Multiresolution Schemes for Conservation Laws

Wolfgang Dahmen Birgit Gottschlich–Müller* Siegfried Müller

Abstract

In recent years a variety of high–order schemes for the numerical solution of conservation laws has been developed. In general, these numerical methods involve expensive flux evaluations in order to resolve discontinuities accurately. But in large parts of the flow domain the solution is smooth. Hence in these regions an unexpensive finite difference scheme suffices. In order to reduce the number of expensive flux evaluations we employ a multiresolution strategy which is similar in spirit to an approach that has been proposed by A. Harten several years ago. Concrete ingredients of this methodology have been described so far essentially for problems in a single space dimension. In order to realize such concepts for problems with several spatial dimensions and boundary fitted meshes essential deviations from previous investigations appear to be necessary though. This concerns handling the more complex interrelations of fluxes across cell interfaces, the derivation of appropriate evolution equations for multiscale representations of cell averages, stability and convergence, quantifying the compression effects by suitable adapted multiscale transformations and last but not least laying grounds for ultimately avoiding the storage of data corresponding to a full global mesh for the highest level of resolution. The objective of this paper is to develop such ingredients for any spatial dimension and block structured meshes obtained as parametric images of Cartesian grids. We conclude with some numerical results for the two–dimensional Euler equations modeling hypersonic flow around a blunt body.

Key Words: conservation laws, hyperbolic partial differential equations, stable completions, multiscale transformations, biorthogonal wavelets, adaptive methods
AMS subject classification: 65M12, 65M55, 42C15, 47A20, 76Axx, 35L65

1 Introduction

Aside from essential theoretical open problems the numerical treatment of conservation laws still poses major challenges, especially with regard to high order schemes for several spatial variables. Interpreting the conservation law as an evolution equation for cell averages finite volume discretizations appear to be particularly well suited and physically motivated. However, the fluxes across cell interfaces, which arise in this context, still depend on conservative variables *not* on their cell averages. Thus if the scheme is to produce highly accurate solutions one needs a correspondingly highly acccurate reconstruction of

*The work of this author has been supported in part by the Deutsche Forschungsgemeinschaft in the Graduiertenkolleg "Transportvorgänge in Hyperschallströmungen" and by the SFB 401.

function values from cell averages. The concept of ENO reconstruction [HEOC87] aims at providing such a reconstruction while reliably detecting singularities across which no information should be passed by the numerical scheme. A major drawback of such schemes is the enormous computational cost. As a potential remedy Harten proposed in [Har95] a multiresolution strategy which quite in the spirit of image compression was to reduce these computational expenses significantly. The basic idea is to perform the actual flux computations only when much less expensive interpolation schemes are not sufficiently accurate. The detection of these different cases is based on a multiresolution representation of the computed data. Coefficients of small absolute value indicate a locally smooth behavior which can be well handled by interpolation while large coefficients require more computational efforts. It should be emphasized that such techniques are not tied to specific discretizations and solvers. It is an acceleration strategy that can, in principle, be hooked on any basic scheme. Accordingly, its ingredients have to be tuned to each particular application.

First realizations of this concept referred to one–dimensional model problems. Subsequently several attempts have been made to extend such techniques to several space variables. For instance, Sjögreen [Sjö95] generalized Harten's original concept with reconstruction via interpolation to two spatial dimensions employing *dimensional splitting*. By this we mean for the computation of the fluxes in one grid direction a multiscale transformation is only performed in this direction but on each line of the remaining direction. He concludes that this works best with a very fine grid but only two or three refinement levels. This means of course that already the coarsest grid is fine and that one essentially gives up exploiting the full potential of multiscale concepts. Moreover, dimensional splitting limits accuracy which motivates considering genuinely multivariate schemes.

In fact, one quickly recognizes that concrete realizations of Harten's multiresolution concept for several spatial variables poses serious problems indicated below. First truly multivariate studies can be found in [BH97]. There, however, the original concept has been modified in that the multiresolution analysis is only used as switching device in a hybrid scheme. Therefore this remains *conceptually* tied to the complexity of the full grid of highest resolution which is perhaps the most serious bottleneck for realistic multidimensional applications.

In general, a principal difficulty is caused by the following fact. An essential ingredient of the multiresolution concept is to compute the numerical fluxes initially on a coarse grid but with respect to data living on the fine grid. While in the univariate case the cell interfaces are just points and those corresponding to the coarsest level automatically belong also to the highest level of resolution this is no longer the case when dealing with several space variables. The derivation of suitable flux interpolation techniques for several spatial variables is a first central objective of this paper. We will focus here on a setting which does involve a hierachy of meshes but is flexible enough to deal with realistic domains with curved boundaries. More precisely, we have block structured grids in mind where each block arises from a parametric mapping of a Cartesian grid in a standard rectangular domain. Aside from the type of meshes the main issues that distinguish the present approach from previous studies is 1) to employ the concept of *stable completions* [CDP96] to raise the compression effect of corresponding adapted multiscale transformations and 2) the derivation of *evolution equations for the multiscale coefficients*. This appears to be an important step towards ultimately avoiding in contrast to Harten's original concept a fully globally refined mesh of the highest occuring level of resolution. Here we focus only on one such mesh component and confine the discussion for simplicity to scalar conservation laws. However, the extension to systems is straightforward.

The paper is organized as follows. In Section 2 we describe the model problem and recall briefly some relevant theoretical results along with the formulation of finite volume methods. In Section 3 we describe a multiresolution setting based on a hierarchy of nested curvilinear grids. Guided by the concept of *biorthogonal wavelets* we employ the idea of stable completions [CDP96] for the construction of multiscale transformations for cell average data that exhibit any desired order of vanishing polynomial moments needed for the reduction of flux evaluations. These transformations map arrays of cell averages into their multiscale representation which consist of cell averages on the coarsest level complemented by additional arrays of *detail coefficients* of incremental nature. Section 4 is devoted to the derivation of evolution equations for the detail coefficients. The reduction of flux evaluations is then based on a thresholding strategy for the detail coefficients. We conclude this section with a convergence and complexity analysis of the proposed scheme. Our numerical experiments in Section 5 concern unsteady hypersonic flow around a blunt body. We have chosen this example since it requires realistic curvilinear meshes and poses serious challanges to the robustness of a scheme since the conserved variables vary over several orders of magnitude. We conclude in Section 6 with a short summary and an overview on ongoing and future research motivated by the present results.

2 Model problem

In this paper we consider partial differential equations of the form

$$\frac{\partial u(t,\boldsymbol{x})}{\partial t} + \sum_{m=1}^{d} \frac{\partial f_m(u(t,\boldsymbol{x}))}{\partial x_m} = 0, \qquad 0 < t < T,\ \boldsymbol{x} \in \Omega, \tag{2.1}$$

with fixed $T > 0$ and a bounded domain $\Omega \subset \mathbb{R}^d$. These equations are referred to as *conservation laws* for the *conservative quantity* $u : [0,T] \times \Omega \to \mathbb{R}$. The *flux* in the m–th coordinate direction $f_m : \mathbb{R} \to \mathbb{R}$ is assumed to be in $\boldsymbol{C}^1(\mathbb{R})$. We have to impose initial and boundary conditions

$$\begin{aligned} u(0,\boldsymbol{x}) &= u_0(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \\ g(u(t,\boldsymbol{x}), u_{\boldsymbol{x}}(t,\boldsymbol{x})) &= 0, \quad 0 \le t \le T,\ \boldsymbol{x} \in \partial\Omega, \end{aligned} \tag{2.2}$$

in order to ensure uniqueness of the solution. It is well–known from the literature (cf. [LeV92]) that a smooth solution u of (2.1) can only be guaranteed to exist for a small period of time even if the initial data u_0 is smooth. Due to the non–linearity of the flux, discontinuities may develop. To cover such physically reasonable effects one has to pass to so–called *weak solutions*. These are characterized by the requirement that

$$\int_V u(t_1,\boldsymbol{x})\,d\boldsymbol{x} = \int_V u(t_0,\boldsymbol{x})\,d\boldsymbol{x} - \int_{t_0}^{t_1} \oint_\Gamma \langle \boldsymbol{f}(u(\tau,\boldsymbol{x})), \boldsymbol{n}(\boldsymbol{x})\rangle \, d\Gamma\, d\tau \tag{2.3}$$

holds for all $V \subset \Omega$ and $[t_0,t_1] \subset [0,T]$. Here, $\boldsymbol{f}$ is the flux vector $(f_1,\ldots,f_d)^T$ and $\Gamma := \partial V$ is the boundary of the volume V with outer normal $\boldsymbol{n}(\boldsymbol{x})$ at the point $\boldsymbol{x} \in \Gamma$.

Equation (2.3) is the starting point for the finite volume scheme. The volumes V are cells induced by suitable meshes. In the present paper we confine the discussion to block structured meshes, i.e., the whole domain Ω is partitioned into blocks and in each block the mesh is a parametric image of a simple Cartesian grid of the unit cube say. For simplicity we will consider here only a single block, i.e., the flow domain is parametrized by a regular coordinate transformation

$$\tau : [0,1]^d \to \Omega \tag{2.4}$$

with Jacobian determinant $J(\boldsymbol{\xi}) := |\det \tau'(\boldsymbol{\xi})| \neq 0,\ \boldsymbol{\xi} \in [0,1]^d$. The parameter domain is discretized by a uniform grid with gridpoints

$$\boldsymbol{\xi}_{\boldsymbol{k}} = (\xi_{\boldsymbol{k},1}, \ldots, \xi_{\boldsymbol{k},d})^T = (k_1 h_1, \ldots, k_d h_d)^T, \quad h_i = 1/N_i,\ N_i \in \mathbb{N}, \tag{2.5}$$

where

$$\boldsymbol{k} \in \bar{\mathcal{I}} := \times_{i=1}^d \{0, 1, \ldots, N_i\}.$$

Furthermore, for $\boldsymbol{k} \in \mathcal{I} := \times_{i=1}^d \{0, 1, \ldots, N_i - 1\}$, we denote the corresponding rectangular cells by

$$\Box_{\boldsymbol{k}} := \{\boldsymbol{\xi} \mid \xi_{\boldsymbol{k},i} \leq \xi_i \leq \xi_{\boldsymbol{k}+\boldsymbol{e}_i,i}\}.$$

where $\boldsymbol{e}_i$ is the i–th unit vector. Thus, we obtain the following structured mesh with grid points $\boldsymbol{x}_{\boldsymbol{k}} = \tau(\boldsymbol{\xi}_{\boldsymbol{k}})$, $\boldsymbol{k} \in \bar{\mathcal{I}}$, and cells $V_{\boldsymbol{k}} = \tau(\Box_{\boldsymbol{k}}) = \{\tau(\boldsymbol{\xi}) \mid \boldsymbol{\xi} \in \Box_{\boldsymbol{k}}\}$, $\boldsymbol{k} \in \mathcal{I}$, so that $\Omega = \bigcup_{\boldsymbol{k} \in I} V_{\boldsymbol{k}}$. By $\Gamma_{m,\boldsymbol{k}} := \{\boldsymbol{x} = \tau(\boldsymbol{\xi}) \mid \xi_{\boldsymbol{k},i} \leq \xi_i \leq \xi_{\boldsymbol{k}+\boldsymbol{e}_i,i},\ i \neq m,\ \xi_m = \xi_{\boldsymbol{k},m}\}$, we denote a typical facet of the cell $V_{\boldsymbol{k}}$ which is transversal to the mth coordinate direction. Specifically, $\Gamma_{m,\boldsymbol{k}}$ separates the neighboring cells $V_{\boldsymbol{k}}$ and $V_{\boldsymbol{k}-\boldsymbol{e}_m}$.

The temporal discretization is given by $t^n = n\,\Delta t$, where Δt is chosen according to the CFL–condition. After these preparations we define

$$|V_{\boldsymbol{k}}| := \int_{V_{\boldsymbol{k}}} d\boldsymbol{x}, \qquad \lambda_{\boldsymbol{k}} = \Delta t / |V_{\boldsymbol{k}}|, \qquad \bar{u}^n_{\boldsymbol{k}} := \frac{1}{|V_{\boldsymbol{k}}|} \int_{V_{\boldsymbol{k}}} u(t^n, \boldsymbol{x})\, d\boldsymbol{x}, \tag{2.6}$$

where $\bar{u}^n_{\boldsymbol{k}}$ are the cell averages of u at time t^n. Finally, this gives rise to the fluxes

$$\bar{f}^n_{m,\boldsymbol{k}} := \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \int_{\Gamma_{m,\boldsymbol{k}}} \boldsymbol{f}(u(t,\boldsymbol{x}))^T \boldsymbol{n}_{m,\boldsymbol{k}}(\boldsymbol{x})\, d\Gamma\, dt,$$

where $\boldsymbol{n}_{m,\boldsymbol{k}}(\boldsymbol{x})$ is the outer normal of $V_{\boldsymbol{k}}$ and $-\boldsymbol{n}_{m,\boldsymbol{k}}(\boldsymbol{x})$ the outer normal of $V_{\boldsymbol{k}-\boldsymbol{e}_m}$ to the boundary $\Gamma_{m,\boldsymbol{k}}$. Hence, we can rewrite (2.3) as

$$\bar{u}^{n+1}_{\boldsymbol{k}} = \bar{u}^n_{\boldsymbol{k}} - \lambda_{\boldsymbol{k}} \sum_{m=1}^d \left(\bar{f}^n_{m,\boldsymbol{k}+\boldsymbol{e}_m} - \bar{f}^n_{m,\boldsymbol{k}}\right), \qquad \boldsymbol{k} \in \mathcal{I}, \tag{2.7}$$

for $V = V_{\boldsymbol{k}},\ t_0 = t^n,\ t_1 = t^{n+1}$.

Equation (2.7) can be interpreted as an evolution equation for cell averages: the average of the solution at the new time level $n+1$ in the cell $V_{\boldsymbol{k}}$ can be computed from the average at the old time level n, the flow over all cell interfaces $\Gamma_{m,\boldsymbol{k}}$. However, for the evaluation of the fluxes we have to know the solution u itself and not just its cell averages at time level n. Except for the linear case the explicit representation of u is not available. Hence, we can determine $\bar{f}^n_{m,\boldsymbol{k}}$ only *approximately* which again results in an approximation $\bar{v}^{n+1}_{\boldsymbol{k}}$ for the average $\bar{u}^{n+1}_{\boldsymbol{k}}$. This yields a numerical method for (2.7), the so–called *finite volume scheme*,

$$\bar{v}^{n+1}_{\boldsymbol{k}} = \bar{v}^n_{\boldsymbol{k}} - \lambda_{\boldsymbol{k}} \sum_{m=1}^{d} \left(\bar{F}^n_{m,\boldsymbol{k}+\boldsymbol{e}_m} - \bar{F}^n_{m,\boldsymbol{k}} \right), \qquad \boldsymbol{k} \in \mathcal{I}. \tag{2.8}$$

The quantities $\bar{v}^n_{\boldsymbol{k}} \approx \bar{u}^n_{\boldsymbol{k}}$ approximate the cell averages of the conservative variables. Moreover, the *numerical fluxes* are given by

$$\bar{F}^n_{m,\boldsymbol{k}} = \bar{F}_m(\bar{v}^n_{\boldsymbol{k}+\boldsymbol{i}} \ : \ \boldsymbol{i} \in S) \approx \bar{f}^n_{m,\boldsymbol{k}}, \ \boldsymbol{k} \in \mathcal{I}, \tag{2.9}$$

where, for some fixed $p \in \mathbb{N}$, we fix a stencil $S = \{-p, \ldots, p-1\}^d$.

At the boundary, the fluxes (2.9) cannot be calculated in this form. Either, we have to modify the stencil S, or we need auxiliary data beyond the boundary. In both cases we have to ensure the boundary condition (2.2). If the latter is nonlinear we prefer the use of auxiliary data.

The numerical flux functions are supposed to be *consistent* with the physical flux, i. e.,

$$\bar{F}_m(\overline{v}^n_{\boldsymbol{k}+\boldsymbol{i}} \ : \ \boldsymbol{i} \in S) = \int_{\Gamma_{m,\boldsymbol{k}}} \boldsymbol{f}(\overline{v})^T \boldsymbol{n}_{m,\boldsymbol{k}}(\boldsymbol{x})\, d\Gamma \tag{2.10}$$

is satisfied for $\overline{v}_{\boldsymbol{k}+\boldsymbol{i}} = \overline{v}, \ \boldsymbol{i} \in S$ and $u|_{V_{\boldsymbol{k}}} = \overline{v} = const.$

Employing the abbreviations

$$\bar{\boldsymbol{v}}^n := \{\bar{v}^n_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathcal{I}}, \quad \Delta\bar{\boldsymbol{F}}^n = \left\{ \sum_{m=1}^{d} \left(\bar{F}^n_{m,\boldsymbol{k}+\boldsymbol{e}_m} - \bar{F}^n_{m,\boldsymbol{k}} \right) \right\}_{\boldsymbol{k}\in\mathcal{I}_L}, \quad \boldsymbol{\Lambda} := \operatorname{diag}\{\lambda_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathcal{I}}, \tag{2.11}$$

(2.8) can be expressed in vector notation as

$$\bar{\boldsymbol{v}}^{n+1} = \bar{\boldsymbol{v}}^n - \boldsymbol{\Lambda}\Delta\bar{\boldsymbol{F}}^n =: E_0\bar{\boldsymbol{v}}^n \tag{2.12}$$

where E_0 is the so–called *evolution operator* which, of course, is in general nonlinear.

It is well–known that stable discretizations have to be upwind–biased in order to avoid oscillations near discontinuities. Typical representatives for these are higher order upwind schemes, e. g. ENO–schemes (essentially non–oscillatory schemes, cf. [HEOC87]). The main drawback of such techniques is that the evaluation of the related numerical fluxes (2.9) is usually very expensive. On the other hand, away from discontinuities, a simple finite difference scheme of high order should be sufficient to resolve the smooth parts of the solution very accurately. Thus, a hybrid scheme should be capable of reducing the computational cost by utilizing expensive computations only where they are necessary. The main difficulty is the reliable detection of discontinuities. Here, multiscale strategies appear to offer appropriate tools. In this paper we will focus on one such variant which builds up on Harten's proposal [Har95]. In fact, we will consider a sequence of nested grids which are numbered from 0 to $L \in \mathbb{N}_0$, where 0 and L will always denote the coarsest and finest discretization level, respectively. With each grid in this hierarchy we associate approximate averages. From the differences between the data for two successive grids we will infer the location of singularities. The basic vehicle will be suitable *multiscale representations* of the data which explicitly yield this difference information in a possibly enhanced fashion.

