k^{n+1} = \hat{\mathbf{v}}_k^{n} \ - \ \lambda_k \left\{ \hat{\mathbf{F}}_{k,+}^{n} - \hat{\mathbf{F}}_{k,-}^{n} \right\} \ + \ \tau\ \hat{\mathbf{S}}_k^{n} \tag{1.19}$$

with

$$\hat{\mathbf{v}}_k^{n} = \left(\hat{v}_{k,j}^{n}\right)_{j=1,\ldots,N_k}, \ \hat{\mathbf{S}}_k^{n} = \left(\hat{S}_{k,j}^{n}\right)_{j=1,\ldots,N_k}, \ \hat{\mathbf{F}}_{k,\pm}^{n} = \left(\hat{F}_{k,j-\frac{1}{2}\pm\frac{1}{2}}^{n}\right)_{j=1,\ldots,N_k}.$$

We notice that (1.15) coincides with (1.9) for $k = L$. For $0 \le k < L$ the numerical fluxes and the source terms are calculated from the data on the finest grid, i.e., on level L. This is essential for the stability, as numerical experiments have shown.

The main idea of Harten's concept is to determine a linear relation between the approximations $\mathbf{v}_k^n \in \mathbb{R}^{N_k}$ and $\mathbf{v}_k^{n+1} \in \mathbb{R}^{N_{k+1}}$ of two neighboring levels. For this purpose the space $\mathbb{R}^{N_k}$ has to be completed to $\mathbb{R}^{N_{k+1}}$ by a vector $\mathbf{d}_k^n \in \mathbb{R}^{N_{k+1}-N_k}$. The quantities $d_{k,j}^n$ can be interpreted as *difference information* between the coarse grid k and the fine grid $k+1$. On level $k+1$, we have N_{k+1} coefficients, whereas there are only N_k coefficients on level k. Hence, we need $N_{k+1} - N_k$ additional information in order to reconstruct the information on the finer level from the data on the coarse level. One possible reconstruction technique is the interpolation of the coarse grid data and the evaluation at the additional points of the fine grid. This value can be

interpreted as a *prediction*. Of course, by this procedure an error is introduced, the so–called *prediction error*. It is, in fact, the difference between the interpolation and the exact solution on the fine level denoted by $d^n_{k,j}$.

The single–scale representation $\hat{v}^n_L$ can be successively transformed into the multiscale representation

$$\hat{\mathbf{v}}^n_{MR,L} = \left((\mathbf{v}^n_0)^T, (\mathbf{d}^n_0)^T, \ldots, (\mathbf{d}^n_{L-1})^T\right)^T . \tag{1.20}$$

Together with (1.15) now, a formula can be derived how to calculate the coefficients $d^{n+1}_{k,j}$ of the new time level t_{n+1}. If $|d^n_{k,j}|$ is sufficiently small, then this coefficient is *truncated*, i.e., it is set to zero. In this case the numerical flux is computed by an interpolation formula instead of the original one, i.e., the computational costs are reduced. Consequently, the computation itself is accelerated. However, we will see later that there must be a minimal number of truncated coefficients in order to gain a speed up since there is a certain overhead due to the multiscale transformation.

§2. Multiscale Transformation

In the following we will derive the multiscale decomposition for the averages $\hat{u}_{k,j}$ of an arbitrary function $u : [a,b] \to \mathbb{R}$ defined on every level $k = 0, \ldots, L$ by

$$\hat{u}_{k,j} := \frac{1}{|I_{k,j}|} \int_{I_{k,j}} u(x)\, dx, \quad j = 1, \ldots, N_k. \tag{2.1}$$

The main task is to determine a completion of $\mathbb{R}^{N_k}$ to $\mathbb{R}^{N_{k+1}}$ such that

$$\hat{u}_{k+1,j} = \sum_{r=1}^{N_k} m^{k,0}_{j,r}\ \hat{u}_{k,r} + \sum_{r=1}^{N_k} m^{k,1}_{j,r}\ d_{k,r}, \quad j = 1, \ldots, N_{k+1} \tag{2.2}$$

holds and, conversely, there exist coefficients $g^{k,0}_{r,j}$ and $g^{k,1}_{r,j}$ such that

$$\hat{u}_{k,j} = \sum_{r=1}^{N_{k+1}} g^{k,0}_{j,r}\ \hat{u}_{k+1,r}, \qquad d_{k,j} = \sum_{r=1}^{N_{k+1}} g^{k,1}_{j,r}\ \hat{u}_{k+1,r}. \tag{2.3}$$

Here, the coefficients $d_{k,j}$ are difference information between the data on levels k and $k+1$. These can be interpreted as a prediction error. Equations (2.2, 2.3) are called two–scale equations. In matrix–vector representation they read

$$\hat{\mathbf{u}}_{k+1} = \mathbf{M}_{k,0}\ \hat{\mathbf{u}}_k + \mathbf{M}_{k,1}\ \mathbf{d}_k \tag{2.4}$$

and

$$\hat{\mathbf{u}}_k = \mathbf{G}_{k,0}\ \hat{\mathbf{u}}_{k+1}, \qquad \mathbf{d}_k = \mathbf{G}_{k,1}\ \hat{\mathbf{u}}_{k+1}, \tag{2.5}$$

respectively, with the vectors

$$\hat{\mathbf{u}}_k = (\hat{u}_{k,j})_{j=1,\ldots,N_k}, \quad \mathbf{d}_k = (d_{k,j})_{j=1,\ldots,N_k}$$

and the *prolongation* and *restriction matrices*

$$\mathbf{M}_{k,0} = \left(m_{j,r}^{k,0}\right)_{\substack{j=1,\dots,N_{k+1}\\ r=1,\dots,N_k}} \quad , \quad \mathbf{M}_{k,1} = \left(m_{j,r}^{k,1}\right)_{\substack{j=1,\dots,N_{k+1}\\ r=1,\dots,N_k}}$$
$$\mathbf{G}_{k,0} = \left(g_{j,r}^{k,0}\right)_{\substack{j=1,\dots,N_k\\ r=1,\dots,N_{k+1}}} \quad , \quad \mathbf{G}_{k,1} = \left(g_{j,r}^{k,1}\right)_{\substack{j=1,\dots,N_k\\ r=1,\dots,N_{k+1}}} .$$

By means of (2.4, 2.5) the *single–scale representation* of $\hat{\mathbf{u}}_L$ can be transformed into the *multiscale representation*

$$\hat{\mathbf{u}}_{MR,L} = \left(\hat{\mathbf{u}}_0^T, \mathbf{d}_0^T, \dots, \mathbf{d}_{L-1}^T\right)^T$$

by introducing the *multiscale transformation* $\mathbf{T}_L : \mathbb{R}^{N_L} \to \mathbb{R}^{N_L}$ and the *inverse multiscale transformation* $\mathbf{T}_L^{-1} : \mathbb{R}^{N_L} \to \mathbb{R}^{N_L}$ defined by

$$\mathbf{T}_L := \mathbf{G}_0 \cdots \mathbf{G}_{L-1} \quad \text{and} \quad \mathbf{T}_L^{-1} := \mathbf{M}_{L-1} \cdots \mathbf{M}_0,$$

respectively, where the matrices $\mathbf{M}_k$ and $\mathbf{G}_k$ are given by

$$\mathbf{G}_k := \begin{pmatrix} \begin{pmatrix} \mathbf{G}_{k,0} \\ \mathbf{G}_{k,1} \end{pmatrix} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_k \end{pmatrix}, \quad \mathbf{M}_k := \begin{pmatrix} (\mathbf{M}_{k,0}, \mathbf{M}_{k,1}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_k \end{pmatrix}, \quad k = 0 \dots, L-1.$$

Here, we denote the identity matrix in $\mathbb{R}^{(N_L - N_{k+1}) \times (N_L - N_{k+1})}$ by $\mathbf{I}_k$. Then the relations between $\hat{\mathbf{u}}_L$ and $\hat{\mathbf{u}}_{MR,L}$ are

$$\hat{\mathbf{u}}_{MR,L} = \mathbf{T}_L \, \hat{\mathbf{u}}_L \quad \text{and} \quad \hat{\mathbf{u}}_L = \mathbf{T}_L^{-1} \, \hat{\mathbf{u}}_{MR,L},$$

respectively.

§3. Prolongation and Restriction

In the previous section, a general multiscale concept has been represented. It is essentially based on the prolongation and restriction matrices which will be derived in the following.

One possible method — examined by Gottschlich–Müller [6] — is the introduction of *refinable basis functions*, where the discrete values $\hat{u}_{k,j}$ and $d_{k,j}$ are identified with the coefficients of a serial expansion with respect to these functions. In this context the multiscale transformation can be interpreted as a change of basis (see [8]). The latter is performed by certain matrices that can be identified with prolongation and restriction matrices. The principal advantage inherently lies in the construction of appropriate basis functions by means of functional analysis that lead to a *stable* multiscale transformation. For details we refer to [4].

In the following, we will present a different concept for the prolongation and restriction matrices which has first been derived by Harten [9]. He has

introduced discretization and reconstruction mappings between a functional space and a sequence of vector spaces related to different grids. Consequently, two discretization levels are connected by appropriately composing the discretization and reconstruction operators. Since the multiresolution transformation is only based on the relation between two discretization levels, the functional space as well as the discretization and reconstruction operators are not really needed here.

Recalling definition (2.1) of the cell average, and assuming nested and uniform grids (1.12, 1.13, 1.14), we obtain a relation between the averages on two nested grids

$$\hat{u}_{k,j} = \frac{1}{h_k} \int\limits_{I_{k,j}} u(x)\, dx = (\hat{u}_{k+1,2j-1} \; + \; \hat{u}_{k+1,2j})/2. \tag{3.1}$$

Thus, the elements of $\mathbf{G}_{k,0}$ are given as

$$g_{j,r}^{k,0} = \begin{cases} 0.5 & , \quad r = 2j-1 \text{ or } r = 2j \\ 0 & , \quad \text{else} \end{cases} .$$

In order to determine the matrices $\mathbf{G}_{k,1}$, $\mathbf{M}_{k,0}$, $\mathbf{M}_{k,1}$, respectively, we now have to derive a prediction of the data on level $k+1$ by those of level k for $k = 0, \ldots, L-1$. Taking relation (3.1) into account, either $\hat{u}_{k+1,2j-1}$ or $\hat{u}_{k+1,2j}$ can be written in terms of $\hat{u}_{k,j}$ and the other cell average on level $k+1$, i.e.,

$$\hat{u}_{k+1,2j-1} = 2\hat{u}_{k,j} - \hat{u}_{k+1,2j} \text{ or } \hat{u}_{k+1,2j} = 2\hat{u}_{k,j} - \hat{u}_{k+1,2j-1}$$

Thus, we only have to predict *one* of those two values. Here, we calculate a prediction for $\hat{u}_{k+1,2j-1}$, i.e., we determine coefficients $a_{r,j}^k$ such that

$$\tilde{u}_{k+1,2j-1} := \sum_{r=1}^{N_k} a_{r,j}^k \; \hat{u}_{k,r} \tag{3.2}$$

is an approximation of $\hat{u}_{k+1,2j-1}$. Consequently, the prediction error is given by

$$d_{k,j} := \hat{u}_{k+1,2j-1} - \tilde{u}_{k+1,2j-1}. \tag{3.3}$$

Relations (3.1, 3.2, 3.3) lead to the matrices $\mathbf{M}_{k,0}$ and $\mathbf{M}_{k,1}$ since

$$\hat{u}_{k+1,2j-1} = \tilde{u}_{k+1,2j-1} \; + \; d_{k,j} = \sum_{r=1}^{N_k} a_{r,j}^k \; \hat{u}_{k,r} \; + \; d_{k,j},$$

$$\hat{u}_{k+1,2j} = 2\; \hat{u}_{k,j} \; - \; \hat{u}_{k+1,2j-1} = 2\; \hat{u}_{k,j} \; - \; \sum_{r=1}^{N_k} a_{r,j}^k \; \hat{u}_{k,r} \; - \; d_{k,j}.$$

Thus, the entries of $\mathbf{M}_{k,0}$ and $\mathbf{M}_{k,1}$ are given by

$$m_{j,r}^{k,0} = \begin{cases} a_{r,l}^k & , j = 2l-1 \\ 2\delta_{r,l} - a_{r,l}^k & , j = 2l \end{cases}, \quad m_{j,r}^{k,1} = \begin{cases} \delta_{r,l} & , j = 2l-1 \\ -\delta_{r,l} & , j = 2l \end{cases}. \tag{3.4}$$

The matrix $\mathbf{M}_{k,1}$ is uniquely determined by the definition of the cell average (2.1) and the prediction error (3.3) as well as by the assumption of nested grids (1.14). The matrix $\mathbf{M}_{k,0}$ is *not* uniquely determined yet. The coefficients $a_{r,j}^k$ of the prediction (3.2) are degrees of freedom which can be used in order to satisfy additional requirements. Thus, an entire class of multiresolution representations is given by (3.4). Clearly, in numerical applications, sparse matrices are preferable. In order to achieve these, the prediction should access only locally to a small number of coefficients, i.e., most of the coefficients should be zero. It remains to determine the matrix $\mathbf{G}_{k,1}$. We infer from equations (3.1, 3.2, 3.3) that

$$d_{k,j} = \hat{u}_{k+1,2j-1} - \frac{1}{2} \sum_{r=1}^{N_k} a_{r,j}^k \, (\hat{u}_{k+1,2r-1} \; + \; \hat{u}_{k+1,2r})$$

such that the matrix entries are given by

$$g_{j,r}^{k,1} = \frac{1}{2} \begin{cases} -a_{l,j}^k & , \quad r = 2l \\ 2\,\delta_{l,j} - a_{l,j}^k & , \quad r = 2l-1 \end{cases}.$$

§4. Prediction via Reconstruction

In order to determine the coefficients $a_{r,j}^k$ of the prediction (3.2), we introduce the reconstruction function $R_{k,j}^N : I_{k,j} \to \mathbb{R}$ defined by

$$R_{k,j}^N(x, \hat{\mathbf{u}}_k) = \sum_{i=0}^{N} \alpha_{k,j}^i(\hat{\mathbf{u}}_k) \, \omega_{k,j}^i(x), \qquad x \in I_{k,j}, \tag{4.1}$$

of the discrete averages $\hat{\mathbf{u}}_k$. The functions $\omega_{k,j}^i$ have to be chosen judiciously. Note first that the coefficients $\alpha_{k,j}^i$ are determined from the *recovery conditions*

$$\frac{1}{|I_{k,l}|} \int_{I_{k,l}} R_{k,j}^N(x, \hat{\mathbf{u}}_k) \, dx = \hat{u}_{k,l}, \qquad l \in S_{k,j}. \tag{4.2}$$

Herein, $S_{k,j}$ is a set of stencils that is chosen locally for each cell j on level k, i.e.,

$$S_{k,j} = \{ i_{k,j,l} \mid l = 0, \ldots, N \}.$$

In order to ensure the uniqueness of the solution of (4.2), there have to be exactly $N+1$ elements in this set. In general, the cell j is a member of $S_{k,j}$.

Otherwise, the recovery conditions result in an extrapolation. An example for $S_{k,j}$ and $\omega^i_{k,j}$ will be given in Section 8. One should mention that the above approach has also been suggested by Schröder–Pander and Sonar [18].

Substituting (4.1) into (4.2) renders a system of linear equations for the coefficients $\alpha^i_{k,j}$, i.e.,

$$\mathbf{A}_{k,j}\ \alpha_{k,j} = \mathbf{b}_{k,j} \tag{4.3}$$

with

$$(\mathbf{A}_{k,j})_{l,r} = \frac{1}{|I_{k,i_{k,j,l}}|} \int\limits_{I_{k,i_{k,j,l}}} \omega^r_{k,j}(x)\, dx, \qquad l,r = 0,\ldots,N,$$

$$(\mathbf{b}_{k,j})_l = \hat{u}_{k,i_{k,j,l}}, \quad (\alpha_{k,j})_l = \alpha^l_{k,j}, \qquad l = 0,\ldots,N.$$

In the context of ENO schemes, special reconstruction polynomials were constructed by Harten et al. [11] based on primitive functions or deconvolution techniques in one space dimension. Later, Harten and Chakravarthy [10] derived general polynomial reconstruction methods for the multidimensional space on unstructured grids. In recent years, Sonar [17] developed so–called *optimal reconstruction techniques* based on splines. They all have in common that they can be written in the form (4.1), where only the functions $\omega^i_{k,j}$ have to be altered.

Note next that the reconstruction functions $R^N_{k,j}$ are approximations of the function u itself, which is not known in general. A quite natural way of predicting $\tilde{u}_{k+1,2j-1}$ is

$$\tilde{u}_{k+1,2j-1} := \frac{1}{|I_{k+1,2j-1}|} \int\limits_{I_{k+1,2j-1}} R^N_{k,j}(x,\hat{\mathbf{u}}_k)\, dx = \sum_{i=0}^{N} \alpha^i_{k,j}\ \hat{\omega}^i_{k,j} = \sum_{r=1}^{N_k} a^k_{r,j}\ \hat{u}_{k,r} \tag{4.4}$$

with

$$\hat{\omega}^i_{k,j} := \frac{1}{|I_{k+1,2j-1}|} \int\limits_{I_{k+1,2j-1}} \omega^i_{k,j}(x)\, dx.$$

Here, the coefficients $\alpha^k_{r,j}$ are just the components of the solution of a system of linear equations (4.3). These depend linearly on the right hand side, which is $\hat{\mathbf{u}}_k$ here. Thus, the prediction also depends linearly on $\hat{\mathbf{u}}_k$, i.e., there exist coefficients $a^k_{r,j}$, $j = 1,\ldots,N_k$, according to (3.2).

§5. Update

We now apply the multiscale transformation to the approximation $\hat{\mathbf{v}}^n_L$ of the finite volume scheme (1.15). According to Section 2, we can calculate the multiscale representation $\hat{\mathbf{v}}^n_{MR,L}$ analogously to (1.20). Let us suppose that the multiresolution representation has been derived for a fixed time level. The question now arises how to calculate the multiresolution representation on the new time level by the one on the old time level. To this end, we make

use of the finite volume scheme on level k and the linearity of the multiscale representation.