3 The Multiresolution Setting

In contrast to Harten's original philosophy of a fully discrete multiresolution concept our approach draws from the concept of biorthogonal wavelets which offers strong tools for the stability and compression analysis. Moreover it suggests natural concrete choices of the relevant multiresolution ingredients that usually remain unspecified in the general methodology. A key observation is that cell averages relative to nested grids are naturally related to *biorthogonal wavelets*. The objective of this section is to clarify this connection.

3.1 Hierarchy of Curvilinear Meshes

We describe next the details of the announced hierachy of meshes and the corresponding multiresolution setting. Recall that we wish to consider a sequence of nested grids which are numbered from 0 to $L \in \mathbb{N}_0$, where 0 and L will always denote the coarsest and finest discretization level, respectively. For each coordinate direction m we fix a number $N_{0,m}$ that indicates the number of intervals in this direction on the coarsest grid and we set $h_{0,m} = 1/N_{0,m}$. For the finer levels we define corresponding quantities

$$N_{j,m} = 2\,N_{j-1,m} = 2^j N_{0,m}, \qquad h_{j,m} = \frac{1}{2}h_{j-1,m} = 2^{-j}h_{0,m}.$$

In analogy to (2.5)–(2.6) we define $\boldsymbol{\xi}_{j,\boldsymbol{k}}$, $\boldsymbol{x}_{j,\boldsymbol{k}}$, $\Box_{j,\boldsymbol{k}}$, $V_{j,\boldsymbol{k}}$, $|V_{j,\boldsymbol{k}}|$, $\lambda_{j,\boldsymbol{k}}$, $\mathcal{I}_j$ and $\bar{\mathcal{I}}_j$ for each level j. Due to the nestedness of the grids, we observe the two–level–relations

$$\boldsymbol{x}_{j,\boldsymbol{k}} = \boldsymbol{x}_{j+1,2\boldsymbol{k}}, \quad \boldsymbol{k} \in \bar{\mathcal{I}}_j, \quad V_{j,\boldsymbol{k}} = \bigcup_{\boldsymbol{e}\in E} V_{j+1,2\boldsymbol{k}+\boldsymbol{e}}, \quad \boldsymbol{k} \in \mathcal{I}_j, \tag{3.1}$$

where $E = \{0,1\}^d$ is the set of vertices of the d–dimensional unit cube. Moreover, since the parameter transformation τ is assumed to be regular we have

$$|V_{j,\boldsymbol{k}}| = \int_{V_{j,\boldsymbol{k}}} d\boldsymbol{x} = \int_{\Box_{j,\boldsymbol{k}}} |J(\boldsymbol{\xi})|d\boldsymbol{\xi} \sim 2^{-jd}. \tag{3.2}$$

In order to deal with cell averages in a multiresolution context we introduce the weighted characteristic functions

$$\varphi_{j,\boldsymbol{k}} := \frac{1}{|V_{j,\boldsymbol{k}}|}\chi_{V_{j,\boldsymbol{k}}} = \sum_{\boldsymbol{e}\in E} \frac{|V_{j+1,2\boldsymbol{k}+\boldsymbol{e}}|}{|V_{j,\boldsymbol{k}}|}\varphi_{j+1,2\boldsymbol{k}+\boldsymbol{e}}, \tag{3.3}$$

for the cell $V_{j,\boldsymbol{k}}$, i.e., $\|\varphi_{j,\boldsymbol{k}}\|_{L_1(\Omega)} = 1$. Note that the two–scale relation (3.3) for $\varphi_{j,\boldsymbol{k}}$ immediately yields a two–scale relation for cell averages

$$\bar{u}_{j,\boldsymbol{k}} = (u, \varphi_{j,\boldsymbol{k}})_\Omega = \sum_{\boldsymbol{e}\in E} \frac{|V_{j+1,2\boldsymbol{k}+\boldsymbol{e}}|}{|V_{j,\boldsymbol{k}}|}\bar{u}_{j+1,2\boldsymbol{k}+\boldsymbol{e}}.$$

Our goal is to transform these data into a format that represents successive updates when progressing to higher levels of resolution. To motivate the subsequent developments we will briefly discuss first a simple univariate situation.

3.2 The Haar Basis, Motivation

Let us consider the unit interval $[0,1]$ and the box-function

$$\varphi(\xi) = \chi_{[0,1)}(\xi), \qquad \xi \in \mathbb{R}.$$

Then the average of a function v on the cell $\Box_{j,k} := 2^{-j}(k + [0,1])$ has the form

$$\bar{v}_{j,k} = (v, \varphi_{j,k})_{[0,1]}, \quad k = 0, \ldots, 2^j - 1,$$

where the counterpart to (3.3) $\varphi_{j,k} := 2^j \varphi(2^j \cdot -k)$ denotes the L_1–normalized scaled translates of the box–function. We will indicate now how to decompose the cell averages with respect to a uniform dyadic partition of $[0,1]$ into cell averages with respect to coarser partitions and associated detail coefficients. To this end, note that in analogy to (3.3)

$$\varphi_{j,k} = \frac{1}{2}\left(\varphi_{j+1,2k} + \varphi_{j+1,2k+1}\right) \tag{3.4}$$

reexpresses a coarse scale box function in terms of fine scale box functions. On the other hand,

$$\psi_{j,k} := \frac{1}{2}\left(\varphi_{j+1,2k} - \varphi_{j+1,2k+1}\right), \tag{3.5}$$

can be used to retrieve together with a coarse scale box function any fine scale box function

$$\varphi_{j+1,2k} = \varphi_{j,k} + \psi_{j,k}, \quad \varphi_{j+1,2k+1} = \varphi_{j,k} - \psi_{j,k}. \tag{3.6}$$

In terms of cell averages (3.4) means

$$(v, \varphi_{j,k})_{[0,1]} = \frac{1}{2}(v, \varphi_{j+1,2k} + \varphi_{j+1,2k+1})_{[0,1]} = \frac{1}{2}\left(\bar{v}_{j+1,2k} + \bar{v}_{j+1,2k+1}\right),$$

while (3.6) says that

$$\bar{v}_{j+1,2k} = \bar{v}_{j,k} + \bar{d}_{j,k}, \quad \bar{v}_{j+1,2k+1} = \bar{v}_{j,k} - \bar{d}_{j,k}, \tag{3.7}$$

where

$$\bar{d}_{j,k} := (v, \psi_{j,k})_{[0,1]} \tag{3.8}$$

are the announced detail coefficients. Thus (3.7) shows how to recapture fine scale cell averages from coarse scale ones and detail coefficients.

It will simplify later developments to rewrite two-scale relations of the form (3.4), (3.5) and (3.6) in matrix vector form. To this end, let us group the box functions $\varphi_{j,k}$ and detail functions $\psi_{j,k}$ on level j as (column) vectors Φ_j, Ψ_j, respectively. Then (3.4) and (3.5) become

$$\Phi_j^T = \Phi_{j+1}^T \boldsymbol{M}_{j,0}, \quad \Psi_j^T = \Phi_{j+1}^T \boldsymbol{M}_{j,1},$$

where the columns of the $2^{j+1} \times 2^j$ matrices $\boldsymbol{M}_{j,0}, \boldsymbol{M}_{j,1}$ contain the two non–zero filter coefficients $\frac{1}{2}, \frac{1}{2}$ and $\frac{1}{2}, -\frac{1}{2}$ of $\varphi_{j,k}$ and $\psi_{j,k}$, respectively. Likewise, (3.6) becomes

$$\Phi_{j+1} = \tilde{\boldsymbol{M}}_{j,0}\Phi_j + \tilde{\boldsymbol{M}}_{j,1}\Psi_j,$$

where

$$\boldsymbol{M}_{j,e} = \frac{1}{2}\tilde{\boldsymbol{M}}_{j,e}, \quad e \in \{0,1\}. \tag{3.9}$$

Of course, since the composed matrices

$$\tilde{\boldsymbol{M}}_j := (\tilde{\boldsymbol{M}}_{j,0}, \tilde{\boldsymbol{M}}_{j,1}), \quad \boldsymbol{M}_j^T := \begin{pmatrix} \boldsymbol{M}_{j,0}^T \\ \boldsymbol{M}_{j,1}^T \end{pmatrix}$$

are inverses of each other

$$\tilde{\boldsymbol{M}}_j \boldsymbol{M}_j^T = \mathbf{I}, \tag{3.10}$$

they realize the change of bases between Φ_{j+1} and $\Phi_j \cup \Psi_j$. Note also that the $\tilde{\boldsymbol{M}}_{j,e}$ are the two–scale matrices for the *dual system*

$$\tilde{\Phi}_j = 2^{-j}\Phi_j, \qquad \tilde{\Psi}_j = 2^{-j}\Psi_j,$$

i.e.,

$$(\Phi_j, \tilde{\Phi}_j)_{[0,1]} = (\Psi_j, \tilde{\Psi}_j)_{[0,1]} = \boldsymbol{I}, \quad (\Phi_j, \tilde{\Psi}_j)_{[0,1]} = (\tilde{\Phi}_j, \Psi_j)_{[0,1]} = \mathbf{0}. \tag{3.11}$$

Here we have used for any finite collections of functions Θ, Φ the compact notation $(\Theta, \Phi)_\Omega := ((\vartheta, \varphi)_\Omega)_{\vartheta \in \Theta, \varphi \in \Phi}$. Thus the projection of every function $v \in L_1([0,1])$ onto piecewise constants with respect to the dyadic mesh with meshsize 2^{-j} is given by

$$v_j = \sum_{k=0}^{2^j-1} (v, \varphi_{j,k})_{[0,1]} \tilde{\varphi}_{j,k} =: \tilde{\Phi}_j^T \boldsymbol{c}, \tag{3.12}$$

that is, the array of cell averages is naturally related to a *function* living on $[0,1]$.

Let us point out next the incremental nature of the array $\boldsymbol{d}_j$, or to put it differently, why these coefficients actually get small when the underlying function is smooth so that there is not much detail to be added when progressing to a higher resolution. Recalling (3.8), noting that the orthogonality relations (3.11) imply that $(c, \psi_{j,k})_{[0,1]} = 0$ for any constant c and bearing the normalization $\|\psi_{j,k}\|_{L_1([0,1])} = 1$ in mind, one has

$$\begin{aligned} |d_{j,k}| &= |(v, \psi_{j,k})_{[0,1]}| = \inf_c |(v - c, \psi_{j,k})_{[0,1]}| \le \inf_c \|v - c\|_{L_\infty(\square_{j,k})} \\ &\le C 2^{-j} \|v'\|_{L_\infty(\square_{j,k})}. \end{aligned} \tag{3.13}$$

Thus, when v is differentiable on $\square_{j,k}$ and has a moderate derivative there the detail coefficients decay like 2^{-j} with increasing level. In other words, the arrays $\boldsymbol{d}_j$ immediately indicate where relevant detail is encountered. This suggests to discard negligible entries thereby *compressing* the original data while controling the loss of accuracy. It is precisely this mechanism that will be exploited in the sequel.

However, in quantitative terms the above decay 2^{-j} is not very strong and would permit little compression if one aims at higher order accurate schemes. It will therefore be a central objective to increase the compression effect by constructing complement functions $\psi_{j,k}$ that are not only orthogonal to constants but even to *all polynomials of some chosen higher degree*. In the simple case $\Omega = [0,1]$ this can be achieved by using instead of piecewise constants $\tilde{\varphi}_{j,k}$ a higher order biorthogonal system as described in [CDF92, DKU]. While this can immediately be carried over to Cartesian grids by means of tensor products our investigations have revealed that in combination with the parametric mapping τ the compression effect degrades significantly. We will therefore employ a different strategy based on *modifying* the factors in (3.10) in a certain systematic way to be described below.

3.3 Initial Stable Completions

A first idea that might come to mind is to consider first the Haar wavelets on the computational domain $\Box$ just as in the above univariate example and transform the whole setting into Ω by means of the mapping τ. The orthogonality (or better biorthogonality) relations (3.11) would then still hold with respect to a modified inner product. However, we found that this severly hurts the compression effect since polynomial moments will no longer vanish. Therefore we will use the Haar wavelets only to construct a simple and efficient *initial* multiscale decomposition. We will then modify this setting so as to increase the order of vanishing moments.

In analogy to the development in Section 3.2 the first step is to identify the two–scale relations that will be needed in the multiscale transformations for multivariate curvilinear meshes. It readily follows from (3.3) that for $\Phi_j := \{\varphi_{j,\boldsymbol{k}} \ : \ \boldsymbol{k} \in \mathcal{I}_j\}$ (see (3.3))

$$\Phi_j^T = \Phi_{j+1}^T \boldsymbol{M}_{j,0} \quad \text{where} \quad (\boldsymbol{M}_{j,0})_{2\boldsymbol{l}+\boldsymbol{e},\boldsymbol{k}} = \frac{|V_{j+1,2\boldsymbol{k}+\boldsymbol{e}}|}{|V_{j,\boldsymbol{k}}|}\delta_{\boldsymbol{k},\boldsymbol{l}}, \quad \boldsymbol{k},\boldsymbol{l} \in \mathcal{I}_j, \ \boldsymbol{e} \in E, \tag{3.14}$$

i.e., the columns of $\boldsymbol{M}_{j,0}$ are the filters from the two–scale relation (3.3). Moreover, in analogy to (3.5) the basis functions for the detail representations have the form

$$\breve{\psi}_{j,\boldsymbol{e},\boldsymbol{k}} := \sum_{\boldsymbol{e}' \in E} \frac{|V_{j+1,2\boldsymbol{k}+\boldsymbol{e}'}|}{|V_{j,\boldsymbol{k}}|}(-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'}\varphi_{j+1,2\boldsymbol{k}+\boldsymbol{e}'}, \quad \boldsymbol{k} \in \mathcal{I}_j, \ \boldsymbol{e} \in E^* := E \setminus \{\boldsymbol{0}\}, \tag{3.15}$$

where we have exploited the tensor product structure of functions on the computational domain $\Box$. Thus we obtain $2^d - 1$ different types of wavelets $\breve{\psi}_{j,\boldsymbol{e},\boldsymbol{k}}$ according to $\boldsymbol{e} \in E^*$. It is therefore useful to formulate a two–scale relation for each type

$$\breve{\Psi}_{j,\boldsymbol{e}}^T = \Phi_{j+1}^T \breve{\boldsymbol{M}}_{j,\boldsymbol{e}}, \quad \boldsymbol{e} \in E^*, \tag{3.16}$$

where

$$\breve{\boldsymbol{M}}_{j,\boldsymbol{e}} = \left(\breve{m}_{\boldsymbol{l},\boldsymbol{k}}^{j,\boldsymbol{e}}\right)_{\boldsymbol{l}\in\mathcal{I}_{j+1},\boldsymbol{k}\in\mathcal{I}_j}, \quad \breve{m}_{2\boldsymbol{l}+\boldsymbol{e}',\boldsymbol{k}}^{j,\boldsymbol{e}} = \frac{|V_{j+1,2\boldsymbol{k}+\boldsymbol{e}'}|}{|V_{j,\boldsymbol{k}}|}(-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'}\delta_{\boldsymbol{k},\boldsymbol{l}}. \tag{3.17}$$

To see that the functions $\varphi_{j,\boldsymbol{k}}$ together with the $\breve{\psi}_{j,\boldsymbol{e},\boldsymbol{k}}$ do span all piecewise constants on the next higher level $j+1$ we have to confirm that the composed matrix $\breve{\boldsymbol{M}}_j := (\breve{\boldsymbol{M}}_{j,\boldsymbol{e}})_{\boldsymbol{e}\in E}$, where we have set for simplicity $\breve{\boldsymbol{M}}_{j,\boldsymbol{0}} := \boldsymbol{M}_{j,0}$, is indeed invertible. We will do this by identifying its inverse $\breve{\boldsymbol{G}}_j$. To this end, it is convenient to define

$$\breve{\psi}_{j,\boldsymbol{0},\boldsymbol{k}} := \varphi_{j,\boldsymbol{k}}, \quad \boldsymbol{k} \in \mathcal{I}_j.$$

The key is the following reconstruction formula analogous to (3.6).

Lemma 3.1 *For any $\boldsymbol{k} \in \mathcal{I}_j$ and any $\boldsymbol{e} \in E$ one has*

$$\varphi_{j+1,2\boldsymbol{k}+\boldsymbol{e}} = 2^{-d}\frac{|V_{j,\boldsymbol{k}}|}{|V_{j+1,2\boldsymbol{k}+\boldsymbol{e}}|}\sum_{\boldsymbol{e}'\in E}(-1)^{\boldsymbol{e}'\cdot\boldsymbol{e}}\breve{\psi}_{j,\boldsymbol{e}',\boldsymbol{k}}.$$

Proof: We infer from (3.15) that

$$\sum_{\boldsymbol{e}'\in E}(-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'}\breve{\psi}_{j,\boldsymbol{e}',\boldsymbol{k}} = \sum_{\boldsymbol{e}''\in E}\frac{|V_{j+1,2\boldsymbol{k}+\boldsymbol{e}''}|}{|V_{j,\boldsymbol{k}}|}\varphi_{j+1,2\boldsymbol{k}+\boldsymbol{e}''}\sum_{\boldsymbol{e}'\in E}(-1)^{\boldsymbol{e}'\cdot(\boldsymbol{e}''+\boldsymbol{e})}.$$

Now note that $e_i'(e_i + e_i'') \in \{0, 2\}$ for $i = 1, \ldots, d$ provided that $\boldsymbol{e} = \boldsymbol{e}''$. Otherwise, one has for at least one index i_0 that $e_{i_0} \neq e_{i_0}''$ and hence $e_{i_0}' + e_{i_0}'' = 1$. Rewriting

$$\sum_{\boldsymbol{e}' \in E} (-1)^{\boldsymbol{e}' \cdot (\boldsymbol{e}'' + \boldsymbol{e})} = \prod_{i=1}^{d} \left(\sum_{e_i'=0}^{1} (-1)^{e_i'(e_i + e_i'')} \right),$$

we see that the right hand side equals $2^d \delta_{\boldsymbol{e}', \boldsymbol{e}''}$, which proves the claim. $\square$

One immediately concludes from Lemma 3.1 that the entries of $\breve{\boldsymbol{G}}_j = (\breve{\boldsymbol{G}}_{j,\boldsymbol{e}})_{\boldsymbol{e} \in E} = \breve{\boldsymbol{M}}_j^{-1}$ are given for $\boldsymbol{k}, \boldsymbol{l} \in \mathcal{I}_j$, $\boldsymbol{e}' \in E$, by

$$\breve{\boldsymbol{G}}_{j,\boldsymbol{e}} = \left(\breve{g}^{j,\boldsymbol{e}}_{\boldsymbol{k},\boldsymbol{l}}\right)_{\boldsymbol{k} \in \mathcal{I}_j, \boldsymbol{l} \in \mathcal{I}_{j+1}}, \quad \breve{g}^{j,\boldsymbol{e}}_{\boldsymbol{k}, 2\boldsymbol{l} + \boldsymbol{e}'} = 2^{-d} \frac{|V_{j,\boldsymbol{k}}|}{|V_{j+1, 2\boldsymbol{k}+\boldsymbol{e}'}|} (-1)^{\boldsymbol{e} \cdot \boldsymbol{e}'} \delta_{\boldsymbol{k}, \boldsymbol{l}}, \quad \boldsymbol{e} \in E,$$

which parallels (3.9).

Gathering the two–scale matrices for the complement basis functions $\breve{\psi}_{j,\boldsymbol{e},\boldsymbol{k}}$, $\boldsymbol{e} \in E^*$ in a single block

$$\breve{\boldsymbol{M}}_{j,1} := \left(\breve{\boldsymbol{M}}_{j,\boldsymbol{e}}\right)_{\boldsymbol{e} \in E^*} \tag{3.18}$$

and likewise $\breve{\boldsymbol{G}}_{j,1}$ we can write as in the univariate example

$$\breve{\boldsymbol{M}}_j = (\boldsymbol{M}_{j,0}, \breve{\boldsymbol{M}}_{j,1}), \quad \begin{pmatrix} \breve{\boldsymbol{G}}_{j,0} \\ \breve{\boldsymbol{G}}_{j,1} \end{pmatrix} = \breve{\boldsymbol{M}}_j^{-1}, \tag{3.19}$$

recall (3.10).

Given an array $\boldsymbol{c}_L = \{c_{L,\boldsymbol{k}} : \boldsymbol{k} \in \mathcal{I}_L\}$ we think of $\boldsymbol{c}_L$ as the cell averages

$$c_{L,\boldsymbol{k}} = (u, \varphi_{L,\boldsymbol{k}})_\Omega$$

of some locally integrable function u on Ω. Of course

$$\tilde{\varphi}_{j,\boldsymbol{k}} := \chi_{V_{j,\boldsymbol{k}}} \tag{3.20}$$

satisfies

$$(\varphi_{j,\boldsymbol{k}}, \tilde{\varphi}_{j,\boldsymbol{k}'})_\Omega = \delta_{\boldsymbol{k},\boldsymbol{k}'}, \quad \boldsymbol{k}, \boldsymbol{k}' \in \mathcal{I}_j,$$

so that as in (3.12)

$$u_L = \sum_{\boldsymbol{k} \in \mathcal{I}_j} c_{L,\boldsymbol{k}} \tilde{\varphi}_{L,\boldsymbol{k}}$$

may be viewed as the projection of u onto the piecewise constants with respect to the mesh on level L. In principle, we have now all the ingredients to transform these data into a multiscale format as follows. We wish to generate from $\boldsymbol{c}_L$ the coarse grid averages $\boldsymbol{c}_{L-1} = (\Phi_{L-1}, u)_\Omega$ and the detail coefficients $\boldsymbol{d}_{L-1} = (\breve{\Psi}_{L-1}, u)_\Omega$, where we have collected in $\breve{\Psi}_{L-1}$ all arrays of complement functions in $\breve{\Psi}_{L-1,\boldsymbol{e}}, \boldsymbol{e} \in E^*$. In fact, one obtains, by (3.14) and (3.16),

$$\boldsymbol{c}_{L-1} = \boldsymbol{M}_{L-1,0}^T (\Phi_L, u)_\Omega, \quad \boldsymbol{d}_{L-1} = \breve{\boldsymbol{M}}_{L-1,1}^T (\Phi_L, u)_\Omega.$$

Repeating this yields the *pyramid scheme* (see e.g. [Dah97])

$$\boldsymbol{c}_j = \boldsymbol{M}_{j,0}^T \boldsymbol{c}_{j+1}, \quad \boldsymbol{d}_j = \breve{\boldsymbol{M}}_{j,1}^T \boldsymbol{c}_{j+1}, \qquad j = L-1, L-2, \ldots, 0, \tag{3.21}$$

taking cell averages on level L into cell averages on the coarsest level complemented by arrays of detail coefficients. By the sparseness of the two–scale matrices, this transformation has the computational complexity of the finest mesh on level L.

Likewise we infer from (3.14), (3.16), (3.18) and (3.19) that

$$\boldsymbol{c}_L = (\Phi_L, u)_\Omega = \left(\breve{\boldsymbol{G}}_{L-1}^T \breve{\boldsymbol{M}}_{L-1}^T \Phi_L, u\right)_\Omega = \breve{\boldsymbol{G}}_{L-1,0}^T \boldsymbol{c}_{L-1} + \breve{\boldsymbol{G}}_{L-1,1} \boldsymbol{d}_{L-1},$$

which shows that the inverse transformation that takes the multiscale format into the original cell average format reads

$$\boldsymbol{c}_{j+1} = \breve{\boldsymbol{G}}_{j,0}^T \boldsymbol{c}_j + \breve{\boldsymbol{G}}_{j,1} \boldsymbol{d}_j, \quad j = 0, 1, \ldots, L-1. \tag{3.22}$$

Again the coefficients in the arrays $\boldsymbol{d}_j$ have the character of updates that should be small in regions where the underlying function is smooth. However, according to (3.13), we can expect at best a decay like 2^{-j} with increasing levels j. We will show next how to improve on this compression effect.

3.4 Changing Stable Completions

Note that the matrix $\breve{\boldsymbol{M}}_{j,1}$ constructed above is only *one* among a continuum of ways to complement the refinement matrix $\boldsymbol{M}_{j,0}$ to an invertible square matrix $\breve{\boldsymbol{M}}_j$. Such *completions* are called *(uniformly) stable* if the $\breve{\boldsymbol{M}}_j$ and their inverses $\breve{\boldsymbol{G}}_j$ have uniformly bounded operator norms with respect to suitable vector norms [CDP96]. Since both $\breve{\boldsymbol{M}}_j$ and $\breve{\boldsymbol{G}}_j$ have a uniformly bounded number of non–zero entries in each row and column whose modulus stays also bounded the above completions are stable in any ℓ_p–norm. Moreover, any special stable completion determines a particular complement of the space of piecewise constants on level j in the space of piecewise constants on level $j+1$ and along with it a basis $\breve{\Psi}_j$ which is stable in the sense that

$$\|\boldsymbol{d}_j^T \breve{\Psi}_j\|_{L_1(\Omega)} \sim \|\boldsymbol{d}_j\|_{\ell_1}.$$

Thus searching for another complement basis Ψ_j, say, that has better cancellation properties as those in (3.13) in the sense that higher order polynomial moments vanish is equivalent to varying the completions of $\boldsymbol{M}_{j,0}$. We will exploit here that this latter task can be treated in a systematic fashion proposed in [CDP96]. In fact, we will need here only a special case that can also be found in [Swe]. In the present setting the idea is very simple. Consider any matrices

$$\boldsymbol{L}_j = (\boldsymbol{L}_{j,\boldsymbol{e}})_{\boldsymbol{e} \in E^*}, \quad \boldsymbol{L}_{j,\boldsymbol{e}} : \mathbb{R}^{N_j} \to \mathbb{R}^{N_j},$$

which will always be required to define (uniformly) bounded mappings on ℓ_1. Then one trivially has

$$\boldsymbol{I} = \breve{\boldsymbol{M}}_j \breve{\boldsymbol{G}}_j = \breve{\boldsymbol{M}}_j \begin{pmatrix} \boldsymbol{I} & \boldsymbol{L}_j \\ \boldsymbol{0} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & -\boldsymbol{L}_j \\ \boldsymbol{0} & \boldsymbol{I} \end{pmatrix} \breve{\boldsymbol{G}}_j =: \boldsymbol{M}_j \boldsymbol{G}_j,$$

i.e.,

$$\boldsymbol{M}_{j,1} = \boldsymbol{M}_{j,0}\boldsymbol{L}_j + \breve{\boldsymbol{M}}_{j,1}, \quad \boldsymbol{G}_{j,0} = \breve{\boldsymbol{G}}_{j,0} - \boldsymbol{L}_j\breve{\boldsymbol{G}}_{j,1} \quad \boldsymbol{G}_{j,1} = \breve{\boldsymbol{G}}_{j,1}, \tag{3.23}$$

where of course $\boldsymbol{M}_{j,0}$, the refinement matrix for the weighted characteristic functions needed for the cell averages, also remain unchanged. Under the above assumptions on the $\boldsymbol{L}_j$ the matrices $\boldsymbol{M}_{j,1}$ are still stable completions of the $\boldsymbol{M}_{j,0}$. It will be usefull to record the relations (3.23) in blockwise form

$$\boldsymbol{M}_{j,\boldsymbol{e}} = \boldsymbol{M}_{j,0}\boldsymbol{L}_{j,\boldsymbol{e}} + \breve{\boldsymbol{M}}_{j,\boldsymbol{e}}, \quad \boldsymbol{G}_{j,0} = \breve{\boldsymbol{G}}_{j,0} - \sum_{\boldsymbol{e}\in E^*} \boldsymbol{L}_{j,\boldsymbol{e}}\breve{\boldsymbol{G}}_{j,\boldsymbol{e}} \quad \boldsymbol{G}_{j,\boldsymbol{e}} = \breve{\boldsymbol{G}}_{j,\boldsymbol{e}}, \quad \boldsymbol{e} \in E^*,$$

as well as componentwise

$$\begin{aligned} m^{j,\boldsymbol{e}}_{2\boldsymbol{l}+\boldsymbol{e}',\boldsymbol{k}} &= \frac{|V_{j+1,2\boldsymbol{l}+\boldsymbol{e}'}|}{|V_{j,\boldsymbol{k}}|}\left(L^{j,\boldsymbol{e}}_{\boldsymbol{l},\boldsymbol{k}} + (-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'}\delta_{\boldsymbol{k},\boldsymbol{l}}\right), \quad \boldsymbol{k},\boldsymbol{l} \in \mathcal{I}_j, \ \boldsymbol{e} \in E^* \qquad (3.24) \\ g^{j,0}_{\boldsymbol{k},2\boldsymbol{l}+\boldsymbol{e}'} &= 2^{-d}\frac{|V_{j,\boldsymbol{k}}|}{|V_{j+1,2\boldsymbol{l}+\boldsymbol{e}'}|}\left(\delta_{\boldsymbol{k},\boldsymbol{l}} - \sum_{\boldsymbol{e}\in E^*}(-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'}L^{j,\boldsymbol{e}}_{\boldsymbol{k},\boldsymbol{l}}\right), \quad \boldsymbol{k},\boldsymbol{l} \in \mathcal{I}_j. \end{aligned}$$

One immediately infers from (3.24) the following relation

$$\frac{m^{j,\boldsymbol{e}}_{2\boldsymbol{l}+\boldsymbol{e}',\boldsymbol{k}}}{|V_{j+1,2\boldsymbol{l}+\boldsymbol{e}'}|} = \frac{m^{j,\boldsymbol{e}}_{2\boldsymbol{l},\boldsymbol{k}}}{|V_{j+1,2\boldsymbol{l}}|} + \frac{(-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'} - 1}{|V_{j,\boldsymbol{l}}|}\delta_{\boldsymbol{k},\boldsymbol{l}}, \quad \boldsymbol{k},\boldsymbol{l} \in \mathcal{I}_j, \boldsymbol{e} \in E^*, \boldsymbol{e}' \in E, \tag{3.25}$$

which is independent of the choice of the $\boldsymbol{L}_j$ and will be needed later.

3.5 Higher Order Vanishing Moments

The new complement bases $\Psi_j^T := \Phi_{j+1}^T \boldsymbol{M}_{j,1}$ given by

$$\psi_{j,\boldsymbol{e},\boldsymbol{k}} = \sum_{\boldsymbol{l}\in\mathcal{I}_{j+1}} m^{j,\boldsymbol{e}}_{\boldsymbol{l},\boldsymbol{k}}\varphi_{j+1,\boldsymbol{l}}, \quad \boldsymbol{k} \in \mathcal{I}_j, \boldsymbol{e} \in E^*, \tag{3.26}$$

will be chosen now so as to satisfy

$$\int_\Omega \boldsymbol{x}^{\boldsymbol{\alpha}}\psi_{j,\boldsymbol{e},\boldsymbol{k}}(\boldsymbol{x})d\boldsymbol{x}, \quad \boldsymbol{k} \in \mathcal{I}_j, \boldsymbol{e} \in E^*, \ \text{for} \ |\boldsymbol{\alpha}| < M, \tag{3.27}$$

where M is some integer and where we have used standard multi–index notation for $\boldsymbol{\alpha} \in \mathbb{Z}^d_+$, i.e., $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_d$ and $\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1}\cdots x_d^{\alpha_d}$. First note that

$$\int_\Omega \breve{\psi}_{j,\boldsymbol{e},\boldsymbol{k}}(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{|V_{j,\boldsymbol{k}}|}\sum_{\boldsymbol{e}'\in E}(-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'}|V_{j+1,2\boldsymbol{k}+\boldsymbol{e}'}|$$

which is generally different from zero unless the cells all have equal volume. Thus strictly speaking one even looses first order vanishing moments due to the coordinate transformation τ so that (3.27) entails

$$M_d := \binom{M+d-1}{d}$$

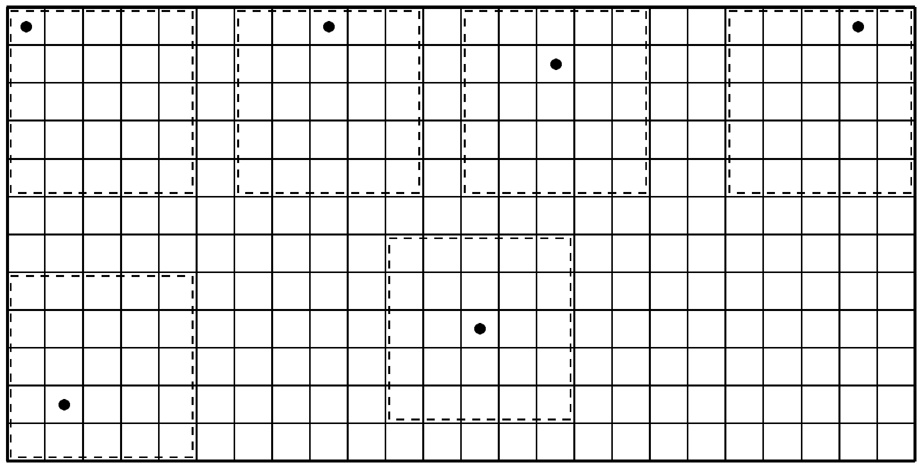

Figure 1: Illustration of the sets $D^s_{j,\boldsymbol{k}}$

conditions. We will use the degrees of freedom offered by $\boldsymbol{L}_j$ to satisfy (3.27) for any given M.

On the other hand, we would like to keep the $\boldsymbol{L}_j$ as sparse as possible so that the efficiency of the corresponding multiscale transformations does not degrade too much. In particular, this means that one should have $L^{j,\boldsymbol{e}}_{\boldsymbol{l},\boldsymbol{k}} = 0$ for indices $\boldsymbol{l}$ for which $|\boldsymbol{k} - \boldsymbol{l}|_\infty$ exceeds a given bound. Moreover, in order to avoid any a–priori directional bias the $L^{j,\boldsymbol{e}}_{\boldsymbol{l},\boldsymbol{k}}$ should differ from zero only in some possibly symmetric neighborhood of $\boldsymbol{k}$, except of course near boundaries where one–sided stencils may be necessary. These requirements can be formalized as follows. First let $s := \min\{r \in \mathbb{N}_0 : (2r+1)^d > M_d\}$ to ensure the availability of sufficiently many parameters in the local index sets specifying possible non–zero entries of $L^{j,\boldsymbol{e}}_{\boldsymbol{l},\boldsymbol{k}}$ to be described next. Setting

$$l_{i,0}(j,\boldsymbol{k}) := \begin{cases} 0, & 0 \le k_i \le s-1, \\ k_i - s, & s \le k_i \le N_{j,i} - 1 - s, \\ N_{j,i} - 1 - 2s, & N_{j,i} - s \le k_i \le N_{j,i} - 1, \end{cases} \qquad 1 \le i \le d,$$

marks the left end point in the ith coordinate direction of consecutive $2s+1$ indices centered symmetrically around the ith coordinate of $\boldsymbol{k}$ as long as the range of indices $\mathcal{I}_j$ permits. Near the boundaries of $\mathcal{I}_j$ one has to move inwards. The local index sets permitting non–zero entries of $L^{j,\boldsymbol{e}}_{\boldsymbol{l},\boldsymbol{k}}$ are then given by

$$D^s_{j,\boldsymbol{k}} := \times_{i=1}^d \{l_{i,0}(j,\boldsymbol{k}), \ldots, l_{i,0}(j,\boldsymbol{k}) + 2s\},$$

i.e.,

$$L^{j,\boldsymbol{e}}_{\boldsymbol{l},\boldsymbol{k}} = 0, \quad \boldsymbol{k} \in \mathcal{I}_j,\ \boldsymbol{l} \in \mathcal{I}_j \setminus D^s_{j,\boldsymbol{k}}, \tag{3.28}$$

see Figure 1 for an illustration in the bivariate case $d = 2$. The dots indicate the positions of $\boldsymbol{k}$ and the dotted lines border the stencils.