On the coarsest level, one has by (1.15),

$$\hat{v}_{0,j}^{n+1} = \hat{v}_{0,j}^{n} - \lambda_0 \left(\hat{F}_{0,j}^{n} - \hat{F}_{0,j-1}^{n} \right) + \tau \hat{S}_{0,j}^{n}, \qquad j = 1, \ldots, N_0. \tag{5.1}$$

By means of definition (3.3) of the prediction error and equations (1.15, 1.19, 4.1, 4.2, 4.4) we get for the multiscale coefficients on level k

$$d_{k,j}^{n+1} = \hat{v}_{k+1,2j-1}^{n+1} \; - \; \tilde{v}_{k+1,2j-1}^{n+1} = d_{k,j}^{n} \; - \; \lambda_{k+1} \, d_{k,j}^{n} \left(\hat{\mathbf{F}}_k^n \right) \; + \; \tau \, d_{k,j}^{n} (\hat{\mathbf{S}}_k^n) \tag{5.2}$$

for all $j = 1, \ldots, N_k$. Here, the prediction error of the source term $d_{k,j}^{n}(\hat{\mathbf{S}}_k^n)$ and the prediction error of the numerical flux $d_{k,j}^{n} \left(\hat{\mathbf{F}}_k^n \right)$ related to the point $x_{k+1,2j-1}$ are defined by

$$d_{k,j}^{n}(\hat{\mathbf{S}}_k^n) := \hat{S}_{k+1,2j-1}^{n+1} \; - \; \tilde{S}_{k+1,2j-1}^{n}, \tag{5.3}$$

$$d_{k,j}^{n} \left(\hat{\mathbf{F}}_k^n \right) := \hat{F}_{k+1,2j-1}^{n} - \tilde{F}_{k+1,2j-1}^{n}, \tag{5.4}$$

with the predictions of the flux $\tilde{F}_{k+1,2j-1}^{n}$ and the source $\tilde{S}_{k+1,2j-1}^{n}$, respectively,

$$\tilde{F}_{k+1,2j-1}^{n} := \hat{F}_{k,j-1}^{n} \; + \; \frac{1}{h_k} \int\limits_{I_{k+1,2j-1}} R_{k,j}^{N} \left(x, \hat{\mathbf{F}}_{k,+}^{n} \; - \; \hat{\mathbf{F}}_{k,-}^{n} \right) \, dx, \tag{5.5}$$

$$\tilde{S}_{k+1,2j-1}^{n} := \frac{1}{h_{k+1}} \int\limits_{I_{k+1,2j-1}} R_{k,j}^{N} \left(x, \hat{\mathbf{S}}_k^n \right) \, dx. \tag{5.6}$$

According to (4.4), the interpolation of the flux (5.5) and the source (5.6) can be written as

$$\tilde{F}_{k+1,2j-1}^{n} = \hat{F}_{k,j-1}^{n} + \frac{1}{2} \sum_{r=1}^{N_k} a_{r,j}^{k} \left(\hat{F}_{k,r}^{n} - \hat{F}_{k,r-1}^{n} \right), \quad \tilde{S}_{k+1,2j-1}^{n} = \sum_{r=1}^{N_k} a_{r,j}^{k} \, \hat{S}_{k,r}^{n}.$$

Hence, the prediction error of the numerical flux takes the form

$$d_{k,j}^{n} \left(\hat{\mathbf{F}}_k^n \right) = \hat{F}_{k+1,2j-1}^{n} - \hat{F}_{k,j-1}^{n} - \frac{1}{2} \sum_{r=1}^{N_k} a_{r,j}^{k} \left(\hat{F}_{k,r}^{n} - \hat{F}_{k,r-1}^{n} \right). \tag{5.7}$$

The values for the flux and the source term on level $k+1$ in the even grid points are, according to (1.16, 3.1), given by

$$\hat{F}_{k+1,2j}^{n} = \hat{F}_{k,j}^{n}, \quad \hat{S}_{k+1,2j}^{n} = 2 \, \hat{S}_{k,j}^{n} - \hat{S}_{k+1,2j-1}^{n}.$$

In vector representation, (5.1, 5.2) read

$$\hat{\mathbf{v}}_0^{n+1} \;=\; \hat{\mathbf{v}}_0^{n} \;-\; \lambda_0 \left(\hat{\mathbf{F}}_{0,+}^{n} - \hat{\mathbf{F}}_{0,-}^{n}\right) \;+\; \tau\, \hat{\mathbf{S}}_0^{n}, \tag{5.8}$$

$$\hat{\mathbf{d}}_k^{n+1} \;=\; \hat{\mathbf{d}}_k^{n} \;-\; \lambda_{k+1}\, \mathbf{d}_k^{n} \left(\hat{\mathbf{F}}_k^{n}\right) \;+\; \tau\, \mathbf{d}_k^{n}(\hat{\mathbf{S}}_k^{n}), \qquad k = 1, \ldots, L. \tag{5.9}$$

Because of the relations (3.4) as well as (5.7), the identity

$$\frac{1}{2}\, \mathbf{M}_{k,0} \left(\hat{\mathbf{F}}_{k,+}^{n} - \hat{\mathbf{F}}_{k,-}^{n}\right) \;+\; \mathbf{M}_{k,1}\, \mathbf{d}_k^{n} \left(\hat{\mathbf{F}}_k^{n}\right) = \hat{\mathbf{F}}_{k+1,+}^{n} - \hat{\mathbf{F}}_{k+1,-}^{n} \tag{5.10}$$

is valid and, thus, we can prove by induction for $k = 0, \ldots, L-1$ the relation

$$\mathbf{M}_{k,0}\; \hat{\mathbf{v}}_k^{n+1} \;+\; \mathbf{M}_{k,1}\; \hat{\mathbf{d}}_k^{n+1} = \hat{\mathbf{v}}_{k+1}^{n+1},$$

where we have applied equations (5.9, 2.4, 5.10, 1.19). This means that the update of the coefficients $d_{k,j}^n$ of the multiscale representation is performed in such a way that it does not modify the finite volume scheme on the finest level, i.e., the inverse multiscale transformation of the updated coefficients $\hat{\mathbf{v}}_0^{n+1}$, $\mathbf{d}_k^{n+1}$ results in the average values $\hat{\mathbf{v}}_L^{n+1}$. This fact is essentially based on the evaluation of the numerical flux (1.16) and the source term (1.17) corresponding to the data of the finest level.

§6. Thresholding

We have already mentioned that Harten's concept is based on the reduction of flux evaluations. Instead of an expensive flux evaluation, a cheap finite difference formula is used in those parts of the flow field where the solution is smooth. In order to determine the smooth part, the multiscale coefficients can be used as follows. If $d_{k,j}^n$ is sufficiently small, we will truncate this coefficient, i.e., we set $d_{k,j}^n := 0$. First, we need to define the *set of significant points*

$$\mathcal{D}^{n+1} \subset \{(k,j) \mid k = 0, \ldots, L-1,\; j = 1, \ldots, N_k\}\,.$$

If $(k,j) \notin \mathcal{D}^{n+1}$, we replace the numerical flux by the predicted value such that in this case the equations (5.4, 5.2) are reduced to

$$d_{k,j}^n \left(\hat{\mathbf{F}}_k^{n}\right) = 0, \quad \hat{F}_{k+1,2j-1}^{n} = \tilde{F}_{k+1,2j-1}^{n}, \quad d_{k,j}^{n+1} = 0.$$

In the smooth part of the solution, the coefficients $d_{k,j}^n$ are of magnitude $O\,(h_k^p)$ since they are just the prediction error. This error becomes smaller when the grid becomes finer. In order to guarantee a prescribed tolerance when truncating the coefficients, we have to define tolerances ϵ_k on each level. Of course, the level specific tolerances should decrease when the grid becomes coarser, i.e.,

$$\epsilon_k := \epsilon_L / 2^{L-k}, \qquad k = 0, \ldots, L.$$

If the coefficient $d^n_{k,j}$ is not cancelled, i.e., it is *significant*, the information is propagated according to the finite speed of propagation — which is characteristic for hyperbolic problems — only into the neighboring cells during one time step (cf. Fig. 1).

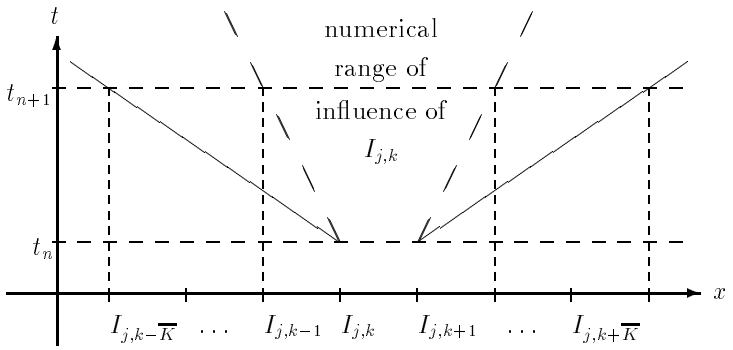

Fig. 1. Propagation of information

This results in significant coefficients in the neighborhood, i.e.,

$$|d^n_{k,j-l}| > \epsilon_k,\ -\overline{K} \le l \le \overline{K},\ 1 \le \overline{K} \le k.$$

It is characteristic for nonlinear conservation laws that discontinuities may develop with increasing time. It may happen that all coefficients on one level are cancelled. With increasing gradients the coefficients may become larger than the prescribed specific tolerance. In this case the scheme should allow that a level influences the next finer one, i.e.,

$$|d^n_{k,j}| > \epsilon_k \ \Rightarrow\ (k+1, 2j-1),\ (k+1, 2j) \in \mathcal{D}^{n+1},\ k < L-1.$$

We emphasize this since because of (1.17), there is no possibility to reduce the number of source evaluations.