In order to determine next the nontrivial entries $L^{j,\boldsymbol{e}}_{\boldsymbol{l},\boldsymbol{k}}$ of $\boldsymbol{L}_j$ we invoke (3.26) to rewrite (3.27) as

$$0 \;=\; \int_\Omega \boldsymbol{x}^{\boldsymbol{\alpha}} \psi_{j,\boldsymbol{e},\boldsymbol{k}}(\boldsymbol{x}) d\boldsymbol{x} = \sum_{\boldsymbol{l} \in \mathcal{I}_{j+1}} m^{j,\boldsymbol{e}}_{\boldsymbol{l},\boldsymbol{k}} \int_\Omega \boldsymbol{x}^{\boldsymbol{\alpha}} \varphi_{j+1,\boldsymbol{l}}(\boldsymbol{x}) d\boldsymbol{x}$$

$$= \sum_{\boldsymbol{l}\in\mathcal{I}_{j+1}} \frac{m^{j,\boldsymbol{e}}_{\boldsymbol{l},\boldsymbol{k}}}{|V_{j+1,\boldsymbol{l}}|} \int_{V_{j+1,\boldsymbol{l}}} \boldsymbol{x}^{\boldsymbol{\alpha}} d\boldsymbol{x}, \tag{3.29}$$

where we have inserted (3.3) in the last step. Defining

$$\nu^j_{\boldsymbol{\alpha},\boldsymbol{l}} := \int_{V_{j,\boldsymbol{l}}} \boldsymbol{x}^{\boldsymbol{\alpha}} d\boldsymbol{x} = \sum_{\boldsymbol{e}\in E} \nu^{j+1}_{\boldsymbol{\alpha},2\boldsymbol{l}+\boldsymbol{e}},$$

(3.24) yields the linear system

$$\sum_{\boldsymbol{l}\in\mathcal{I}_j} \frac{\nu^j_{\boldsymbol{\alpha},\boldsymbol{l}}}{|V_{j,\boldsymbol{l}}|} L^{j,\boldsymbol{e}}_{\boldsymbol{l},\boldsymbol{k}} = -\frac{1}{|V_{j,\boldsymbol{k}}|} \sum_{\boldsymbol{e}'\in E} (-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'} \nu^{j+1}_{\boldsymbol{\alpha},2\boldsymbol{k}+\boldsymbol{e}'}, \quad |\boldsymbol{\alpha}| < M, \boldsymbol{e} \in E^*.$$

By (3.28) this becomes

$$\sum_{\boldsymbol{l}\in D^s_{j,\boldsymbol{k}}} \frac{\nu^j_{\boldsymbol{\alpha},\boldsymbol{l}}}{|V_{j,\boldsymbol{l}}|} L^{j,\boldsymbol{e}}_{\boldsymbol{l},\boldsymbol{k}} = -\frac{1}{|V_{j,\boldsymbol{k}}|} \sum_{\boldsymbol{e}'\in E} (-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'} \nu^{j+1}_{\boldsymbol{\alpha},2\boldsymbol{k}+\boldsymbol{e}'}, \quad |\boldsymbol{\alpha}| < M, \boldsymbol{e} \in E^*. \tag{3.30}$$

Since the system (3.30) is underdetermined we employ the pseudo–inverse to determine the least squares solution of minimal Euclidean norm. This has to be done only once in a preprocessing step. We could have also required that (3.29) holds for all polynomials of *coordinatewise degree* at most $M-1$ which would yield a square system of linear equations. Since this system turned out to be extremely ill conditioned, i.e., numerically singular we have dispensed with this option.

Recall that $\breve{\boldsymbol{G}}^T_{j,0}$ played the role of the refinement matrix for the dual functions $\tilde{\varphi}_{j,k}$ from (3.20). The change of stable completions described by (3.23) modifies these matrices. On uniform Cartesian grids one can choose the matrices $\boldsymbol{L}_j$ such that the $\boldsymbol{G}_{j,0}$ given by (3.23) are the refinement matrices for *higher order and more regular* dual functions. In the general case of curvilinear grids one should still think of the new $\boldsymbol{G}_{j,0}$ as refinement matrices for higher order dual functions although it may be generally hard to identify them explicitly. Therefore it is important to note that this dual system is *never* used explicitly in our algorithms.

We conclude this section with some comments on the modified multiscale transformation. One step in the pyramid schemes of the type (3.21) and (3.22) is given by

$$\bar{\boldsymbol{v}}_j = \boldsymbol{M}^T_{j,0}\bar{\boldsymbol{v}}_{j+1}, \quad \bar{\boldsymbol{d}}_j = \boldsymbol{M}^T_{j,1}\bar{\boldsymbol{v}}_{j+1}, \quad \bar{\boldsymbol{v}}_{j+1} = \boldsymbol{G}^T_{j,0}\bar{\boldsymbol{v}}_j + \boldsymbol{G}^T_{j,1}\bar{\boldsymbol{d}}_j, \tag{3.31}$$

for $\boldsymbol{M}_j, \boldsymbol{G}_j$ given by (3.23) and we set as usual

$$\boldsymbol{M}_{j,1} = (\boldsymbol{M}_{j,\boldsymbol{e}})_{\boldsymbol{e}\in E^*}, \quad \bar{\boldsymbol{d}}_j = (\bar{\boldsymbol{d}}_{j,\boldsymbol{e}})_{\boldsymbol{e}\in E^*}. \tag{3.32}$$

Recall from (3.17) that the original completions $\breve{\boldsymbol{M}}_{j,\boldsymbol{e}}$ were very sparse while the modifications (3.23) will certainly cause some fill–in thereby degrading the efficiency of the multiscale transformation. However, instead of assembling beforehand the matrices $\boldsymbol{M}_{j,1}$ according to (3.23) one can factor the new transformation as follows, (see also [Swe])

$$\boldsymbol{c}_j = \boldsymbol{M}^T_{j,0}\boldsymbol{c}_{j+1}, \quad \boldsymbol{d}_j = \boldsymbol{L}^T_j\boldsymbol{c}_{j+1} + \breve{\boldsymbol{M}}^T_{j,1}\boldsymbol{c}_{j+1}. \tag{3.33}$$

Thus in addition to the previous step from (3.21) one only has to apply $\boldsymbol{L}_j^T$ which is much less expensive than a global application of $\boldsymbol{M}_{j,1}^T$. Similarly the converse transformation consists of the operations

$$\boldsymbol{c}_{j+1} = \breve{\boldsymbol{G}}_{j,0}^T \boldsymbol{c}_j + \boldsymbol{G}_{j,1}^T(\boldsymbol{d}_j - \boldsymbol{L}_j^T \boldsymbol{c}_j) = \breve{\boldsymbol{G}}_{j,0}^T \boldsymbol{c}_j + \breve{\boldsymbol{G}}_{j,1}^T(\boldsymbol{d}_j - \boldsymbol{L}_j^T \boldsymbol{c}_j), \tag{3.34}$$

which up to a separate application of $\boldsymbol{L}_j^T$ also reduces to the original steps in (3.22).

At this point one should note that Harten's discrete multiresolution concept in its general form imposes conditions on its ingredients that have to be verified in each realization. The present combination of stable completions with the concept of vanishing moments offers a concrete strategy for realizing such ingredients. Our concept facilitates to improve and control the compression effect which may not be obtained from the completely discrete point of view alone.

4 The Multiresolution Scheme

The starting point for the multiresolution scheme is the finite volume scheme (2.12) for the arrays $\bar{\boldsymbol{v}}_L^n$ of approximate cell averages on the finest level L.

$$\bar{\boldsymbol{v}}_L^{n+1} = \bar{\boldsymbol{v}}_L^n - \boldsymbol{\Lambda}_L \Delta \bar{\boldsymbol{F}}_L^n \equiv E_L \bar{\boldsymbol{v}}_L^n, \tag{4.1}$$

where we use the abbreviation

$$\Delta \bar{\boldsymbol{F}}_L^n := \{\Delta \bar{F}_{L,\boldsymbol{k}}^n\}_{\boldsymbol{k}\in\mathcal{I}_L}, \quad \Delta \bar{F}_{L,\boldsymbol{k}}^n := \sum_{m=1}^{d} \left(\bar{F}_{m,L,\boldsymbol{k}+\boldsymbol{e}_m}^n - \bar{F}_{m,L,\boldsymbol{k}}^n\right) \tag{4.2}$$

to denote the *flux balances*. The remaining quantities are defined in complete analogy to (2.11) where the grid level is indicated by the additional index L and more generally by $j \le L$ for lower level discretizations.

4.1 Multiscale Transformation and Thresholding

Our objective is to apply the multiscale transformation T_L defined by (3.21) but with $\breve{\boldsymbol{M}}_{j,1}$ replaced by the *new* stable completions $\boldsymbol{M}_{j,1}$ defined by (3.23). Thus, upon applying T_L, one obtains the *multiscale representations* of the averages of the conservative variables

$$\bar{\boldsymbol{v}}_{MS}^n := T_L \bar{\boldsymbol{v}}_L^n = \left((\bar{\boldsymbol{v}}_0^n)^T, (\bar{\boldsymbol{d}}_0^n)^T, \ldots, (\bar{\boldsymbol{d}}_{L-1}^n)^T\right)^T.$$

Recall that the arrays $\bar{\boldsymbol{d}}_j^n$ consist of entries which can be viewed as inner products of the conservative variable with the new complement basis functions $\psi_{j,\boldsymbol{e},\boldsymbol{k}}$ from (3.26). Since they have been arranged to have higher order vanishing polynomial moments the same argument as in (3.13) combined with a Whitney–type estimate (see e.g. [DL93]) yields

$$|(v, \psi_{j,\boldsymbol{e},\boldsymbol{k}})_\Omega| \le \inf_{P\in\Pi_M} \|v - P\|_{L_\infty(\operatorname{supp}\psi_{j,\boldsymbol{e},\boldsymbol{k}})} \le c\,\omega_M(v, 2^{-j}, \operatorname{supp}\psi_{j,\boldsymbol{e},\boldsymbol{k}})_\infty,$$

where Π_M is the space of polynomials of total degree $\le M-1$ on $\mathbb{R}^d$ and $\omega_M(v,t,\Omega)_\infty$ is the usual Mth order *modulus of continuity*, i.e.,

$$\omega_M(v,t,\Omega)_\infty := \sup_{|\boldsymbol{h}|_2\le t} \|\Delta_{\boldsymbol{h}}^M v\|_{L_\infty(\Omega_{M,\boldsymbol{h}})}, \quad \Delta_{\boldsymbol{h}} v := v(\cdot + \boldsymbol{h}) - v, \ \Delta_{\boldsymbol{h}}^l v = \Delta_{\boldsymbol{h}}(\Delta_{\boldsymbol{h}}^{l-1} v),$$

and $\Omega_{M,\boldsymbol{h}} := \{\boldsymbol{x} \in \Omega : \boldsymbol{x} + j\boldsymbol{h} \in \Omega, 0 \leq j \leq M\}$. The modulus of continuity is a measure of smoothness. For instance, when v has bounded derivatives of order M it decays like t^M [DL93]. In this case one therefore has

$$|(v, \psi_{j,\boldsymbol{e},\boldsymbol{k}})_\Omega| \leq c2^{-Mj} \max_{|\boldsymbol{\alpha}|=M} \|D^{\boldsymbol{\alpha}} v\|_{L_\infty(\operatorname{supp} \psi_{j,\boldsymbol{e},\boldsymbol{k}})}. \tag{4.3}$$

Therefore the size of the coefficients in the arrays $\bar{\boldsymbol{v}}^n_{MS}$ indicate the local smoothness properties of v. A small entry represents a negligible detail. The idea already put forward by Harten is to replace negligible entries by zero thereby achieving a compression effect. The point of the present construction is that estimates of the type (4.3) allow us to *quantify* this compression effect and to decide what negligible means. The common procedure is as follows. Fixing a vector of level dependent thresholds $\boldsymbol{\varepsilon} = (\varepsilon_0, \ldots, \varepsilon_{L-1})$, we define the truncated arrays

$$\tilde{\boldsymbol{v}}^n_{MS} := thr_{\boldsymbol{\varepsilon}}(\bar{\boldsymbol{v}}^n_{MS}) = \left((\tilde{\boldsymbol{v}}^n_0)^T, (\tilde{\boldsymbol{d}}^n_0)^T, \ldots, (\tilde{\boldsymbol{d}}^n_{L-1})^T\right)^T$$

by

$$\tilde{d}^n_{j,\boldsymbol{e},\boldsymbol{k}} := \begin{cases} \bar{d}^n_{j,\boldsymbol{e},\boldsymbol{k}} & , \quad |\bar{d}^n_{j,\boldsymbol{e},\boldsymbol{k}}| > \varepsilon_j, \\ 0 & , \quad \text{otherwise,} \end{cases} \qquad \boldsymbol{e} \in E^*, \boldsymbol{k} \in \mathcal{I}_j,\ j = 0, \ldots, L-1. \tag{4.4}$$

We will have to analyse later the effect of this perturbation of the form

$$|\bar{d}^n_{j,\boldsymbol{e},\boldsymbol{k}} - \tilde{d}^n_{j,\boldsymbol{e},\boldsymbol{k}}| \leq \varepsilon_j. \tag{4.5}$$

A typical choice of the level dependent thresholds is

$$\varepsilon_j := a^{-(L-j)}\varepsilon, \tag{4.6}$$

where $a > 1$ and $\varepsilon > 0$ are fixed parameters.

Defining now by

$$\mathcal{I}^L := \times_{j=0}^{L-1} \{(j, \boldsymbol{k}) : \boldsymbol{k} \in \mathcal{I}_j\}$$

the set of *all* multilevel–indices, the set

$$\mathcal{D}^n := \{(j, \boldsymbol{k}) \in \mathcal{I}^L : \exists \boldsymbol{e} \in E^* \text{ such that } |\bar{d}^n_{j,\boldsymbol{e},\boldsymbol{k}}| > \varepsilon_j\}$$

identifies the set of *significant* coefficients (relative to $\boldsymbol{\varepsilon}$).

Note that when determining significant indices we do not distinguish between the different types $\boldsymbol{e} \in E^*$. For some pair $(j, \boldsymbol{k})$ to qualify it suffices that the threshold is exceeded for only *one* $\boldsymbol{e} \in E^*$.

Of course, eventually one would like to take advantage of the compression gained from thresholding the multiscale representations by evolving only the significant part in time. However, this is justified only when being sure that the insignificant part at time level n is not excited too much. At this point one can exploit the finite speed of propagation in hyperbolic conservation laws. In fact, if the CFL condition is satisfied we know that $\bar{v}^n_{j,\boldsymbol{k}}$ influences at most the data $\bar{v}^{n+1}_{j,\boldsymbol{k}+\boldsymbol{e}}$, $\boldsymbol{e} \in \{-1, 0, 1\}^d$, at time level $n+1$. Thus choosing for some fixed integer q the *q-neighborhood*

$$\mathcal{N}^q_{j,\boldsymbol{k}} := \{\boldsymbol{l} \in \mathcal{I}_j : |\boldsymbol{l} - \boldsymbol{k}|_\infty \leq q\},$$

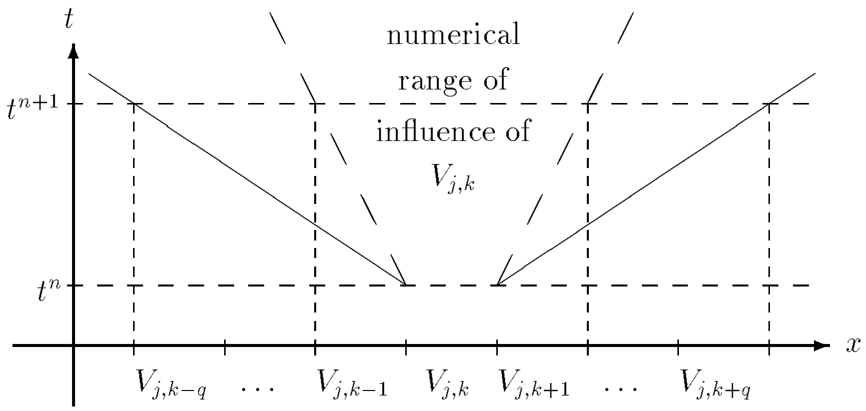

Figure 2: Propagation of information

the set

$$\tilde{\mathcal{D}}^{n+1} := \{(j,\boldsymbol{k}) \in \mathcal{I}^L : \exists \boldsymbol{e} \in E^*, \boldsymbol{l} \in \mathcal{N}^q_{j,\boldsymbol{k}} \text{ s.t. } |\bar{d}^n_{j,\boldsymbol{e},\boldsymbol{k}}| > \varepsilon_j\} \tag{4.7}$$

is guaranteed to cover the significant part of the data at time level $n+1$ and satisfies

$$\mathcal{D}^n \cup \mathcal{D}^{n+1} \subseteq \tilde{\mathcal{D}}^{n+1} \tag{4.8}$$

for q large enough. This means that $\{(j,\boldsymbol{k}) \in \mathcal{I}^L : \boldsymbol{k} \in \mathcal{N}^q_{j,\boldsymbol{l}}\} \subset \tilde{\mathcal{D}}^{n+1}$ for each index $(j,\boldsymbol{l}) \in \mathcal{I}^L$ such that for some $\boldsymbol{e} \in E^*$ one has $|\bar{d}^n_{j,\boldsymbol{e},\boldsymbol{l}}| > \varepsilon_j$. The propagation of information in the case $d = 1$ is illustrated in Figure 2.

4.2 Numerical Fluxes

Eventually we will have to study the time evolution of the multiscale coefficients. This requires numerical fluxes on intermediate levels. It has already been observed by Harten that in the present context $\bar{F}^n_{m,j,\boldsymbol{k}} := \bar{F}^n_m(\bar{v}^n_{j,\boldsymbol{i}} : \boldsymbol{i} \in S_{m,j,\boldsymbol{k}})$ for suitable stencils $S_{m,j,\boldsymbol{k}} \subset \mathcal{I}_j$ would not be an appropriate choice for $j < L$. Instead at each instance one has to employ fluxes that result from combining flux values on the *finest grid*. To organize this we have to take a closer look at the cell boundaries. At this point we take advantage of the particular structure of the cells $V_{j,\boldsymbol{k}}$ being combinatorially equivalent to a cube. Thus the boundary of $V_{j,\boldsymbol{k}}$ consists of pairs of adjacent facets

$$\partial V_{j,\boldsymbol{k}} = \bigcup_{m=1}^{d} (\Gamma_{m,j,\boldsymbol{k}} \cup \Gamma_{m,j,\boldsymbol{k}+\boldsymbol{e}_m}),$$

where we recall from Section 2 that for $\boldsymbol{k} \in \mathcal{I}^m_j$, $m = 1, \dots d$, $j = 0, \dots, L$,

$$\Gamma_{m,j,\boldsymbol{k}} = \{\boldsymbol{x} = \boldsymbol{\tau}(\boldsymbol{\xi}) : k_i \le \xi_i/h_{j,i} \le k_i + 1,\ i \ne m, \xi_m = k_m h_{j,m}\}.$$

Here the index (coordinatewise) sets $\mathcal{I}^m_j$ are given by

$$\mathcal{I}^m_j := \times_{i=1}^{m-1} \mathcal{I}_{j,i} \times \bar{\mathcal{I}}_{j,m} \times \times_{i=m+1}^{d} \mathcal{I}_{j,i},$$

with

$$\mathcal{I}_{j,i} := \{0, 1, \dots, N_{j,i} - 1\}, \quad \bar{\mathcal{I}}_{j,i} := \{0, 1, \dots, N_{j,i}\}.$$

It will be important to use the fact that cell facets on lower levels are unions of cell facets from higher levels. To make this concrete let us introduce for $\boldsymbol{e} \in E$ the sets

$$E_{\boldsymbol{e}} := \{\boldsymbol{e}' \in E : \boldsymbol{e} \cdot \boldsymbol{e}' = 0\}.$$

In other words $E_{\boldsymbol{e}}$ consists of those elements in E whose support is disjoint from that of $\boldsymbol{e}$. In the special case $\boldsymbol{e} = \boldsymbol{e}_m$ of coordinate vectors we briefly write $E_m := E_{\boldsymbol{e}_m}$. Geometric considerations immediately yield then

$$\Gamma_{m,j,\boldsymbol{k}} = \bigcup_{\boldsymbol{e} \in E_m} \Gamma_{m,j+1,2\boldsymbol{k}+\boldsymbol{e}},$$

which motivates the recursive defintion

$$\bar{F}^n_{m,L,\boldsymbol{k}} := \bar{F}_m(\bar{v}^n_{L,\boldsymbol{i}} : \boldsymbol{i} \in S_{m,L,\boldsymbol{k}}), \tag{4.9}$$

and

$$\bar{F}^n_{m,j,\boldsymbol{k}} := \sum_{\boldsymbol{e} \in E_m} \bar{F}^n_{m,j+1,2\boldsymbol{k}+\boldsymbol{e}}, \quad j = L-1, L-2, \ldots, 0. \tag{4.10}$$

Remark 4.1 *Hence the calculation of $\bar{F}^n_{m,j,\boldsymbol{k}}$ requires the evaluation of $2^{(L-j)(d-1)}$ fine level fluxes (4.9).*

In fact, recurring (4.9) provides

$$\bar{F}^n_{m,j,\boldsymbol{k}} = \sum_{\boldsymbol{e} \in E^{L-j}_m} \bar{F}^n_{m,L,2^{L-j}\boldsymbol{k}+\boldsymbol{e}}, \tag{4.11}$$

where

$$E^{L-j}_m = \{\boldsymbol{e} \in \mathbb{Z}^d : 0 \leq e_i < 2^{L-j},\ i \neq m, e_m = 0\}$$

extends the above definition of $E_m = E^1_m$.

When moving up to higher discretization levels it is of course not sufficient to decompose the coarse grid facets but one has to include also new interfaces that split coarse level cells. However, when working with *flux balances* (4.2) this difficulty can be avoided to some extent. In analogy to (4.2) we define the flux balances on coarser levels by

$$\Delta\bar{F}^n_{j,\boldsymbol{k}} := \sum_{m=1}^{d} \left(\bar{F}^n_{m,j,\boldsymbol{k}+\boldsymbol{e}_m} - \bar{F}^n_{m,j,\boldsymbol{k}}\right), \quad \boldsymbol{k} \in \mathcal{I}_j, \tag{4.12}$$

which requires evaluating $(2d)2^{(L-j)(d-1)}$ fine level fluxes (4.9). Noting that

$$E \setminus E_m = E_m + \boldsymbol{e}_m := \{\boldsymbol{e} + \boldsymbol{e}_m : \boldsymbol{e} \in E_m\},$$

one derives the two–scale relation

$$\Delta\bar{F}^n_{j,\boldsymbol{k}} = \sum_{\boldsymbol{e} \in E} \Delta\bar{F}^n_{j+1,2\boldsymbol{k}+\boldsymbol{e}}, \tag{4.13}$$

which is analogous to the second relation in (3.1) and benefits from the fact that fluxes across interior facets cancel.

$$\begin{tikzcd}
\bar{\boldsymbol{v}}^n_{MS} \arrow[r, "E_{MS}"] \arrow[d, "T_L^{-1}"'] & \bar{\boldsymbol{v}}^{n+1}_{MS} \\
\bar{\boldsymbol{v}}^n_L \arrow[r, "E_L"'] & \bar{\boldsymbol{v}}^{n+1}_L \arrow[u, "T_L"']
\end{tikzcd}$$

Figure 3: Schematic representation of E_{MS}

4.3 The Multiscale Scheme

With the above prerequisits at hand we are now ready to describe a multiscale version E_{MS} of the evolution operator E_L given by (4.1). This representation will then allow us to reduce the number of expensive flux evaluations. Conceptually E_{MS} is given by

$$E_{MS} := T_L E_L T_L^{-1}, \tag{4.14}$$

where T_L is the multiscale transformation associated with the new stable completions derived above. This is illustrated in Figure 3.

Of course, the definition (4.14) is only a formal one since the application of E_L would require *all* flux evaluations. The key is therefore to derive an alternative representation that permits savings on the flux evaluations. This will be based on the derivation of evolution equations for the coarser cell averages as well as for the detail coefficients produced by the multiscale transformation.

Lemma 4.1 *For $j = L, L-1, \ldots, 0$, one has*

$$\bar{\boldsymbol{v}}_j^{n+1} = \bar{\boldsymbol{v}}_j^n - \boldsymbol{\Lambda}_j^n \Delta \bar{\boldsymbol{F}}_j^n, \tag{4.15}$$

where

$$\boldsymbol{\Lambda}_j^n := \operatorname{diag}\,(\lambda_{j,\boldsymbol{k}}^n)_{\boldsymbol{k}\in\mathcal{I}_j}, \qquad \lambda_{j,\boldsymbol{k}}^n := \frac{\Delta t^n}{|V_{j,\boldsymbol{k}}|}. \tag{4.16}$$

Proof: For $j = L$ (4.15) agrees with (4.1). Suppose (4.15) has been verified for $j+1$. Then by (3.33) the coarse cell averages on level j are given by

$$\bar{\boldsymbol{v}}_j^{n+1} = \boldsymbol{M}_{j,0}^T \bar{\boldsymbol{v}}_{j+1}^{n+1} = \boldsymbol{M}_{j,0}^T \left(\bar{\boldsymbol{v}}_{j+1}^n - \boldsymbol{\Lambda}_{j+1}^n \Delta \bar{\boldsymbol{F}}_{j+1}^n\right) = \bar{\boldsymbol{v}}_j^n - \boldsymbol{M}_{j,0}^T \boldsymbol{\Lambda}_{j+1}^n \Delta \bar{\boldsymbol{F}}_{j+1}^n.$$

Recalling the definition (3.14) of $\boldsymbol{M}_{j,0}$ and the flux (balance) representations (4.10), (4.12), we see that (4.16) yields

$$\left(\boldsymbol{M}_{j,0}^T \boldsymbol{\Lambda}_{j+1}^n \Delta \bar{\boldsymbol{F}}_{j+1}^n\right)_{\boldsymbol{k}} = \left(\boldsymbol{\Lambda}_j^n \Delta \bar{\boldsymbol{F}}_j^n\right)_{\boldsymbol{k}} = \lambda_{j,\boldsymbol{k}}^n \sum_{m=1}^d \left(\bar{\boldsymbol{F}}_{m,j,\boldsymbol{k}+\boldsymbol{e}_m}^n - \bar{\boldsymbol{F}}_{m,j,\boldsymbol{k}}^n\right)_{\boldsymbol{k}},$$

which completes the proof. $\square$

Note that (4.15) is *not* just the original finite volume scheme applied to level j instead of L since the involved fluxes always refer to data on the finest level L (4.10). Nevertheless, the scheme is still in conservative form with respect to level j. Moreover, since

$$\lambda_{j,\boldsymbol{k}} = \frac{\Delta t^n}{|V_{j,\boldsymbol{k}}|} < \frac{\Delta t^n}{|V_{j+1,2\boldsymbol{k}+\boldsymbol{e}}|} = \lambda_{j+1,2\boldsymbol{k}+\boldsymbol{e}}, \quad \boldsymbol{e} \in E, \tag{4.17}$$

the CFL condition holds whenever it holds for the finest level L. We will take up the implications of this fact later on in Section 6.

It remains to establish the evolution equations for the detail coefficients. To derive its ingredients it will be convenient to employ the vector description. To this end, let

$$\boldsymbol{V}_j := \operatorname{diag}\left(|V_{j,\boldsymbol{k}}|\right)_{\boldsymbol{k}\in\mathcal{I}_j} = \Delta t^n (\boldsymbol{\Lambda}_j^n)^{-1},$$

and consider the submatrices

$$\boldsymbol{V}_{j+1,\boldsymbol{e}} := \operatorname{diag}\left(|V_{j+1,2\boldsymbol{k}+\boldsymbol{e}}|\right)_{\boldsymbol{k}\in\mathcal{I}_j}.$$

Similarly we decompose the matrices $\boldsymbol{M}_{j,\boldsymbol{e}}$ into the blocks

$$\boldsymbol{M}_{j,\boldsymbol{e},\boldsymbol{e}'} := \left(m^{j,\boldsymbol{e}}_{2\boldsymbol{l}+\boldsymbol{e}',\boldsymbol{k}}\right)_{\boldsymbol{l},\boldsymbol{k}\in\mathcal{I}_j}.$$

One easily checks that in these terms the relation (3.25) reads

$$\boldsymbol{M}_{j,\boldsymbol{e},\boldsymbol{e}'} \boldsymbol{V}^{-1}_{j+1,\boldsymbol{e}'} = \boldsymbol{M}_{j,\boldsymbol{e},0} \boldsymbol{V}^{-1}_{j+1,0} + \left((-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'} - 1\right) \boldsymbol{V}_j^{-1}. \tag{4.18}$$

In addition we need the following flux quantities

$$\Delta \bar{\boldsymbol{F}}^n_{j+1,\boldsymbol{e}} := \{\Delta \bar{F}^n_{j+1,2\boldsymbol{k}+\boldsymbol{e}}\}_{\boldsymbol{k}\in\mathcal{I}_j}, \quad \boldsymbol{e} \in E,$$

and

$$\Delta \hat{\boldsymbol{F}}^n_{j,\boldsymbol{e}} := \frac{1}{2} \boldsymbol{V}_j \boldsymbol{M}^T_{j,\boldsymbol{e},0} \boldsymbol{V}^{-1}_{j+1,0} \Delta \bar{\boldsymbol{F}}^n_j, \quad \boldsymbol{e} \in E. \tag{4.19}$$

Note that the components of $\Delta \hat{\boldsymbol{F}}^n_{j,\boldsymbol{e}}$ are given by

$$\Delta \hat{F}^n_{j,\boldsymbol{e},\boldsymbol{k}} = \frac{|V_{j,\boldsymbol{k}}|}{2} \sum_{\boldsymbol{l}\in\mathcal{I}_j} \frac{m^{j,\boldsymbol{e}}_{2\boldsymbol{l},\boldsymbol{k}}}{|V_{j+1,2\boldsymbol{l}}|} \Delta \bar{F}^n_{j,\boldsymbol{l}}, \tag{4.20}$$

and (4.13) says that

$$\sum_{\boldsymbol{e}\in E} \Delta \bar{\boldsymbol{F}}^n_{j+1,\boldsymbol{e}} = \Delta \bar{\boldsymbol{F}}^n_j.$$

To describe how these quantities are related to the evolution of the detail coefficients in the arrays $\bar{\boldsymbol{d}}^n_j$ we need for $\boldsymbol{e} \in E^*$ the sets

$$\widehat{E}_{\boldsymbol{e}} := \{\boldsymbol{e}' \in E^* : (-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'} = -1\} = \{\boldsymbol{e}' \in E^* : \boldsymbol{e}\cdot\boldsymbol{e}' = \text{ odd }\}. \tag{4.21}$$

It is clear that

$$\bigcup_{\boldsymbol{e}\in E^*} \widehat{E}_{\boldsymbol{e}} = E^*.$$

Lemma 4.2 *For $j = L, L-1, \ldots, 0$, $\boldsymbol{e} \in E^*$, one has*

$$\bar{\boldsymbol{d}}_{j,\boldsymbol{e}}^{n+1} = \bar{\boldsymbol{d}}_{j,\boldsymbol{e}}^{n} - 2\boldsymbol{\Lambda}_j^n \left(\Delta\hat{\boldsymbol{F}}_{j,\boldsymbol{e}}^n - \sum_{\boldsymbol{e}'\in\widehat{E}_{\boldsymbol{e}}} \Delta\bar{\boldsymbol{F}}_{j+1,\boldsymbol{e}'}^n \right). \tag{4.22}$$

Proof: Combining (3.31), (3.32) with Lemma 4.1, provides

$$\bar{\boldsymbol{d}}_{j,\boldsymbol{e}}^{n+1} = \boldsymbol{M}_{j,\boldsymbol{e}}^T \bar{\boldsymbol{v}}_{j+1}^{n+1} = \boldsymbol{M}_{j,\boldsymbol{e}}^T \left(\bar{\boldsymbol{v}}_{j+1}^n - \boldsymbol{\Lambda}_{j+1}^n \Delta\bar{\boldsymbol{F}}_{j+1}^n \right) = \bar{\boldsymbol{d}}_{j,\boldsymbol{e}}^n - \boldsymbol{M}_{j,\boldsymbol{e}}^T \boldsymbol{\Lambda}_{j+1}^n \Delta\bar{\boldsymbol{F}}_{j+1}^n. \tag{4.23}$$

Employing (4.18), (4.19) and (4.21) above, we conclude that

$$\begin{aligned}
\boldsymbol{M}_{j,\boldsymbol{e}}^T \boldsymbol{\Lambda}_{j+1}^n \Delta\bar{\boldsymbol{F}}_{j+1}^n &= \Delta t^n \boldsymbol{M}_{j,\boldsymbol{e}}^T \boldsymbol{V}_{j+1}^{-1} \Delta\bar{\boldsymbol{F}}_{j+1}^n = \Delta t^n \sum_{\boldsymbol{e}'\in E} \boldsymbol{M}_{j,\boldsymbol{e},\boldsymbol{e}'}^T \boldsymbol{V}_{j+1,\boldsymbol{e}'}^{-1} \Delta\bar{\boldsymbol{F}}_{j+1,\boldsymbol{e}'}^n \\
&= \Delta t^n \sum_{\boldsymbol{e}'\in E} \left(\boldsymbol{M}_{j,\boldsymbol{e},0}^T \boldsymbol{V}_{j+1,0}^{-1} + \left((-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'} - 1 \right) \boldsymbol{V}_j^{-1} \right) \Delta\bar{\boldsymbol{F}}_{j+1,\boldsymbol{e}'}^n \\
&= \Delta t^n \boldsymbol{M}_{j,\boldsymbol{e},0}^T \boldsymbol{V}_{j+1,0}^{-1} \Delta\bar{\boldsymbol{F}}_j^n - \Delta t^n \sum_{\boldsymbol{e}'\in E} \left((-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'} - 1 \right) \boldsymbol{V}_j^{-1} \Delta\bar{\boldsymbol{F}}_{j+1,\boldsymbol{e}'}^n \\
&= 2\boldsymbol{\Lambda}_j^n \left(\Delta\hat{\boldsymbol{F}}_{j,\boldsymbol{e}}^n - \sum_{\boldsymbol{e}'\in\widehat{E}_{\boldsymbol{e}}} \Delta\bar{\boldsymbol{F}}_{j+1,\boldsymbol{e}'}^n \right).
\end{aligned}$$

Inserting this into (4.23), yields (4.22) and finishes the proof. □

We are now prepared to describe the evolution operator E_{MS}.

Coarse Level Flux Balances

By (4.12) the coarse level flux balances are given for $\boldsymbol{k} \in \mathcal{I}_{0,m}$ by

$$\Delta\bar{F}_{0,\boldsymbol{k}}^n = \sum_{m=1}^{d} \sum_{\boldsymbol{r}\in E_m^L} \left(\bar{F}_m(\bar{v}_{L,\boldsymbol{l}}^n : \boldsymbol{l} \in S_{m,L,2^L(\boldsymbol{k}+\boldsymbol{e}_m)+\boldsymbol{r}}) - \bar{F}_m(\bar{v}_{L,\boldsymbol{l}}^n : \boldsymbol{l} \in S_{m,L,2^L\boldsymbol{k}+\boldsymbol{r}}) \right), \tag{4.24}$$

which are used to propagate the coarse level cell averages according to Lemma 4.1.

Evolution of Detail Coefficients

Here we use Lemma 4.2 in combination with compression. In fact, whenever $(j,\boldsymbol{k}) \in \tilde{\mathcal{D}}^{n+1}$ corresponds to a significant (group of) wavelet coefficient(s) we compute according to (4.22)

$$\bar{d}_{j,\boldsymbol{e},\boldsymbol{k}}^{n+1} = \bar{d}_{j,\boldsymbol{e},\boldsymbol{k}}^n - 2\lambda_{j,\boldsymbol{k}}^n \left(\Delta\hat{F}_{j,\boldsymbol{e},\boldsymbol{k}}^n - \sum_{\boldsymbol{e}'\in\widehat{E}_{\boldsymbol{e}}} \Delta\bar{F}_{j+1,2\boldsymbol{k}+\boldsymbol{e}'}^n \right), \quad \boldsymbol{e} \in E^*. \tag{4.25}$$

Thus advancing the detail coefficients in time requires the flux balances on level $j+1$ which according to (4.9), (4.11) and (4.12) are given by

$$\begin{aligned}
\Delta\bar{F}_{j+1,2\boldsymbol{k}+\boldsymbol{e}}^n &= \sum_{m=1}^{d} \sum_{\boldsymbol{r}\in E_m^{L-j-1}} \left(\bar{F}_m(\bar{v}_{L,\boldsymbol{l}}^n : \boldsymbol{l} \in S_{m,L,2^{L-j-1}(2\boldsymbol{k}+\boldsymbol{e}+\boldsymbol{e}_m)+\boldsymbol{r}}) \right. \\
&\qquad \left. - \bar{F}_m(\bar{v}_{L,\boldsymbol{l}}^n : \boldsymbol{l} \in S_{m,L,2^{L-j-1}(2\boldsymbol{k}+\boldsymbol{e})+\boldsymbol{r}}) \right), \quad \boldsymbol{e} \in E^*.
\end{aligned} \tag{4.26}$$

Note that one has to compute $\Delta \bar{F}^n_{j+1,2\boldsymbol{k}+\boldsymbol{e}}$ actually only for $\boldsymbol{e} \in E^*$, say since by (4.13), one has

$$\Delta \bar{F}^n_{j+1,2\boldsymbol{k}} = \Delta \bar{F}^n_{j,\boldsymbol{k}} - \sum_{\boldsymbol{e} \in E^*} \Delta \bar{F}^n_{j+1,2\boldsymbol{k}+\boldsymbol{e}}, \tag{4.27}$$

and $\Delta \bar{F}^n_{j,\boldsymbol{k}}$ has been computed before.