§7. Computational Costs

The multiscale decomposition results in a certain overhead of floating point operations for each time level t_n. In order to accelerate the computation, this must be compensated by reducing the number of expensive numerical flux evaluations below a certain number. In the following, we determine the maximal number of non–significant points in order to gain a speed up. To this end, we recall that the matrices $\mathbf{G}_{k,1}$ and $\mathbf{M}_{k,0}$ are uniformly sparse, i.e., the number of non–vanishing coefficients $m^{k,0}_{j,r}$ and $g^{k,1}_{j,r}$, respectively, is not larger than a fixed number K independent of k and j. The numbers of floating point operations C^{MR}_D, C^{MR}_U, C^{MR}_F, C^{MR}_S for decoding, updating the multiscale coefficients on the new time level and evaluating the numerical flux and source

term (cf. algorithms in [7]), respectively, are determined by

$$\begin{aligned} C_D^{MR} &= 2\,m\,(2\,K+1)\,N_T, \\ C_U^{MR} &= m\,\left((3+2\sigma)(N_0+|\mathcal{D}^{n+1}|)+(3\,K+1+2\,(K+2)\sigma)\,N_T\right), \\ C_F^{MR} &= m\,\left(N_0+1+|\mathcal{D}^{n+1}|\right)\,F_f, \\ C_S^{MR} &= m\,N_L\,F_s\,\sigma, \end{aligned}$$

where F_f and F_s are the number of floating point operations for the evaluation of the numerical flux and the source per equation, and m is the number of equations. The parameter $\sigma = 1$ characterizes the inhomogeneous conservation law and $\sigma = 0$ the homogeneous case. The number $N_T := \sum_{k=0}^{L-1} N_k = N_L - N_0$ is calculated by (1.14). For the original scheme without multiscale decomposition and thresholding (1.9) we get analogously

$$C_U = m\,(3+2\sigma)\,N_L, \qquad C_F = m\,(N_L+1)\,F_f, \qquad C_S = m\,N_L\,F_s\,\sigma.$$

Therefore, the total amount of floating point operations is

$$C^{MR} := C_D^{MR} + C_U^{MR} + C_F^{MR} + C_S^{MR}, \quad C := C_U + C_F + C_S. \tag{7.1}$$

It is not quite correct that we have omitted in C^{MR} the costs for the thresholding procedure.

In order to achieve a speed up of the computation we need $C^{MR} < C$. Consequently, the number of significant points has to be bounded by

$$|\mathcal{D}^{n+1}| < \frac{F_f - 7\,K - 2\,(K+1)\sigma}{F_f + 3 + 2\sigma}\,N_T.$$

Hence, the minimal reduction factor $R_{sp,min}$ is

$$R_{sp} := \frac{N_T - |\mathcal{D}^{n+1}|}{N_T} > \frac{7\,K + 2\,(K+2)\sigma + 3}{F_f + 3 + 2\sigma} =: R_{sp,min} \tag{7.2}$$

Thus, for $F_f \le 7\,K + 2\,(K+1)\sigma$ no speed up is possible. However, with increasing F_f the factor $R_{sp,min}$ becomes smaller. This means that the more expensive the numerical flux evaluation becomes, the smaller the minimal reduction factor may become in order to accelerate the computation.

On the other hand, we are interested in the speed up factor $S := C/C_{MR}$ that can be achieved asymptotically. To this end, we determine the factor for an increasing number of refinement levels, i.e., $L \to \infty$. If we assume $|\mathcal{D}^{n+1}|/N_L \to 0,\ L \to \infty$, then the asymptotic speed up factor is given by

$$S_\infty = \frac{3 + 2\sigma + F_f + F_s\,\sigma}{7\,K + 2\,(K+2)\sigma + 3 + F_s\,\sigma}. \tag{7.3}$$

Consequently, the computation cannot be improved beyond a certain refinement level which also depends on the costs of the numerical flux. This constraint results from the fact that the costs of the algorithm correspond to the complexity N_L of the finest grid, and not to the complexity $N_0 + |\mathcal{D}^{n+1}|$ of the significant coefficients.

§8. Polynomial Reconstruction

Up to now, the multiscale transformation described in Section 2 is not uniquely determined. There is still one free parameter which has to be specified, namely, the reconstruction operator (4.1) introduced in Section 4. For our calculations presented in Section 9, we choose according to [10]

$$\omega_{k,j}^{i} := (x - \hat{x}_{k,j})^{i}, \qquad i = 0, \ldots, N$$

where $\hat{x}_{k,j}$ is the centroid of the cell $I_{k,j}$ given by (1.18) and the set of stencils

$$S_{k,j} := \{j - s + l \mid l = 0, \ldots, N\} = \{j - s, \ldots, j + s\}, \quad N = 2s.$$

Consequently, the reconstruction function $R_{k,j}^{N}$ defined by (4.1) is just the Taylor expansion of u in the point $\hat{x}_{k,j}$. We conclude from this fact and equations (1.14,) that

$$(A_{k,j})_{l,r} = \frac{h_k^r}{r+1} \left(\left(l - s + \frac{1}{2} \right)^{r+1} - \left(l - s - \frac{1}{2} \right)^{r+1} \right), \; l, r = 0, \ldots, N$$

$$\alpha_{k,j}^{l} = \frac{\partial^l u}{\partial x^l} + O\left(h_k^{N+1-l}\right) \quad \text{and} \quad R_{k,j}^{N}(x, \hat{\mathbf{u}}_k) = u(x) + O\left(h_k^{N+1}\right), \quad x \in I_{k,j}.$$

Opposite to this setting, Harten uses the reconstruction via the primitive function (see [9]), i.e., he interpolates the primitive function of u

$$U(x) := \int_{x_{k,0}}^{x} u(x)\, dx$$

at the points $x_{k,j}$

$$U_{k,j} := U(x_{k,j}) = h_k \sum_{i=1}^{j} \hat{u}_{k,i}, \quad l = j - s - 1, \ldots, j + s, \tag{8.1}$$

using the Lagrange interpolation polynomial $L_{k,j}^{N+1}$ of degree $N + 1$

$$L_{k,j}^{N+1}(x, \mathbf{U}_k) := \sum_{i=0}^{N+1} l_{i,j}^{k}(x)\, U_{k,j-s+i-1}, \qquad x \in I_{k,j},$$

where the Lagrange coefficients are defined by

$$l_{i,j}^{k}(x) := \prod_{\substack{l=0 \\ l \neq i}}^{N+1} \frac{x - x_{k,j-s+l-1}}{x_{k,j-s+i-1} - x_{k,j-s+l-1}}.$$

Recall that these satisfy

$$l_{i,j}^k(x_{k,j-s+l-1}) = \delta_{i,l}, \qquad i,l \in \{0,\ldots,N+1\}\,. \tag{8.2}$$

Equations (1.14, 8.1, 8.2) imply the relation

$$\hat{u}_{k,l} = \frac{1}{h_k} \int\limits_{I_{k,l}} \frac{d}{dx} L_{k,j}^{N+1}(x, \mathbf{U}_k)\, dx \tag{8.3}$$

between the cell averages on level k and the primitive function. Thus, the reconstruction polynomial via the primitive function is defined by

$$R_{k,j}^{N,RP}(x, \hat{\mathbf{u}}_k) := \frac{d}{dx} L_{k,j}^{N+1}(x, \overline{\mathbf{U}}_k), \qquad x \in I_{k,j}.$$

Comparing equations (8.3) and (4.2), the fundamental theorem of polynomials yields

$$R_{k,j}^{N,RP}(x, \hat{\mathbf{u}}_k) \;=\; R_{k,j}^{N}(x, \hat{\mathbf{u}}_k), \qquad x \in I_{k,j}.$$

Consequently, the multiscale transformation presented by Harten (see [9]) coincides with our transformation for nested and uniform grids. The prediction (3.2) can now be written as

$$\tilde{u}_{k+1,2j-1} = \frac{1}{h_{k+1}} \left(L_{k,j}^{N+1}(x_{k+1,2j-1}, \overline{\mathbf{U}}_k) - U_{k,j-1} \right)$$

such that the coefficients $a_{r,j}^k$ result from

$$\tilde{u}_{k+1,2j-1} = 2 \left(\sum_{l=0}^{N} \left(\sum_{i=l+1}^{N+1} l_{i,j}^k \right) \hat{u}_{k,j-s+l} \;-\; \sum_{l=0}^{s-1} \hat{u}_{k,j-s+l} \right) = \sum_{r=1}^{N_k} a_{r,j}^k \; \hat{u}_{k,r},$$

where

$$\begin{aligned} a_{j-s+l,j}^k &:= -2 \sum_{i=0}^{l} l_{i,j}^k, \qquad l = 0,\ldots,s-1, \\ a_{j+s-l,j}^k &:= -2 \sum_{i=l+1}^{N} l_{i,j}^k, \qquad l = 0,\ldots,s. \end{aligned}$$

For $N = 0, 2, 4$ the coefficients are listed in Table 1 (see $\mu = 0$). Note that they are *antisymmetric*, i.e.,

$$a_{j,j}^k = 1 \quad \text{and} \quad a_{j-s+l,j}^k = -a_{j+s-l,j}^k, \qquad l = 0,\ldots,s-1.$$

s	μ	...	$a^k_{j-4,j}$	$a^k_{j-3,j}$	$a^k_{j-2,j}$	$a^k_{j-1,j}$	$a^k_{j,j}$	$a^k_{j+1,j}$	$a^k_{j+2,j}$	$a^k_{j+3,j}$	$a^k_{j+4,j}$	...
0	0		...	...	...	0	1	0	...	...	...	...
1	-1	...	...	0	$-\frac{1}{8}$	$\frac{1}{2}$	$\frac{5}{8}$	0	...	...	...	...
1	0	...	...	...	0	$\frac{1}{8}$	1	$-\frac{1}{8}$	0	...	...	...
1	1	...	...	...	...	0	$\frac{11}{8}$	$-\frac{1}{2}$	$\frac{1}{8}$	0	...	...
2	-2	0	$-\frac{7}{128}$	$\frac{19}{64}$	$-\frac{11}{16}$	$\frac{61}{64}$	$\frac{63}{128}$	0	...	...	...	...
2	-1	...	0	$\frac{3}{128}$	$-\frac{9}{64}$	$\frac{13}{32}$	$\frac{49}{64}$	$-\frac{7}{128}$	0	...	...	...
2	0	...	...	0	$-\frac{3}{128}$	$\frac{11}{64}$	1	$-\frac{11}{64}$	$\frac{3}{128}$	0	...	...
2	1	...	...	...	0	$\frac{7}{128}$	$\frac{79}{64}$	$-\frac{13}{32}$	$\frac{9}{64}$	$-\frac{3}{128}$	0	...
2	2	...	...	...	...	0	$\frac{193}{128}$	$-\frac{61}{64}$	$\frac{11}{16}$	$-\frac{19}{64}$	$\frac{7}{128}$	0

Tab. 1. Coefficients $a^k_{r,j}$ for stencil $S^\mu_{k,j} = \{j - s + \mu + l \mid l = 0, \ldots, N = 2s\}$

In the special case considered here, the prediction error is given by the interpolation error of the interpolation polynomial $L^{N+1}_{k,j}(x, \bar{\mathbf{U}}_k)$ in the point $x_{k+1,2j-1}$, i.e.,

$$d_{k,j} = \hat{u}_{k+1,2j-1} \; - \; \tilde{u}_{k+1,2j-1} = \frac{1}{h_{k+1}} \, \frac{\omega(x_{k+1,2j-1}) \, U^{(N+2)}(\xi)}{(N+2)!},$$

where

$$\omega(x) := \prod_{l=0}^{N+1} (x - x_{k,j-s+l-1}), \quad \omega(x_{k+1,2j-1}) = h^{N+2}_{k+1} \prod_{l=0}^{N+1} (N - 2l + 1).$$

In order to determine the flux prediction $\tilde{F}^n_{k+1,2j-1}$ for the update of the multiscale coefficients, we observe that

$$R^N_{k,j}(x, \hat{\mathbf{F}}^n_{k,\pm}) \; = \; \frac{d}{dx} L^{N+1}_{k,j}(x, \hat{\mathbf{H}}^n_{k,\pm}), \qquad x \in I_{k,j},$$

with

$$\hat{\mathbf{H}}^n_{k,\pm} := \left(h_k \sum_{i=1}^{j} \hat{\mathbf{F}}^n_{k,i-\frac{1}{2}\pm\frac{1}{2}} \right)_{j=1,\ldots,N_k}.$$

This results in

$$\int_{I_{k+1,2j-1}} R^N_{k,j} \left(x, \hat{\mathbf{F}}^n_{k,\pm} \right) \, dx = L^{N+1}_{k,j} \left(x_{k+1,2j-1}, \hat{\mathbf{H}}^n_{k,\pm} \right) \; - \; \hat{H}^n_{k,j-\frac{3}{2}\pm\frac{1}{2}}$$

and

$$\frac{1}{h_k} \int\limits_{I_{k+1,\,2j-1}} R^N_{k,j}\left(x, \hat{\mathbf{F}}^n_{k,+} - \hat{\mathbf{F}}^n_{k,-}\right) dx = L^{N+1}_{k,j}\left(x_{k+1,2j-1}, \hat{\mathbf{F}}^n_k\right) - \hat{F}^n_{k,j-1}.$$

Thus, the flux prediction $\tilde{f}^n_{k+1,2j-1}$ is just the interpolation of the flux

$$\tilde{F}^n_{k+1,2j-1} = L^{N+1}_{k,j}\left(x_{k+1,2j-1}, \hat{\mathbf{F}}^n_k\right).$$

Comparing this result with Harten's multiresolution representation, the general concept derived here is the same.

One problem still remains: how to calculate $\hat{v}^n_{k,j}$, $j \in I_a$ and $\hat{v}^n_{k,N_k+j}$, $j \in I_b$ for $I_a := \{1-s, \ldots, 0\}$, $I_b := \{1, \ldots, s\}$ at the boundary of the computational domain? In general, we have two alternatives. We can either introduce artificial quantities outside of the computational domain, or we can use one–sided interpolation stencils, i.e.,

$$S_{k,j} = \begin{cases} \{1, \ldots, N\} & , \quad 1 \le j \le, s, \\ \{N_k - N, \ldots, N_k\} & , \quad N_k - s + 1 \le j \le N_k \end{cases}.$$

Here, we prefer one–sided stencils because this is more rigorous from a mathematical point of view. The definition of artificial quantities is based on physical arguments. For the stencils

$$S^\mu_{k,j} := \{j - s + \mu + l \mid l = 0, \ldots, N\}, \quad \mu = -s, \ldots, s,$$

the coefficients $a^k_{r,j}$ are summarized in Table 1.

§9. An Application

The calculation of steady hypersonic blunt body flow is of great importance for the design of re–entry vehicles because of the high heat transfer at the body surface. The flow field is characterized by a detached bow shock and a boundary layer at the wall (see Fig. 2).

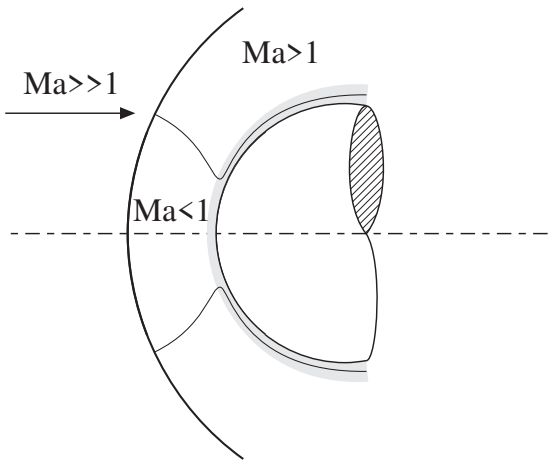

Fig. 2. Hypersonic blunt body flow

For a rotational symmetric flow around a sphere with radius R_{sph}, Klomfass and Müller [12] derived a dimensionally reduced model. This can be written in the form

$$\frac{\partial\,\mathbf{u}(t,r)}{\partial\,t}+\frac{\partial\,\mathbf{f}(\mathbf{u}(t,r))}{\partial\,r}=\mathbf{s}(r,\mathbf{u}(t,r)),\qquad 0<t,\ a<r<b, \tag{9.1}$$

where the flux $\mathbf{f}$ and the source $\mathbf{s}$ consist of two parts

$$\mathbf{f}(\mathbf{u})=\mathbf{f}^I(\mathbf{u})+\mathbf{f}^V(\mathbf{u},\partial\,\mathbf{u}/\partial\,r)\,,\qquad \mathbf{s}(\mathbf{u})=\mathbf{s}^I(\mathbf{u})+\mathbf{s}^V(\mathbf{u},\partial\,\mathbf{u}/\partial\,r)\,.$$