Our objective is to reduce the cost of computing the remaining numerical fluxes appearing in (4.26) by using already computed fluxes from lower levels whenever suggested by the detail coefficients. More precisely, the corresponding decision is based on the sets $\mathcal{D}^{n+1}$:

- If $(j, \boldsymbol{k}) \in \tilde{\mathcal{D}}^{n+1}$ employ the exact formula (4.26).
- If $(j, \boldsymbol{k}) \notin \tilde{\mathcal{D}}^{n+1}$ we replace the exact balances by the quantities

$$\Delta \bar{F}^n_{j+1,2\boldsymbol{k}+\boldsymbol{e}} \longleftarrow \Delta \tilde{F}^n_{j+1,2\boldsymbol{k}+\boldsymbol{e}} := -2^{1-d} \sum_{\boldsymbol{e}' \in E^*} (-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'} \Delta \hat{F}^n_{j,\boldsymbol{e}',\boldsymbol{k}}, \quad \boldsymbol{e} \in E^*, \tag{4.28}$$

  where, in view of (4.27), we set

$$\Delta \bar{F}^n_{j+1,2\boldsymbol{k}} \longleftarrow \Delta \tilde{F}^n_{j+1,2\boldsymbol{k}} := \Delta \bar{F}^n_{j,\boldsymbol{k}} - \sum_{\boldsymbol{e} \in E^*} \Delta \bar{F}^n_{j+1,2\boldsymbol{k}+\boldsymbol{e}}.$$

  Recall that, by (4.19), the quantities $\Delta \bar{F}^n_{j,\boldsymbol{k}}$ are known from the previous level.

We will now justify our choice (4.28). First note that $(j, \boldsymbol{k}) \notin \tilde{\mathcal{D}}^{n+1}$ means that the detail coefficients $\bar{d}^{n+1}_{j,\boldsymbol{e},\boldsymbol{k}}$, $\boldsymbol{e} \in E^*$, are negligible and should be replaced by zero

$$\bar{d}^{n+1}_{j,\boldsymbol{e},\boldsymbol{k}} := 0, \quad \boldsymbol{e} \in E^*. \tag{4.29}$$

This in turn suggest that the expression in paranthesis on the right hand side of (4.25) should vanish since by definition of $\tilde{\mathcal{D}}^{n+1}$ $\bar{d}^n_{j,\boldsymbol{e},\boldsymbol{k}}$ is also negligible. Thus we seek quantities $\Delta \tilde{F}^n_{j+1,2\boldsymbol{k}+\boldsymbol{e}}$ satisfying

$$\sum_{\boldsymbol{e}' \in \widehat{E}_{\boldsymbol{e}}} \Delta \tilde{F}^n_{j+1,2\boldsymbol{k}+\boldsymbol{e}'} = \Delta \hat{F}^n_{j,\boldsymbol{e},\boldsymbol{k}}, \qquad \boldsymbol{e} \in E^*,$$

which is a $(2^d-1) \times (2^d-1)$ linear system of equations. The fact that this system has a unique solution given by (4.28) is an immediate consequence of the following lemma.

Lemma 4.3 *For $\widehat{E}_{\boldsymbol{e}}$ defined by (4.21) and any $\boldsymbol{e} \in E^*$ the following statements hold.*

(i) The cardinality of $\widehat{E}_{\boldsymbol{e}}$ is always 2^{d-1}.

(ii) For any $\boldsymbol{e}'' \in E^* \setminus \{\boldsymbol{e}\}$ one has $\sum_{\boldsymbol{e}' \in \widehat{E}_{\boldsymbol{e}}} (-1)^{\boldsymbol{e}'\cdot\boldsymbol{e}''} = 0$.

(iii) The $(2^d-1) \times (2^d-1)$ matrix $\boldsymbol{A}$ be defined by

$$\boldsymbol{A}_{\boldsymbol{e},\boldsymbol{e}'} = \begin{cases} 1 & , \text{ if } \boldsymbol{e}' \in \widehat{E}_{\boldsymbol{e}} \\ 0 & , \text{ otherwise} \end{cases} \tag{4.30}$$

is always nonsingular and its inverse is given by $\boldsymbol{A}^{-1} = -2^{-(d-1)} \left((-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'}\right)_{\boldsymbol{e},\boldsymbol{e}' \in E^*}$.

Proof: Let us denote by $s_{\boldsymbol{e}}$ the support of $\boldsymbol{e} \in E^*$, i.e., the indices $i_l \in \{1,\dots,d\}$ for which $e_{i_l} = 1$ and let $m_{\boldsymbol{e}} := |\boldsymbol{e}| = \# s_{\boldsymbol{e}}$. Since the entries in the elements of $\widehat{E}_{\boldsymbol{e}}$ can take any values outside $s_{\boldsymbol{e}}$ while otherwise being supported in any subset of $s_{\boldsymbol{e}}$ of odd cardinality, we immdediately conclude that

$$\#\widehat{E}_{\boldsymbol{e}} = 2^{d-m_{\boldsymbol{e}}} \left(\sum_{2i+1 \le m_{\boldsymbol{e}}} \binom{m_{\boldsymbol{e}}}{2i+1} \right) = 2^{d-m_{\boldsymbol{e}}} 2^{m_{\boldsymbol{e}}-1},$$

which proves (i).

As for (ii) note first that, by definition of $\widehat{E}_{\boldsymbol{e}}$, one has $(-1)^{\boldsymbol{e}'\cdot\boldsymbol{e}''} = -(-1)^{\boldsymbol{e}'\cdot(\boldsymbol{e}''-\boldsymbol{e})}$ Hence we may assume without loss of generality that $s_{\boldsymbol{e}} \cap s_{\boldsymbol{e}''} = \emptyset$. Let us therefore split $\boldsymbol{e}' \in \widehat{E}_{\boldsymbol{e}}$ as $\boldsymbol{e}' = \boldsymbol{e}_1' + \boldsymbol{e}_2'$ where $s_{\boldsymbol{e}_2'} \subseteq s_{\boldsymbol{e}''}$. Therefore the entries of $\boldsymbol{e}_2'$ can take any values in $\{0,1\}$. Hence

$$\sum_{\boldsymbol{e}' \in \widehat{E}_{\boldsymbol{e}}} (-1)^{\boldsymbol{e}'\cdot\boldsymbol{e}''} = \sum_{\boldsymbol{e}_1' \in \widehat{E}_{\boldsymbol{e}}, s_{\boldsymbol{e}_1'} \cap s_{\boldsymbol{e}''} = \emptyset} \left(\sum_{s_{\boldsymbol{e}_2'} \subseteq s_{\boldsymbol{e}''}} (-1)^{\boldsymbol{e}''\cdot\boldsymbol{e}_2'} \right).$$

Since the number of subsets of $s_{\boldsymbol{e}''}$ of odd support equals the number of subsets of even support the sum in parentheses on the right hand side of the above relation vanishes which confirms (ii).

Concerning (iii), consider the $(2^d-1)\times(2^d-1)$ matrix $\boldsymbol{B} := \left((-1)^{\boldsymbol{e}\cdot\boldsymbol{e}'}\right)_{\boldsymbol{e},\boldsymbol{e}' \in E^*}$ By definition (4.30) of $\boldsymbol{A}$, (i) and (ii), one has

$$(\boldsymbol{AB})_{\boldsymbol{e},\boldsymbol{e}''} = \sum_{\boldsymbol{e}' \in \widehat{E}_{\boldsymbol{e}}} (-1)^{\boldsymbol{e}'\cdot\boldsymbol{e}''} = \begin{cases} 0 & \text{if } \boldsymbol{e} \neq \boldsymbol{e}'', \\ -2^{d-1} & \text{if } \boldsymbol{e} = \boldsymbol{e}'', \end{cases}$$

which means that $\boldsymbol{A}^{-1} = -2^{-(d-1)}\boldsymbol{B}$, completing the proof. $\Box$.

Finally, the outline of the multiscale scheme, in the following denoted by $\tilde{E}_{MS}$, can be summarized as follows:

1.) Thresholding detail coefficients:

- Threshold the coefficients $\bar{d}^n_{j,\boldsymbol{e},\boldsymbol{k}}$ according to (4.4);
- Determine the predicted set of significant detail coefficients $\tilde{\mathcal{D}}^{n+1}$ according to (4.7).

2.) Update of single–scale coefficients on the coarsest level:

- For $\boldsymbol{k} \in \mathcal{I}_0$ compute $\Delta \bar{F}^n_{0,\boldsymbol{k}}$ according to (4.24);
- For $\boldsymbol{k} \in \mathcal{I}_0$ compute $\bar{v}^{n+1}_{0,\boldsymbol{k}}$ according to (4.15);

3.) Update of detail coefficients for $j = 0,1,\dots,L-1$:

- For $\boldsymbol{k} \in \mathcal{I}_j$, $\boldsymbol{e} \in E^*$ compute $\Delta \bar{F}^n_{j+1,2\boldsymbol{k}+\boldsymbol{e}}$ according to (4.26), if $(j,\boldsymbol{k}) \in \tilde{\mathcal{D}}^{n+1}$, else according to (4.28).
- For $\boldsymbol{k} \in \mathcal{I}_j$ compute $\Delta \bar{F}^n_{j+1,2\boldsymbol{k}}$ according to (4.27);

4.3.1 Consistency

If the solution is locally constant, i.e., $\bar{v}^n_{L,\boldsymbol{k}} \equiv \bar{v} = const$ one has $\Delta\bar{F}^n_{j,\boldsymbol{k}} \equiv 0$ when the scheme is consistent in the sense of definition (2.10). In this case the quantities $\Delta\hat{F}^n_{j,\boldsymbol{k}}$ also vanish by (4.20) so that $\Delta\tilde{F}^n_{j+1,2\boldsymbol{k}+\boldsymbol{e}} = 0$ by (4.28), which guarantees the *conservation property* on level $j+1$, i.e., the quantity $\sum_{k\in\mathcal{I}_{j+1}} |V_{j+1,k}|\overline{v}^n_{j+1,k}$ is preserved provided that the flow over the boundary of the computational domain is zero. In our numerical computations this actually turned out to be essential. In an earlier attempt to simply transport a tensor product multiscale scheme based on univariate *biorthogonal wavelets* through the mapping τ, see (2.4), the conservation property turned out to be violated and the overall efficiency of that version was by far not competitive.

4.4 Computational Cost

In this subsection, we examine the computational cost for the multiscale scheme. In order to bring out the principle features we confine the discussion to the simple univariate case of scalar conservation laws. In particular, we are interested in the conditions under which it accelerates expensive finite volume schemes. Of course, we can only expect a reduction of the computational cost if the set of significant indices is sufficiently small, i. e., the finest resolution is only needed in a small part of the computational domain. Otherwise one cannot compensate the overhead caused by the multiscale transformation.

In the following examination of the computational cost, we will only take floating point operations (flops) into account. As usual in such kind of analysis, memory access times, assignment of variables as well as decisions in if–statements will be neglected. Consequently, this consideration will only yield an upper bound of the speed–up factor. In practice, this factor is smaller than the theoretical value since, in general, memory access times are higher for adaptive schemes than for non–adaptive ones.

The cost for the evaluation of the numerical flux at one single facet is denoted by C_F. By C_{FV} and C_{MS}, we denote the total computational cost per time step for the underlying finite volume scheme and the multiscale scheme, respectively.

As an example, we consider the one–dimensional case. The cost for the algorithm without multiscale approach are

$$C_{FV} = (N_L + 1)\, C_F + 3\, N_L.$$

The theoretical speed–up factor is defined by $\sigma_{theoret} := C_{FV}/C_{MS}$ which depends on several parameters such as the number L of refinement levels, the resolution numbers N_0 and N_L and the number N_F of flux evaluations needed for the multiscale scheme. Additionally, we define the ratio $\eta := N_F/N_L^*$ which compares this number to the number of all flux evaluations which have to be performed in the underlying finite volume scheme. Here, the number of flux evaluations for the multiscale scheme is given by

$$N_F = \#\widetilde{\mathcal{D}}^{n+1} + N_0 + 1. \tag{4.31}$$

We assume that the number of non–vanishing entries of the matrices $\boldsymbol{M}^T_{j,0}$, $\breve{\boldsymbol{M}}^T_{j,1}$, $\breve{\boldsymbol{G}}^T_{j,0}$ and $\boldsymbol{G}^T_{j,1}$ is uniformly bounded by κ in each line and that of $\boldsymbol{L}^T_j$ by s. According to (4.20), the number of non–zero coefficients of the flux approximation is also uniformly bounded by s for each (j,k). Thus, the cost for the multiscale scheme are bounded by

$$C_{MS} = K\, N_D + C_F + (N_0 + \#\widetilde{\mathcal{D}}^{n+1})\,(3 + C_F)$$

where $N_D = \sum_{j=0}^{L-1} N_j = N_L - N_0$ and $K = (8\,\kappa + 6\,s - 5)$. An acceleration is achieved if $C_{MS} < C_{FV}$ which leads to

$$\eta < \eta_* := 1 - \frac{K}{3 + C_F}\,\frac{N_D}{N_L + 1}.$$

Then the theoretical speed–up factor is given by

$$\sigma_{theoret} = \frac{3 + \left(1 + \frac{1}{N_L}\right) C_F}{(3 + C_F)\,\frac{N_0 + \#\widetilde{\mathcal{D}}^{n+1}}{N_L} + \left(1 - \frac{N_0}{N_L}\right) K + \frac{C_F}{N_L}}.$$

Now we keep the coarse level resolution N_0 fixed and consider the limiting process $L \to \infty$, i. e., $N_L = 2^L\,N_0 \to \infty$. In the prescence of discontinuities, see e.g. the example in Section 5, the number of significant detail coefficients is expected to become small compared to the number of all coefficients up to the current level of highest resolution. Therefore it is reasonable to assume that in a typical situation

$$\frac{\#\widetilde{\mathcal{D}}^{n+1}}{N_L} \to 0, \qquad L \to \infty, \tag{4.32}$$

since otherwise it does not make sense to apply the multiscale scheme. Consequently, the limit of $\sigma_{theoret}$ for $L \to \infty$ is determined by

$$\sigma_\infty^L := \lim_{L\to\infty} \sigma_{theoret} = \frac{3 + C_F}{K} < \infty.$$

In particular, this means that we *cannot* obtain arbitrarily high speed–up factors when increasing the number L of refinement levels.

Finally, we want to take a different point of view. Sjögreen notices in [Sjö95] for his multiscale approach for a two–dimensional scheme based on dimensional splitting that an acceleration can only be achieved if the highest resolution N_L is very large and L is small. This corresponds to the consideration of the limiting process $N_0 \to \infty$ where we keep $L = const$ such that

$$\zeta := \frac{N_0}{N_L} = const < 1. \tag{4.33}$$

Here, we want to emphasize that this process contradicts inherently the general idea of multiscale methods where a fixed number N_0 is taken and the limit process $L \to \infty$ is considered. In case of relation (4.33) N_0 is proportional to N_L so that as in the above case it makes sense to assume that

$$\frac{\#\widetilde{\mathcal{D}}^{n+1}}{N_0} \to 0, \qquad N_0 \to \infty,$$

instead of (4.32). Based on these assumptions, the limit of $\sigma_{theoret}$ is

$$\sigma_\infty^{N_0} := \lim_{N_0\to\infty} \sigma_{theoret} = \frac{3 + C_F}{(3 + C_F)\,\zeta + (1 - \zeta)\,K}$$

which is also a finite value. The reason that these two limiting values are finite is that the complexity is $\mathcal{O}(N_L)$ for both the finite volume and the multiscale scheme. The difference between the two methods is only that the constant is better for the multiscale scheme. We will briefly discuss some consequences of these observations below in Section 6.

4.5 Stability and Convergence

The above multiscale scheme can only be expected to converge if the underlying finite volume scheme converges which will always be assumed in the following. For sufficient conditions in various special cases such as scalar conservation laws see e.g. [Krö97]. For simplicity we will also assume a uniform temporal step size $\Delta t_L = \Delta t_L^n$ that meets, of course, the CFL condition. Moreover, we consider only consistent schemes E_L which are either linear or l_1–contractive, i.e.,

$$\|E_L \bar{\boldsymbol{v}} - E_L \bar{\boldsymbol{w}}\|_{1,V} \leq \|\bar{\boldsymbol{v}} - \bar{\boldsymbol{w}}\|_{1,V},$$

where $\|\bar{\boldsymbol{v}}\|_{1,V} := \sum_{\boldsymbol{k}\in\mathcal{I}} |V_{\boldsymbol{k}}||\bar{v}_{\boldsymbol{k}}|$. This is the only instance where our restriction to *scalar* conservation laws matters. Under such assumptions Harten already showed convergence of E_{MS} for one–dimensional scalar homogeneous conservation laws provided that the multiscale transformations are *stable* in a certain sense. In this case his reasoning carries over to the multi–dimensional case. As before let T_L denote the transformation that takes an array $\bar{\boldsymbol{w}}_L$ of cell averages on level L into the array $\bar{\boldsymbol{w}}_{MS}$ of multiscale coefficients, see (3.21), (3.33). Since stability of multiscale transformations is better understood in the L_2-setting it is convenient to consider the corresponding rescaled versions T_L° composed of the matrices

$$\boldsymbol{M}_j^\circ := 2^{d/2}\boldsymbol{M}_j, \quad \boldsymbol{G}_j^\circ := 2^{-d/2}\boldsymbol{G}_j,$$

see [Dah96]. T_L° is said to be l_2–stable if the condition numbers $\|T_L^\circ\|\|(T_L^\circ)^{-1}\|$ stay uniformly bounded in L, where $\|\cdot\|$ denotes here the spectral norm. A necessary (and in typical cases also sufficient) condition for l_2–stability is that the underlying wavelet basis has a dual (biorthogonal) basis in $L_2(\Omega)$. Of course, the original basis composed of the $\breve{\psi}_{j,\boldsymbol{e},\boldsymbol{k}}$ has this property because this is essentially the Haar basis. Our modifications through the matrices $\boldsymbol{L}_j$ changed the filters in $\breve{\boldsymbol{G}}_{j,0}$ though. Hence the dual basis (if it existed) would be generated by functions $\tilde{\varphi}_{j,\boldsymbol{k}}$ satisfying the two–scale relations

$$\tilde{\Phi}_j = \boldsymbol{G}_{j,0}\tilde{\Phi}_{j+1}. \tag{4.34}$$

We have never made explicit use of these dual functions in the above algorithm. In fact, strictly speaking we have not proved that solutions to the refinement relation (4.34) exist in $L_2(\Omega)$. Although this is very likely to be the case and can be verified on Cartesian grids it is perhaps not quite obvious for arbitrary transformations τ. We expect though that the choice of $\boldsymbol{L}_j$ in the modified multiscale transformations supports the solvability of (4.34) in L_2 since the vanishing moments of our primal wavelets $\psi_{j,\boldsymbol{e},\boldsymbol{k}}$ means a corresponding order of polynomial exactness of the dual system. At any rate, we feel that it is legitimate to assume l_2–stability of the transformations T_L° at this point.