Here, the inviscid part describes the convection in the flow field, and the viscous part models the dissipative effects due to viscosity and heat conduction. In the sequel, we distinguish the inviscid from the viscid flow field by the superscripts I and V, respectively. In particular, one has

$$\mathbf{u}=\begin{pmatrix}\varrho\\ \varrho\,\overline{v}_r\\ \varrho\,\overline{v}_\theta\\ \varrho\,\overline{E}\end{pmatrix},\ \mathbf{f}^I=\begin{pmatrix}\varrho\,\overline{v}_r\\ \varrho\,\overline{v}_r^2+p\\ \varrho\,\overline{v}_r\overline{v}_\theta\\ \varrho\,\overline{v}_r\,\overline{H}\end{pmatrix},\ \mathbf{s}^I=-\frac{1}{r}\begin{pmatrix}2\ \varrho(\overline{v}_r+\overline{v}_\theta)\\ 2\ \varrho\overline{v}_r(\overline{v}_r+\overline{v}_\theta)\\ 3\ \varrho\overline{v}_\theta(\overline{v}_r+\overline{v}_\theta)-\overline{p}\\ 2\ \varrho\overline{H}(\overline{v}_r+\overline{v}_\theta)\end{pmatrix},$$

$$\mathbf{f}^V=-\begin{pmatrix}0\\ \overline{\tau}_{rr}\\ \overline{\tau}_{\theta\theta}\\ \overline{\tau}_{rr}\,\overline{v}_r-\overline{q}_r\end{pmatrix},\ \mathbf{s}^V=\frac{1}{r}\begin{pmatrix}0\\ \overline{\tau}_{rr}+\overline{\tau}_{r\theta}-\overline{\tau}_{\theta\theta}\\ 3\,\overline{\tau}_{r\theta}-\overline{\tau}_{\theta\theta}\\ 2\ (\overline{\tau}_{rr}\,\overline{v}_r+(\overline{\tau}_{r\theta}+\overline{\tau}_{\theta\theta})\ \overline{v}_\theta-\overline{q}_r)\end{pmatrix},$$

where $\overline{E}=\overline{e}+\frac{1}{2}\overline{v}_r^2$ and $\overline{H}=\overline{E}+p/\varrho$ denote total energy and total enthalpy The elements of the stress tensor are given by

$$\overline{\tau}_{rr}=\frac{4}{3}\,\mu\left(\frac{\partial\,\overline{v}_r}{\partial\,r}-\frac{\overline{v}_r+\overline{v}_\theta}{r}\right)=-2\,\overline{\tau}_{\theta\theta},\quad \overline{\tau}_{r\theta}=\mu\left(\frac{\partial\,\overline{v}_\theta}{\partial\,r}-\frac{\overline{v}_r+\overline{v}_\theta}{r}\right),$$

and $\overline{q}_r=-\lambda\,\partial\overline{T}/\partial r$ denotes the heat flux. For the viscosity coefficient μ, we apply Sutherland's law $\mu=\mu_\infty\,\overline{T}^{3/2}/(\overline{T}+110.4)$ and determine the heat conduction coefficient λ from the Prandtl number Pr which is assumed to be constant, i.e., $\lambda=\mu\,c_p/Pr$, $Pr=0.7$. We only consider a perfect gas, i.e., the specific heats $c_p=969.9\overline{3}$ [J/kg K] and $c_v692.8\overline{3}$ [J/kg K] — and hence the ratio $\gamma=c_p/c_v=1.4$ — are constant. Finally, the system is closed by the thermal and caloric equation of state

$$\begin{aligned}p &= \varrho\,R\,\overline{T}=p_\infty+\overline{p},\quad R=277.1\overline{3}\ \ [\mathrm{J/kg\ K}],\\ \overline{e} &= c_v\,\overline{T}=p/(\varrho(\gamma-1)),\end{aligned}$$

where p_∞ is the free–stream pressure. In the following, system (9.1) is referred to as the Navier–Stokes equations or, if the second order terms are neglected, as the Euler equations.

In the previous sections only *hyperbolic* conservation laws have been considered. System (9.1) is no longer hyperbolic but hyperbolic–parabolic. However, for high Reynolds–number flows, i.e., if the Reynolds number $Re = \overline{v}_{r,\infty}\, \varrho_\infty\, R_{sph}/\mu_\infty$ is sufficiently large, it is dominated by convective effects which are described by the first order terms, thus, our methods derived before can be applied.

In our computations, we examine a flow field prescribed by the free–stream conditions denoted by the subscript ∞ for the Mach number, the velocity and the pressure

$$Ma_\infty = 10,\ \overline{v}_r = 2928\ [\mathrm{m/s}]\ ,\ p_\infty = 19.16\ [\mathrm{Pa}]\ . \tag{9.2}$$

Consequently, the initial conditions for the conservative quantities are given by

$$\mathbf{u}(0,r) = \begin{pmatrix} \varrho \\ \varrho\,\overline{v}_r \\ \varrho\,\overline{v}_\theta \\ \varrho\,\overline{E} \end{pmatrix}_\infty = \begin{pmatrix} 3.1288 \times 10^{-4} & [\mathrm{kg/m^3}] \\ -9.1612 \times 10^{-1} & [\mathrm{kg/m^2\ s}] \\ 9.1612 \times 10^{-1} & [\mathrm{kg/m^2\ s}] \\ 1.3891 \times 10^{3} & [\mathrm{J/m^3}] \end{pmatrix},$$

and the remaining quantities are

$$T_\infty = 220.97\ \ [\mathrm{K}], \quad \mu_\infty = 1.458 \times 10^{-6}\ \ [\mathrm{kg/m\,s}], \quad Re = 95759\,.$$

We consider the computational domain $a = -0.1784,\ b = -R_{sph} = -0.1524$, and at the inflow boundary we use the free–stream conditions $\mathbf{u}(t,a) = \mathbf{u}_\infty$. The wall is assumed to be non–adiabatic, i.e., we prescribe a constant wall temperature $\overline{T}(t,b) = 1500\ \ [\mathrm{K}]$. Additionally, we suppose that the pressure is approximately constant in the boundary layer

$$\frac{\partial\, p(t,b)}{\partial\, r} = 0\,.$$

For the velocity, we apply the *no–slip condition* $\overline{v}_r(t,b) = \overline{v}_\theta(t,b) = 0$ for the Navier–Stokes equations, while for the Euler equations, we choose the *slip condition* at the wall $\overline{v}_r(t,b) = 0$ as well as vanishing gradients for the remaining quantities

$$\frac{\partial\ \varrho(t,b)}{\partial\, r} = \frac{\partial\, \overline{v}_\theta(t,b)}{\partial\, r} = \frac{\partial\, \overline{T}(t,b)}{\partial\, r} = 0\,.$$

The inviscid part of (9.1) is approximated by an ENO scheme of second order. This is based on the reconstruction of the characteristic variables via the primitive function, the so–called *characteristic reconstruction* (see [11]). At the cell interfaces, the Riemann problems are solved by the approximate Riemann solver of Roe (see [15]), and the viscous fluxes are approximated by central differences inside the computational domain and one–sided differences at the boundaries. Thus, the number of floating point operations of the numerical flux and the source per component are

$$F_{f^I} = 66.5, \qquad F_{f^V} = 4.75, \qquad F_{s^I} = 1.5, \qquad F_{s^V} = 4\,. \tag{9.3}$$

For our calculations, we apply the Harten's coefficients for $s = 1$ to each component of the vector of the conservative quantities (cf. Sec. 8). We have used $L = 6$ refinement levels and $N_0 = 10$ points on the coarsest grid. For the truncation process we replace $|d_{k,j}|$ by $\|\mathbf{d}_{k,j}\|_\infty$ in Section 6 and choose $\overline{K} = 1$. The threshold for the truncation on the finest level is $\epsilon_L = 0.05$. In order to achieve the steady state, we execute 60000 time steps with a fixed time discretization $\tau = 8 \times 10^{-9}$ for calculations for the Euler equations and 240000 time steps with $\tau = 2 \times 10^{-9}$ for the Navier–Stokes equations. All calculations were performed on an SGI workstation with an R8000 processor.

In Figures 3 and 4 the temperature is plotted for different time levels. Note that the shock wave detaches from the wall — here at the right side — and runs to the left. The shock speed decreases towards zero with increasing integration time. At time 4.8×10^{-4}, the steady state is approached. If dissipative effects are taken into account, a boundary layer develops additionally. In the shock layer, the temperature is approximately constant, because chemical and thermal relaxation processes are neglected here. Of course, for a more realistic calculation they have to be taken into account.

In inviscid stagnation–point calculations, the temperature at the wall is of special interest. For a perfect, gas the analytic solution can be calculated explicitly (see [1]) from $\overline{H}_s = \overline{H}_\infty$, $\overline{\mathbf{v}}_s = 0$ as

$$T_s = T_\infty \left(1 + \frac{\gamma - 1}{2}\, Ma_\infty^2\right).$$

From (9.2), we obtain for the present test case $T_s = 4640$ K, whereas our calculations with and without multiresolution predict $T_s = 4649$ K.

Opposite to the inviscid flow field, the wall heat–flux q_w is a very important quantity if viscosity and heat conduction are taken into account. However, there exists no explicit solution even in the case of an ideal gas. A reference value for the stagnation–point heat–flux can be determined with the formula of Fay and Riddell [5]

$$q_w = 0.76\,(Pr)^{-0.6}\,(\varrho_e\,\mu_e)^{0.4}\,(\varrho_w\,\mu_w)^{0.1}\,(H_e - H_w)\sqrt{\left(\frac{\partial\, v}{\partial\, y}\right)_s}$$

derived from numerical solutions of the boundary layer equations at the stagnation streamline. Here $(\partial\, v/\partial\, y)_s$ is the stagnation–point velocity gradient which can be estimated by the Newtonian theory (see [2]), i.e., $Ma \to \infty$, as

$$\left(\frac{\partial\, v}{\partial\, y}\right)_s = \frac{1}{R_{sph}}\sqrt{\frac{2(p_e - p_\infty)}{\varrho_e}}.$$

Here, the subscripts e and w denote the value of the quantity at the outer edge of the boundary layer and at the wall respectively. We identify the values

at the outer edge with the stagnation–point conditions predicted by the Euler equations, i.e.,

$$\begin{aligned} H_e &= 4500959\ [\mathrm{J/kg}]\ , & T_e &= 4648\ [\mathrm{K}]\ , \\ p_e &= 2475\ [\mathrm{Pa}]\ , & \varrho_e &= 0.001921\ [\mathrm{kg/m^3}]\ . \end{aligned}$$

From these we calculate $q_w = 127660\ [\mathrm{W/m^2}]$, whereas our calculations with and without multiresolution technique predict $q_w = 106970\ [\mathrm{W/m^2}]$ and $q_w = 107800\ [\mathrm{W/m^2}]$, respectively.

From the previous discussion, we conclude that the numerical results are not essentially influenced by the truncation process. In the following, we want to examine the gain of efficiency that is related to the calculations. The distribution of significant points as well as the temperature are presented in Figures 9 – 14 for the Euler equations and in Figures 15 – 20 for the Navier–Stokes equations. This shows that local changes in the gradient can, in fact, be detected by multiresolution techniques. The coefficients of the finest grid level are only involved in the neighborhood of the bow shock as well as the wall. Although no temperature gradient exists at the wall in the inviscid flow field (see Fig. 13), we know from earlier investigations that the error due to numerical dissipation results in a gradient of the total enthalpy (see [14]). This explains the significant points in the neighborhood of the wall.

In Figures 5 and 6, the number of significant points $|\mathcal{D}^{n+1}|$ is plotted for certain time levels. The calculations for the Euler and the Navier–Stokes equations show a very different behavior. In the latter case, the number is approximately constant. This is different for the inviscid flow field. When the bow shock detaches, this number increases until the shock speed is approximately zero, i.e., at $t \approx 1 \times 10^{-4}$. Approaching the steady state, the gradients within the shock layer decrease (see Fig. 3). Correspondingly, the number of significant points becomes smaller. This behavior may be explained by the different degrees of smoothness in the shock layer. The Euler equations are a system of first order whereas the Navier–Stokes equations are of second order. In order to determine an optimal thresholding procedure, the relation between the truncation process and the finite volume scheme applied to conservation laws taking viscosity into account has to be examined more thoroughly. Recently, Bihari [3] developed a method for this.

Relations (7.2) and (9.3) infer that at least a reduction factor of $R^I_{sp,min} = 0.48$ and $R^V_{sp,min} = 0.45$, respectively, is required in order to achieve a speed up of the computation where we have taken $K = 3$. Since the averages of the reduction number plotted in Figure 7 and 8 are higher, we predict a speed up of the computation. The reduction factors R_{sp} obtained in the computation are inversely proportional to the number $|\mathcal{D}^{n+1}|$ shown in Figures 5 and 6. However, a pure reduction of the numerical flux evaluations does not result in a speed up of the computation. The computation based on the multiresolution strategy takes 4214 seconds for the inviscid and 19478 seconds for the viscous flow field. For the same calculation without the multiresolution transformation, we need 5614 and 27542 seconds. Thus, the gain of computational time

by the multiresolution technique in connection with the thresholding strategy is about 25% and 29%, respectively. This corresponds to a speed up factor of $S^I = 1.33$ and $S^V = 1.41$. If we assume $|\mathcal{D}^{n+1}| = 200$ in (7.1) then we predict the values $S^I = 1.25$ and $S^V = 1.28$. In comparison, the asymptotic limit according to (7.3) is given by $S^I_\infty = 2.06$ and $S^V_\infty = 2.07$.

§10. Conclusion

In the present paper, we have derived a general multiscale concept for conservation laws. Our prediction is based on general reconstruction functions which are not necessarily restricted to polynomials (see [17]) in contrast to the interpolation of the primitive function as applied in Harten's ansatz. Hence, our strategy can be extended in a straightforward way to multidimensional and unstructured grids. First investigations have been performed in [8].

The application to dimensionally reduced hypersonic blunt body flow has rendered a gain of computational time of 25% for inviscid and 29% for viscous flow fields. Further investigations are necessary in order to verify the speed up by the multiscale strategy for multidimensional problems. It will be interesting to see whether the dimensionally splitting strategy of Sjögreen (see [16]) in two space dimensions based on Harten's one–dimensional multiscale transformation can be improved by our technique.