Proposition 4.1 *If the T_L° are uniformly l_2–stable and $\boldsymbol{\varepsilon}$ is chosen according to (4.6) then $\tilde{\boldsymbol{w}}_L := T_L^{-1}\tilde{\boldsymbol{w}}_{MS}$ for $\tilde{\boldsymbol{w}}_{MS} := thr_{\boldsymbol{\varepsilon}}(\bar{\boldsymbol{w}}_{MS})$ satisfies*

$$\|\bar{\boldsymbol{w}}_L - \tilde{\boldsymbol{w}}_L\|_{1,V_L} = \sum_{\boldsymbol{k}\in\mathcal{I}_L} |\bar{w}_{L,\boldsymbol{k}} - \tilde{w}_{L,\boldsymbol{k}}||V_{L,\boldsymbol{k}}| \ \lesssim\ \varepsilon.$$

Proof: As before let $\bar{\boldsymbol{d}}_j$ denote the array of detail coefficients on level j produced by T_L. The corresponding coefficients generated by the scaled version T_L° are easily seen to be

$$\bar{\boldsymbol{w}}_0^\circ = 2^{Ld/2}\bar{\boldsymbol{w}}_0, \quad \bar{\boldsymbol{d}}_j^\circ = 2^{(L-j)d/2}\bar{\boldsymbol{d}}_j.$$

Thus (4.5) implies that

$$|\bar{d}^{\circ}_{j,\boldsymbol{e},\boldsymbol{k}} - \tilde{d}^{\circ}_{j,\boldsymbol{e},\boldsymbol{k}}| = 2^{(L-j)d/2}|\bar{d}_{j,\boldsymbol{e},\boldsymbol{k}} - \tilde{d}_{j,\boldsymbol{e},\boldsymbol{k}}| \leq 2^{(L-j)d/2}\varepsilon_j.$$

By (3.2) we have

$$\|\bar{\boldsymbol{w}}_L - \tilde{\boldsymbol{w}}_L\|_{1,V_L} \sim 2^{-Ld}\|\bar{\boldsymbol{w}}_L - \tilde{\boldsymbol{w}}_L\|_{l_1},$$

where $\|\cdot\|_{l_1}$ is the usual l_1–norm. Since

$$\|\bar{\boldsymbol{w}}_L - \tilde{\boldsymbol{w}}_L\|_{l_1} \leq N_L^{1/2}\|\bar{\boldsymbol{w}}_L - \tilde{\boldsymbol{w}}_L\|_{l_2} \sim 2^{Ld/2}\|\bar{\boldsymbol{w}}_L - \tilde{\boldsymbol{w}}_L\|_{l_2},$$

we conclude that

$$\|\bar{\boldsymbol{w}}_L - \tilde{\boldsymbol{w}}_L\|_{1,V_L} \sim 2^{-Ld/2}\|\bar{\boldsymbol{w}}_L - \tilde{\boldsymbol{w}}_L\|_{l_2}.$$

Thus by the uniform stability of the transformations T_L° we obtain

$$\|\bar{\boldsymbol{w}}_L - \tilde{\boldsymbol{w}}_L\|_{l_2}^2 \sim \|\bar{\boldsymbol{w}}_{MS} - \tilde{\boldsymbol{w}}_{MS}\|_{l_2}^2 = \sum_{j=0}^{L-1} |\bar{\boldsymbol{d}}_j^{\circ} - \tilde{\boldsymbol{d}}_j^{\circ}|_2^2 \ \lesssim\ 2^{Ld}(2^d-1)\sum_{j=0}^{L-1}\varepsilon_j \ \lesssim\ \varepsilon,$$

which completes the proof. $\square$

This result enables us to prove the convergence of the scheme $\tilde{E}_{MS}$ derived in Section 4.3. Although, we follow in essence Harten's reasoning in [Har95] we deviate from his proof in the beginning. He assumes the representation $E_{MS} \circ thr_\varepsilon = T_L E_L T_L^{-1} \circ thr_\varepsilon$ for the compressed scheme. However, we think that this does not quite reflect the content of the present scheme. For this purpose, we introduce the vector $\bar{\boldsymbol{u}}_L^{n+1}$ of cell averages of the exact solution at time level t_{n+1}, the vectors

$$\tilde{\boldsymbol{v}}_L^{n+1} := T_L^{-1}\tilde{\boldsymbol{v}}_{MS}^{n+1},\ \tilde{\boldsymbol{v}}_{Ms}^{n+1} := \tilde{E}_{MS}\tilde{\boldsymbol{v}}_L^{n}$$

of all cell averages computed by the scheme $\tilde{E}_{MS}$ and the auxiliary vector $\bar{\boldsymbol{v}}_L^{n+1} := E_L\tilde{\boldsymbol{v}}_L^{n}$ which is the evolution of $\tilde{\boldsymbol{v}}_L^{n}$ *without* truncation. Then the error $\boldsymbol{e}_L^{n+1} := \tilde{\boldsymbol{v}}_L^{n+1} - \bar{\boldsymbol{u}}_L^{n+1}$ of time level t_{n+1} can be split into

$$\boldsymbol{e}_L^{n+1} = (\tilde{\boldsymbol{v}}_L^{n+1} - \bar{\boldsymbol{v}}_L^{n+1}) + (E_L\tilde{\boldsymbol{v}}_L^{n} - E_L\bar{\boldsymbol{u}}_L^{n}) + (E_L\bar{\boldsymbol{u}}_L^{n} - E_L\bar{\boldsymbol{u}}_L^{n+1})$$

where the first term corresponds to the stability condition according to Proposition 4.1, the second and the third term to the stability and the truncation error of the underlying finite volume scheme on the level L. The perturbation caused by thresholding are easily estimated when the sets of significant coefficients $\tilde{\mathcal{D}}^{n+1}$ have the structure of graded trees. By this we mean that $(j+1, 2\boldsymbol{k}+\boldsymbol{e}) \in \tilde{\mathcal{D}}^{n+1}$ implies $(j, \boldsymbol{k}+\boldsymbol{q}) \in \tilde{\mathcal{D}}^{n+1}$ for some stencil $Q = \{\boldsymbol{q} \in \mathbb{Z}^d \ :\ |q_i| \leq q\}$. In fact, in this case $\bar{d}^{n+1}_{j,\boldsymbol{e},\boldsymbol{k}} = \tilde{d}^{n+1}_{j,\boldsymbol{e},\boldsymbol{k}}$, $(j,\boldsymbol{k}) \in \tilde{\mathcal{D}}^{n+1}$. In our experiments the sets $\tilde{\mathcal{D}}^{n+1}$ automatically exhibit such a tree structure, see Figure 6. In general the required gradedness could always be arranged by a slight expansion that does not increase the asymptotic order of complexity. Due to the definition of the predicited set $\tilde{\mathcal{D}}^{n+1}$ (4.7) and the hyperbolicity (4.8), the assumption

$$|\bar{d}_{j,\boldsymbol{e},\boldsymbol{k}} - \tilde{d}_{j,\boldsymbol{e},\boldsymbol{k}}| \leq \varepsilon_j$$

of Proposition 4.1 is in fact satisfied such that

$$\|\tilde{\boldsymbol{v}}_L^{n+1} - \bar{\boldsymbol{v}}_L^{n+1}\|_{1,V_L} \ \lesssim\ \varepsilon.$$

From here on we can follow now Harten's proof of convergence which results in the following theorem; see also [Got98]) for more details.

Theorem 4.1 *Let the $\{T_L^\circ\}_{L\in\mathbb{N}_0}$ be uniformly stable and the parameter transformation τ (2.4) be regular such that $|V_{j,\boldsymbol{k}}| \sim 2^{-jd}$ holds. Suppose that E_L is a consistent scheme with the bound τ_{1,V_L} for the local truncation error. If E_L is either linear with*

$$\|E_L\|_{1,V_L} \leq 1 + K\,\Delta t_L, \tag{4.35}$$

or l_1–contractive, i. e.,

$$\|E_L\overline{\boldsymbol{v}}_L - E_L\overline{\boldsymbol{w}}_L\|_{1,V_L} \leq \|\overline{\boldsymbol{v}}_L - \overline{\boldsymbol{w}}_L\|_{1,V_L}\,, \qquad \overline{\boldsymbol{v}}_L,\ \overline{\boldsymbol{w}}_L \in \mathbb{R}^{N_L},$$

then the multiscale scheme $\tilde{E}_{MS}$ is convergent. In particular, choosing

$$\varepsilon = \mathcal{O}(\Delta t_L\,\tau_{1,V_L}), \qquad \left\|\boldsymbol{e}_L^0\right\|_{1,V_L} = \mathcal{O}(\tau_{1,L})$$

preserves the order of accuracy of E_L for $\tilde{E}_{MS}$.

5 Numerical Results

In this section, we consider a hypersonic flow around a blunt body. This occurs, for instance, when a spacecraft re–enters the atmosphere. In case of supersonic flow, i. e., if the free–stream Mach number satisfies

$$Ma_\infty := \frac{v_\infty}{a_\infty} > 1, \tag{5.1}$$

where v_∞ and a_∞ are the free–stream quantities of the velocity and the sound speed, respectively, this leads to a detached bow shock (see Figure 4). Between the body and the shock, there is a subsonic region, i. e., $Ma < 1$, near the stagnation streamline. Through the shock, the high kinetic energy resulting from the large velocity is transformed into internal energy. Additionally, the medium is compressed. Consequently, this leads to an increase of density, pressure and temperature. Moreover, dissociation and ionization of the molecules occur for sufficiently high temperatures. If additionally viscosity and heat conduction are taken into account, a laminar boundary layer builds up. For numerical computations, it is important that the subsonic region is fully contained in the computational domain.

In our example, we take an idealized cylinder of infinite length with the radius $r = 0.1$ [m] as blunt body. The direction of the free–stream is assumed to be parallel to the x–axis such that we can consider a cross–section of the cylinder. Thus, the problem becomes two–dimensional.

We are aware of the fact that the most adequate physical model requires a viscous calculation taking into account chemical and thermal non–equilibrium. However, our primary objective here is to test the proposed scheme by an example which is quite demanding in at least two respects. Firstly, it involves curvilinear two–dimensional meshes. Secondly, the conserved quantities exhibit a highly heterogeneous behaviour. Therefore we neglect at this point dissipative effects and suppose that the gas is perfect such that we can employ the *Euler equations*

$$\begin{array}{rcl} \varrho_t + \mathrm{div}\;(\varrho\boldsymbol{v}) & = & 0 \\ (\varrho\boldsymbol{v})_t + \mathrm{div}\;(\varrho\boldsymbol{v}\boldsymbol{v}^T + p\boldsymbol{I}) & = & \boldsymbol{0} \\ (\varrho E)_t + \mathrm{div}\;((\varrho\, E + p)\boldsymbol{v}) & = & 0 \end{array}\;, \tag{5.2}$$

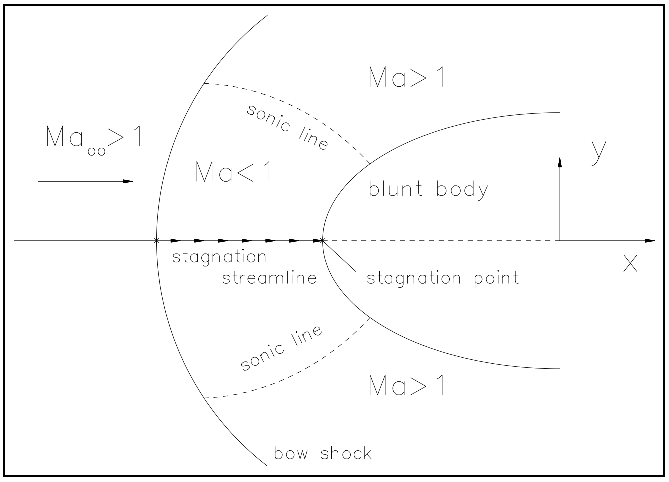

Figure 4: Supersonic blunt body flow

which is closed by the *thermal* and *caloric equation of state*

$$p := \varrho RT = (\gamma - 1)\varrho e \quad \text{and} \quad e := c_v T \tag{5.3}$$

for the pressure p and the internal energy e. The latter is related to the total energy E by $E = e + \boldsymbol{v}^2/2$. Since we only consider undissociated air, the specific gas constant R, the specific heat capacity at constant volume and the ratio of the heat capacities are given by

$$R = 277.133 \text{ [J/kg K]}, \quad c_v = 692.833 \text{ [J/kg K]}, \quad \gamma = 1 - R/c_v = 1.4.$$

In this case, the sound speed is given by

$$a = \sqrt{\frac{\gamma\, p}{\varrho}}. \tag{5.4}$$

The free–stream direction is supposed to be parallel to the x–axis as in Figure 4. The parameters

$$Ma_\infty = 10.0 \text{ [–]}, \quad p_\infty = 191.6 \text{ [Pa]}, \quad v_\infty = 2928.0 \text{ [m/s]}$$

determine the free–stream conditions. According to (5.1) and (5.4), the density is given by

$$\varrho_\infty = \frac{\gamma\, p_\infty\, Ma_\infty^2}{v_\infty^2} = 3.129 \cdot 10^{-3} \qquad \text{[kg/m}^3\text{]}$$

which yields the momentum

$$(\varrho\, v)_\infty = \varrho_\infty\, v_\infty = 9.162 \qquad \text{[kg/m}^2\text{s]}$$

in the free–stream. Applying equation (5.3) we obtain the specific internal and total energy

$$(\varrho\, e)_\infty = \frac{p_\infty}{\gamma - 1}, \quad (\varrho\, E)_\infty = (\varrho\, e)_\infty + \frac{1}{2}\,\varrho_\infty\, v_\infty^2 = 13891.746 \text{ [J/m}^3\text{]}$$

and, hence, the temperature is given by

$$T_\infty = \frac{e_\infty}{c_v} = 220.954 \qquad \text{[K]}. \tag{5.5}$$

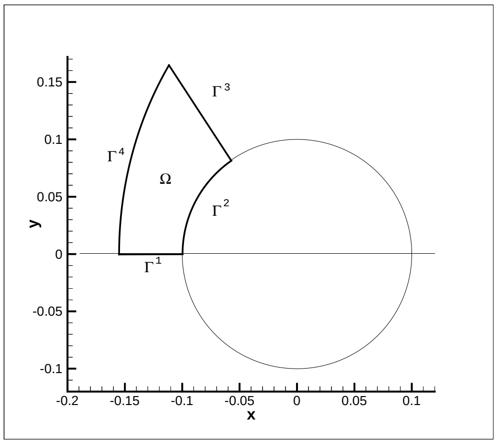

Figure 5: Computational domain

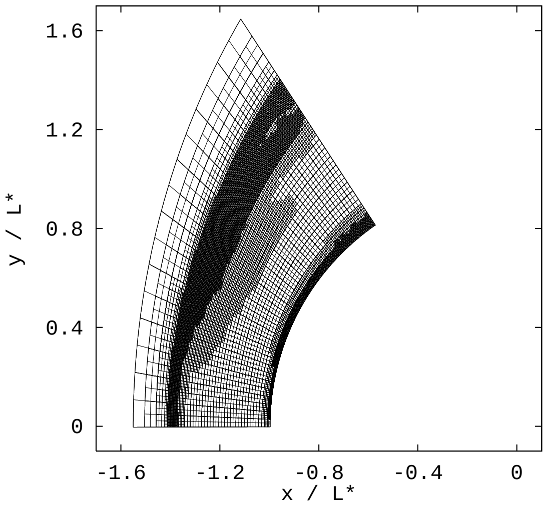

Figure 6: Adaptive grid

Since the flow domain is axially symmetric with respect to the x–axis, the computational domain has been chosen as shown in Figure 5. The boundary Γ^1 is below the x–axis such that the bottom layer of cells is axially symmetric with respect to the x–axis, so that we are able to access the numerical data on the stagnation line.

Before discussing initial data and boundary conditions, we specify the parameters for the numerical experiments. Concerning the underlying finite volume scheme, we have

- a spatial discretization involving $N_x \times N_y = 192 \times 256$ quadrilateral cells approximating a curvilinear grid which is formally produced by the parameter transformation τ; the grid points are determined by an algebraic grid generator;
- temporal step size $\Delta t = 10^{-8}$ and $N_T = 100000$ time steps;
- second order ENO scheme [Mül93], reconstruction of the primitive function of the characteristic variables and internal approximative Riemann solver of Roe [Roe81].

The following multiscale transformation has been applied:

- $M = 3$ vanishing moments with respect to the grid on the finest level;
- $L = 5$ refinement levels such that one has a resolution of $N_{0,1} \times N_{0,2} = 6 \times 8$ and $N_{L,1} \times N_{L,2} = N_x \times N_y = 192 \times 256$ cells on the coarsest and on the finest level, respectively;
- truncation tolerance $\varepsilon = 10^{-4}$ and $a = 2.0$ in (4.6);

For the initial data, we have chosen dimensionless values since then the orders of magnitude of the conservative quantities are equilibrating. We have made the experience that this improves the multiscale scheme. Otherwise, one should introduce for each component independent truncation tolerances.

ϱ	$1.921 \cdot 10^{-2}$	[kg/m^3]
p	$2.475 \cdot 10^4$	[Pa]
T	$4.649 \cdot 10^3$	[K]
Ma	$1.776 \cdot 10^{-3}$	[-]

Table 1: Stagnation point data

$N_{F,min}$	13248
$N_{F,max}$	48980
η_{min}	13.42
η_{max}	49.60
σ	1.54

Table 2: Comparison of multiscale scheme and basic finite volume scheme

The Euler equations (5.2) and equation (5.3) for the pressure are invariant under the transformation to dimensionless variables. Thus, the program remains unchanged. Only the data has to be scaled with respect to the reference values for which we choose $L^* = r$, $\varrho^* = \varrho_\infty$, $v^* = v_\infty$ for the length, the density and the velocity, respectively. Consequently, the dimensionless free–stream data is determined by

$$\widetilde{\varrho}_\infty = 1.0, \quad \widetilde{(\varrho\,\boldsymbol{v})}_\infty = (1.0, 0.0)^T, \quad \widetilde{(\varrho\,E)}_\infty = 0.5179. \tag{5.6}$$

In general, problems occur in the computation when the free–stream conditions act directly onto the body. In order to avoid that, we introduce an artificial layer composed of ten layers of cells at the body. In this layer, we assume the gas to be at rest separated by a discontinuity from the free–stream conditions, i. e.,

$$\overline{\boldsymbol{v}}^0_{L,(k,l)} := \begin{cases} \widetilde{\boldsymbol{u}}_\infty, & 0 \le k \le N_{L,1} - 11 \\ \widetilde{\boldsymbol{u}}_{rest}, & N_{L,1} - 10 \le k \le N_{L,1} - 1 \end{cases}, 0 \le l \le N_{L,2} - 1,$$

where the components of $\widetilde{\boldsymbol{u}}_\infty$ are given by (5.6). The gas in rest is given by $\widetilde{\boldsymbol{u}}_{rest}$ with the components

$$\widetilde{\varrho}_{rest} = 1.0, \quad \widetilde{(\varrho\,\boldsymbol{v})}_{rest} = \boldsymbol{0}, \quad \widetilde{(\varrho\,E)}_{rest} = 1.786 \cdot 10^{-2}.$$

It remains to discuss the boundary conditions. We have already mentioned in Section 2 that we prefer to employ auxiliary data outside of the computational domain. For the second order ENO scheme, we need two auxiliary layers of cells.

The conditions at the boundaries Γ^1 and Γ^2 (see Figure 5) are quite similar since they are both based on reflection. If $\overline{\boldsymbol{u}} = (\overline{\varrho},\ \overline{\varrho\,\boldsymbol{v}}^T,\ \overline{\varrho\,E})^T$ is a vector of cell averages of density, momentum and total energy, the vector

$$\overline{\boldsymbol{u}}_{refl,\boldsymbol{n}} = \left(\overline{\varrho},\ (\overline{\varrho\,\boldsymbol{v}} - 2(\overline{\varrho\,\boldsymbol{v}} \cdot \boldsymbol{n})\boldsymbol{n})^T,\ \overline{\varrho\,E}\right)^T$$

is the vector whose velocity is reflected at a straight line with normal unit vector $\boldsymbol{n}$. At the boundary Γ^1, we have axial symmetry with respect to the x–axis whose normal vector is $\boldsymbol{e}_2$. In addition, we have to take into account that the cells $V_{L,(k,0)}$ are axially symmetric with respect to this axis. Thus, the boundary conditions for Γ^1 read

$$\overline{\boldsymbol{v}}^n_{L,(k,-i)} = \left(\overline{\boldsymbol{v}}^n_{L,(k,i)}\right)_{refl,\boldsymbol{e}_2}, \qquad k = 0, 1, \ldots, N_{L,1} - 1,\ i = 1, 2.$$

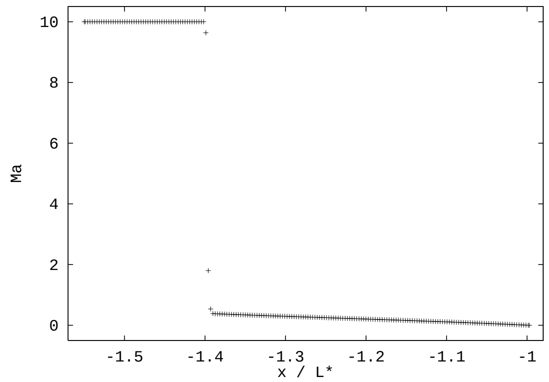

Figure 7: Mach number

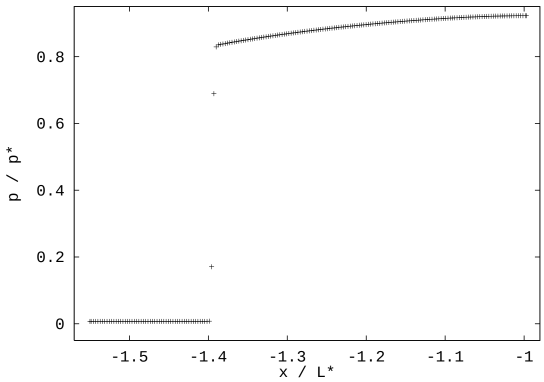

Figure 8: Pressure

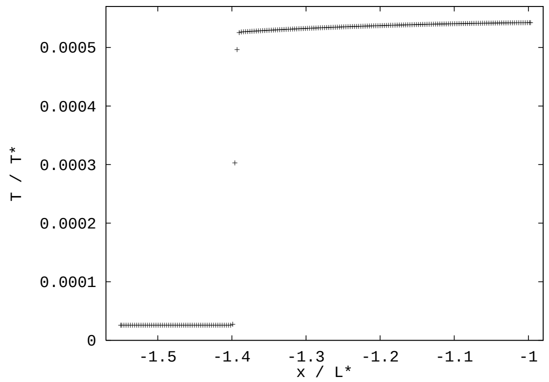

Figure 9: Temperature

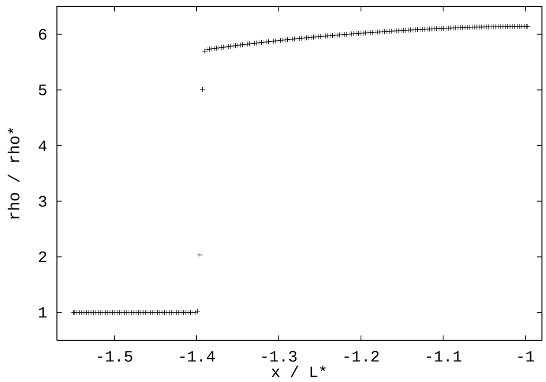

Figure 10: Density

The boundary Γ^2 is a solid wall where the velocity is reflected. It is approximated by a polygonal connection consisting of the facets $\Gamma_{1,L,(N_{L,1},l)}$ with the normal vectors $\boldsymbol{n}_{1,L,(N_{L,1},l)}$. Consequently, the boundary conditions for Γ^2 are given by

$$\overline{\boldsymbol{v}}^n_{L,(N_{L,1}-1+i,l)} = \left(\overline{\boldsymbol{v}}^n_{L,(N_{L,1}-i,l)}\right)_{refl,\boldsymbol{n}_{1,L,(N_{L,1},l)}}, \quad i=1,2,\ l=0,1,\dots,N_{L,2}-1.$$

The boundary Γ^3 is an outflow boundary. Here, we use linear extrapolation of the flow data along a straight line through the centroids of the cells $V_{L,(k,N_{L,2}-2)}$ and $V_{L,(k,N_{L,2}-1)}$ for the data of the auxiliary cells, i. e.,

$$\overline{\boldsymbol{v}}^n_{L,(k,N_{L,2}-1+i)} = \overline{\boldsymbol{v}}^n_{L,(k,N_{L,2}-1)} + i\left(\overline{\boldsymbol{v}}^n_{L,(k,N_{L,2}-1)} - \overline{\boldsymbol{v}}^n_{L,(k,N_{L,2}-2)}\right)$$
$$i=1,2,\ k=0,1,\dots,N_{L,1}-1.$$

For the boundary Γ^4, inflow has to be modeled with respect to the free–stream data. This corresponds to inflow boundary conditions, i. e.,

$$\overline{\boldsymbol{v}}^n_{L,(-i,l)} = \widetilde{\boldsymbol{u}}^n_\infty, \qquad i=1,2,\ l=0,1,\dots,N_{L,2}-1.$$

The steady state of the numerical solution has been approximately achieved after 100000 time steps. In Table 1, density, pressure, temperature and Mach number at the stagnation point which corresponds to cell $V_{L,(N_{L,1}-1,0)}$ are given. The plots of these quantities on the stagnation streamline — corresponding to cells $V_{L,(k,0)}$, $k = 0, 1, \ldots, N_{L,1} - 1$ — are shown in Figures 7 – 10. The contour plots of the same state variables can be seen in Figures 11 – 14.

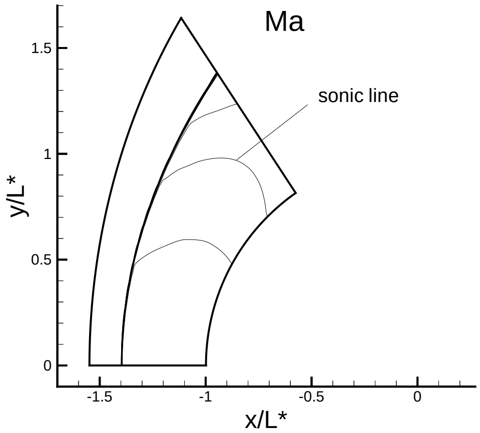

Figure 11: Mach number

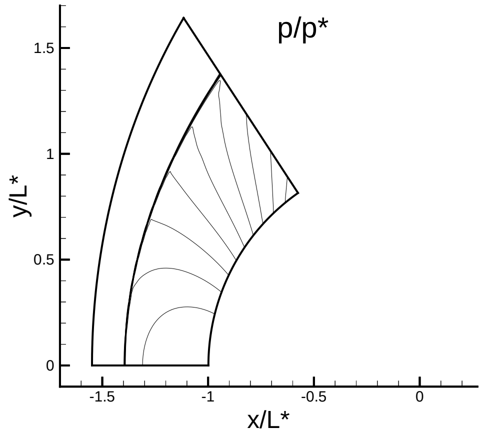

Figure 12: Pressure

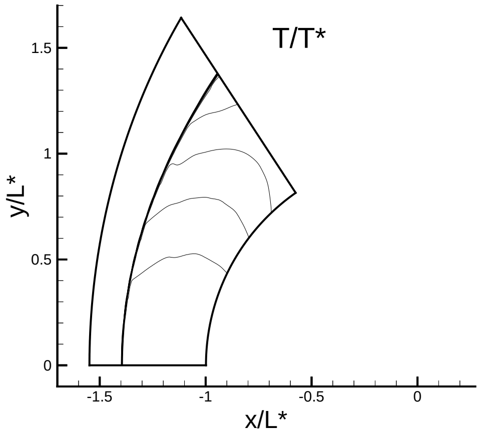

Figure 13: Temperature

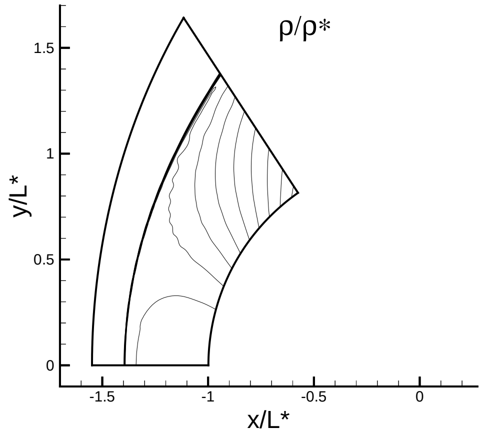

Figure 14: Density

Here, the reference values for the length and the density are taken from (5.6). For the pressure, one takes $p^* = \varrho^* (v^*)^2 = 26825.493$ [Pa]. For the reference value of the temperature, we have one degree of freedom. Here, we choose the reference value for the specific heat at constant volume as $c_V^* := 1.0$ [J/kg K] and $\widetilde{c}_V = c_V / c_V^* = 692.833$ [–]. Then equation (5.5) for the temperature remains invariant under the transformation to dimensionless variables, and we obtain for the reference temperature $T^* = (v^*)^2 / c_V^* =$

8573184 [K]. The Mach number is already dimensionless and, thus, one has $Ma^* = 1$ or $\widetilde{Ma} = Ma$.

In Figure 6, the spatial distribution of the significant detail coefficients is shown, i. e., the boundary lines of the four cells $V_{j+1,2\boldsymbol{k}+\boldsymbol{e}}$, $\boldsymbol{e} \in E$ are inserted if $(j, \boldsymbol{k}) \in \widetilde{\mathcal{D}}^{100001}$. The finest level $L = 5$ is only required in the neighborhood of the shock and on parts of the outflow boundary Γ^3. Note that the grid refinement near the boundary Γ^2 results from the choice of boundary conditions, causing a jump in the normal velocity component. Since this enters the numerical fluxes, it is detected by the multiscale decomposition.

In Table 2, the values for $N_{F,min/max}$, $\eta_{min/max}$ and σ are collected. Here, the numbers $N_{F,min/max}$ denote the minimum and the maximum of the numbers $N_F(n)$ of flux evaluations necessary for the multiscale scheme in each time step, i. e.,

$$N_{F,min} = \min_{0 \le n \le N_T} N_F(n), \qquad N_{F,max} = \max_{0 \le n \le N_T} N_F(n).$$

These have to be compared to N_L^* which is the number of flux evaluations for the underlying finite volume scheme without multiscale approach, i. e.,

$$N_L^* = \#I_{L,1} + \#I_{L,2} = (N_{L,1} + 1)\, N_{L,2} + N_{L,1}\, (N_{L,2} + 1).$$

The ratios $\eta_{min/max} = N_{F,min/max}/N_L^*$ and $\sigma = T_{FV}/T_{MS}$ where T_{FV} and T_{MS} are the computational times for the basic finite volume scheme and the multiscale scheme, respectively, are efficiency measures. Due to the high computational cost, we have only performed the computation for one set of parameters.

6 Conclusions and Outlook

We have developed in this paper a multiscale scheme that reduces the cost of expensive flux evaluations arising in highly accurate finite volume discretizations for boundary fitted multidimensional meshes. The underlying compression effect is based on judiciously chosen efficient multiscale transformations.

In the spirit of Harten's original proposal the scheme still produces flux values for every cell in the mesh of highest resolution. As quantified by our complexity considerations one can therefore not expect yet to accomplish a reduction of the asymptotic *order* of computational and storage cost. However, in contrast to other approaches a core ingredient of our scheme are evolution equations for the detail coefficients. While these equations on one hand provide the interpolated unexpensive flux substitutes (4.28) they also allow us on the other hand, in principle, to avoid single scale representations of all averages on the full mesh of highest resolution when performing the time step. Therefore this part of the scheme yields an important prerequisite for ultimately arriving at a fully adaptive scheme involving meshes with locally varying level of highest resolution as indicated already in Figure 6. This will be taken up in a forthcoming paper. In this context locally variable time steps become feasible so that the severe constraints for large L due to the CFL condition could be relaxed significantly, see (4.17).

We have described the derivation of our multiscale scheme for *scalar* equations. Its extension to systems is straightforward as reflected by the above numerical example, see also [Got98].

The present use of structured grids eases the organization of flux balances. However, the approach extends also to unstructured grids, see also [GM97]. In particular, when no natural nesting of grids can be exploited, agglomoration provides a suitable hierarchy as detailed in [SS95].

References

[BH97] B. Bihari and A. Harten. Multiresolution schemes for the numerical solution of 2–D conservation laws I. *SIAM J. Sci. Comput.*, 18(2):315–354, 1997.

[CDF92] A. Cohen, I. Daubechies, and J. Feauveau. Bi–orthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 45:485–560, 1992.

[CDP96] J.M. Carnicer, W. Dahmen, and J.M. Peña. Local decomposition of refinable spaces and wavelets. *Appl. Comput. Harmon. Anal.*, 3:127–153, 1996.

[Dah96] W. Dahmen. Stability of multiscale transformations. *J. Fourier Anal. Appl.*, 4:341–362, 1996.

[Dah97] W. Dahmen. Wavelet and multiscale methods for operator equations. *Acta Numerica*, 6:55–228, 1997.

[DKU] W. Dahmen, A. Kunoth, and K. Urban. Biorthogonal spline-wavelets on the interval – stability and moment conditions. To appear in Appl. Comput. Harmon. Anal.

[DL93] R.A. DeVore and G.G. Lorentz. *Constructive Approximation.* Grundlehren der Mathematischen Wissenschaften. Springer, Berlin, 1993.

[GM97] B. Gottschlich–Müller and S. Müller. On multi–scale concepts for multi–dimensional conservation laws. IGPM–Report 140, RWTH Aachen, 1997.

[Got98] B. Gottschlich–Müller. *On Multiscale Concepts for Multidimensional Conservation Laws.* PhD thesis, RWTH Aachen, 1998. submitted.

[Har95] A. Harten. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Comm. Pure Appl. Math.*, 48(12):1305–1342, 1995.

[HEOC87] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy. Uniformly high order accurate essentially non–oscillatory schemes III. *J. Comp. Phys.*, 71:231–303, 1987.

[Krö97] D. Kröner. *Numerical Schemes for Conservation Laws.* Advances in Numerical Mathematics. Wiley–Teubner, 1997.

[LeV92] R.J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhäuser Verlag, Basel, 2nd edition, 1992.

[Mül93] S. Müller. *Erweiterung von ENO–Verfahren auf zwei Raumdimensionen und Anwendung auf hypersonische Staupunktprobleme.* PhD thesis, RWTH Aachen, April 1993.

[Roe81] P. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comp. Phys.*, 43:357–372, 1981.

[Sjö95] B. Sjögreen. Numerical experiments with the multiresolution scheme for the compressible Euler equations. *J. Comp. Phys.*, 117:251–261, 1995.

[SS95] F. Schröder–Pander and T. Sonar. Preliminary investigations on multiresolution analysis on unstructured grids. DLR Forschungsbericht IB 223–95 A 36, DLR Göttingen, 1995.

[Swe] W. Sweldens. The lifting scheme: A construction of second generation wavelets. To appear in SIAM J. Numer. Anal.