§11. Plots

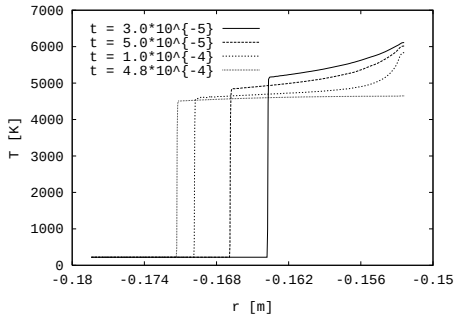

Fig. 3. Temperature, Euler equations

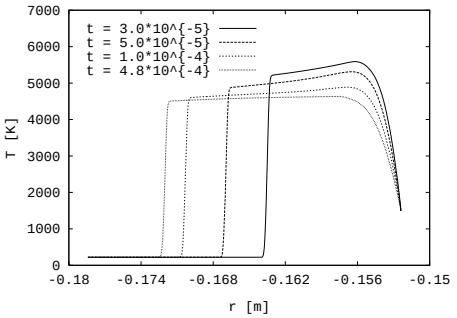

Fig. 4. Temperature, Navier–Stokes equations

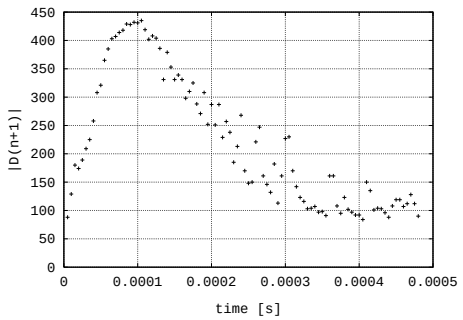

Fig. 5. Number of significant points, Euler equations

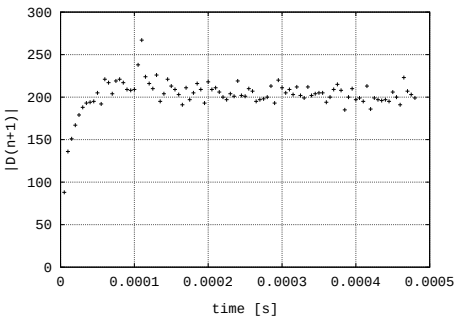

Fig. 6. Number of significant points, Navier–Stokes equations

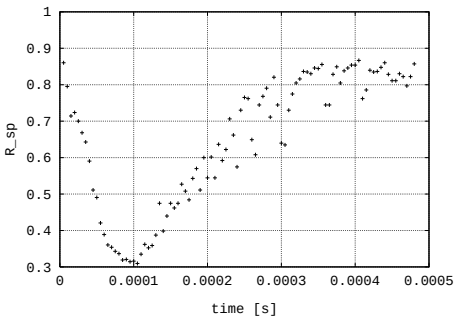

Fig. 7. Reduction factor, Euler equations

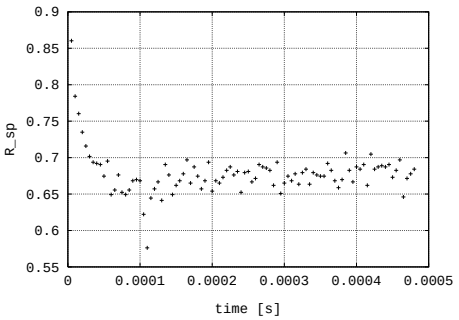

Fig. 8. Reduction factor, Navier–Stokes equations

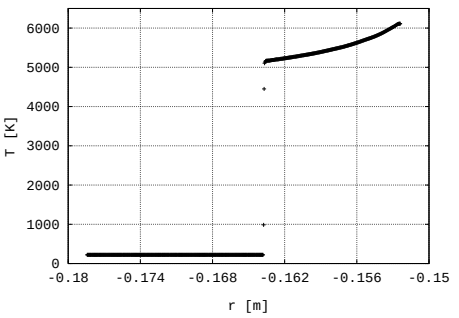

Fig. 9. Temperature at time 3×10^{-5}, Euler equations

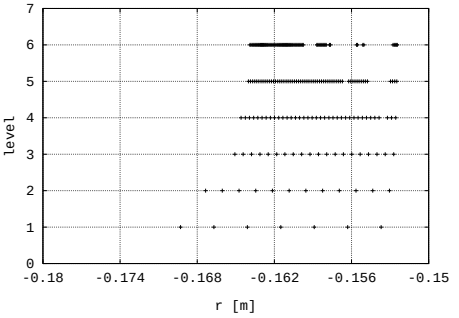

Fig. 10. Significant points at time 3×10^{-5}, Euler equations

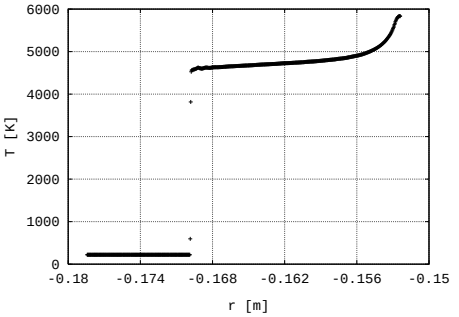

Fig. 11. Temperature at time 1×10^{-4}, Euler equations

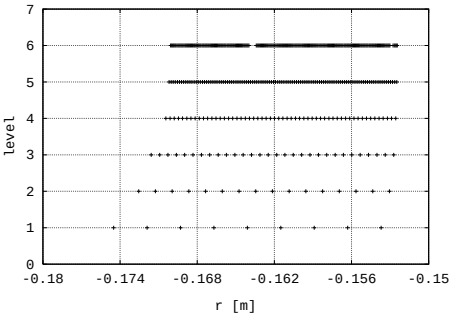

Fig. 12. Significant points at time 1×10^{-4}, Euler equations

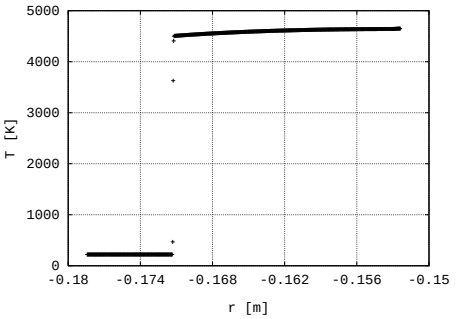

Fig. 13. Temperature at time 4.5×10^{-4}, Euler equations

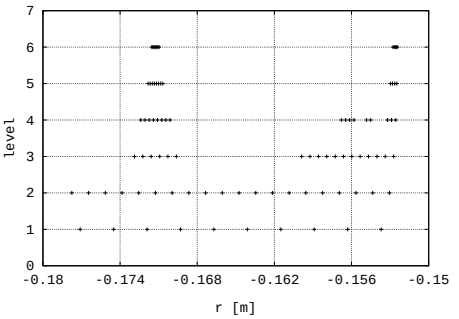

Fig. 14. Significant points at time 4.5×10^{-4}, Euler equations

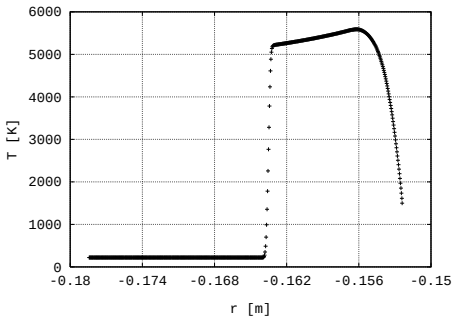

Fig. 15. Temperature at time 3×10^{-5}, Navier–Stokes equations

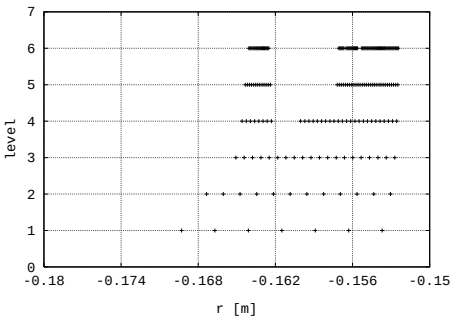

Fig. 16. Significant points at time 3×10^{-5}, Navier–Stokes equations

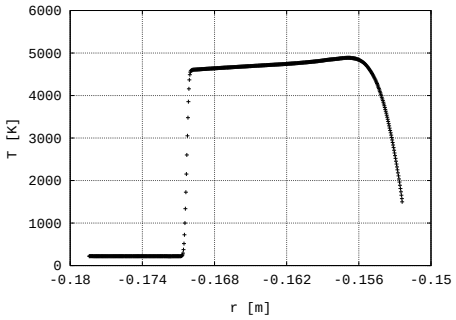

Fig. 17. Temperature at time 1×10^{-4}, Navier–Stokes equations

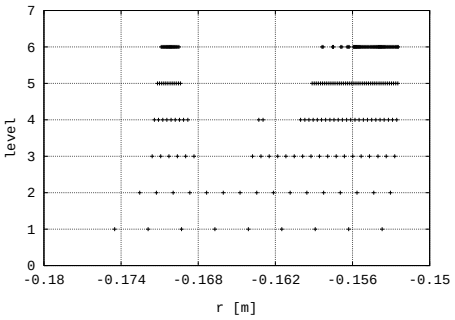

Fig. 18. Significant points at time 1×10^{-4}, Navier–Stokes equations

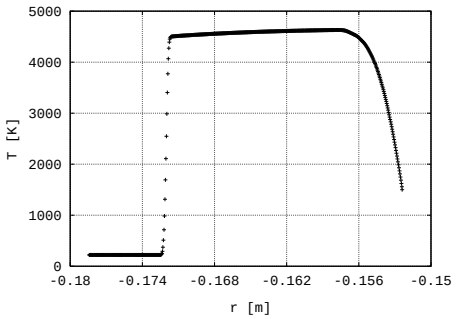

Fig. 19. Temperature at time 4.5×10^{-4}, Navier–Stokes equations

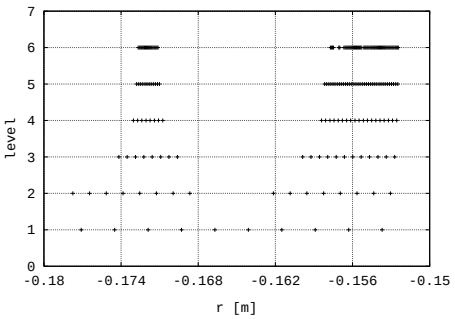

Fig. 20. Significant points at time 4.5×10^{-4}, Navier–Stokes equations

Acknowledgments. The authors are partially supported by the DFG (German Research Foundation) in the Collaborative Research Center SFB 401 "Modulation of Flow and Fluid–Structure Interaction at Airplane Wings" and the Postgraduate Program "Transport Processes in Hypersonic Flows" at the University of Technology Aachen.

References

1. J.D. Anderson, *Modern Compressible Flow*, MacGraw–Hill Series in Mechanical Engineering, 1982.

2. J.D. Anderson, *Introduction to Computational Fluid Dynamics*, VKI–Lecture Series, von Karman Institute for Fluid Dynamics, 1995.
3. B.L. Bihari, *Multiresolution schemes for conservation laws with viscosity*, J. Comp. Phys., **123**, 1996, 207–225.
4. J.M. Carnicer, W. Dahmen, J.M. Peña, *Local decomposition of refinable spaces and wavelets*, Appl. Comp. Harm. Anal. **3**, 1996, 127–153.
5. J. A. Fay, F. R. Riddell, *Theory of stagnation point heat transfer in dissociated air*, J. Aero. Sci. **25**(2), 1958, 73–85.
6. B. Gottschlich–Müller, *A Multiscale Scheme for Multidimensional Conservation Laws*, PhD thesis, RWTH Aachen, submitted 1998.
7. B. Gottschlich–Müller, S. Müller, *Multiscale concepts for conservation laws*, IGPM preprint No. 128, 1996.
8. B. Gottschlich–Müller, S. Müller, *On multiscale concepts for multidimensional conservation laws*, IGPM preprint No. 140, 1997.
9. A. Harten, *Multiresolution algorithms for the numerical solution of hyperbolic conservation laws*, Comm. Pure Appl. Math., **48**(12), 1995, 1305–1342.
10. A. Harten, S.R. Chakravarthy, *Multidimensional ENO schemes for general geometries*, ICASE Report, NASA Contractor Report NAS1-18605, 1992.
11. A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, *Uniformly high order accurate essentially non–oscillatory schemes III*, J. Comp. Phys., **71**, 1987, 231–303.
12. A. Klomfass, S. Müller, *Calculation of stagnation streamline quantities in hypersonic blunt body flows*, Shock Waves, **7**(1), 1997, 13–23.
13. R. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser, 1990.
14. S. Müller, *Erweiterung von ENO–Verfahren auf zwei Raumdimensionen und Anwendung auf hypersonische Staupunktprobleme*, PhD thesis, RWTH Aachen, Shaker–Verlag, 1993.
15. P.L. Roe, *Approximate Riemann solvers, parameter vectors and difference schemes*, J. Comp. Phys., **43**, 1981, 357–372.
16. B. Sjögreen, *Numerical experiments with the multiresolution scheme for the compressible Euler equations*, J. Comp. Phys., **117**, 1995, 251–261.
17. Th. Sonar, *Multivariate Rekonstruktionsverfahren zur numerischen Berechnung hyperbolischer Erhaltungsgleichungen*, Habilitationsschrift, TH Darmstadt, 1995.
18. F. Schröder–Pander, Th. Sonar, *Preliminary Investigations on Multiresolution Analysis on Unstructured Grids*, DLR Forschungsbericht IB 223-95 A 36, 1995.

Birgit Gottschlich–Müller
Institut für Geometrie und Praktische Mathematik
RWTH Aachen
52056 Aachen
Germany

birgit@igpm.rwth-aachen.de
http://www.igpm.rwth-aachen.de/∼birgit

Siegfried Müller
Institut für Geometrie und Praktische Mathematik
RWTH Aachen
52056 Aachen
Germany
mueller@igpm.rwth-aachen.de
http://www.igpm.rwth-aachen.de/∼mueller