An adaptive multiresolution discontinuous Galerkin scheme for conservation laws

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von
Nils Gerhard, M.Sc.
aus Gießen

Berichter:  apl. Professor Dr. rer. nat. Siegfried Müller
           Univ.-Prof. Dr. rer. nat. Wolfgang Dahmen

Tag der mündlichen Prüfung:  6.07.2017

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.
Acknowledgments

I am very grateful for all the support I received while writing this thesis. First of all, I would like to thank Prof. Dr. Siegfried Müller for his guidance and support throughout the development of this thesis. I enjoyed working with him very much for the last five years. Whenever assistance was required, he had time for discussions. Furthermore, I want to thank Prof. Dr. Wolfgang Dahmen for reviewing this thesis. I am grateful for his comments which helped improving this work. Moreover, I want to thank Daniel Caviedes-Voullième for the fruitful discussions we had and for sharing his knowledge in shallow water modeling.

I want to thank all my colleagues from the IGPM. It was a pleasure to work there. In particular, I appreciate the technical assistance provided by Frank Knoben regarding problems with the cluster and with debugging my code. Many thanks go to Jochen Schütz for all the fruitful discussions we had. I especially thank Michael Rom and Kirsten Pottbacker for proofreading a first version of this thesis.

Finally, I want to thank my family and friends for their support and encouragement.
Contents

1 Introduction ................................................................. 7
  1.1 Motivation and background ........................................ 7
  1.2 Scope of the thesis .................................................. 11
  1.3 Outline of the thesis ............................................... 13

2 Runge-Kutta discontinuous Galerkin schemes ...................... 15
  2.1 The DG space, jumps and averages ................................ 16
  2.2 DG discretization of hyperbolic conservation laws ............ 18
  2.3 Convections dominated problems ................................ 23
  2.4 Basis expansions .................................................. 25
  2.5 Temporal discretization and limiting ............................ 26
  2.6 Numerical results .................................................. 30
    2.6.1 Linear convection-diffusion equation ....................... 30
    2.6.2 Compressible Euler equations ................................. 31
    2.6.3 Compressible Navier-Stokes equations ...................... 33

3 Multiresolution analysis .............................................. 41
  3.1 Nested grid hierarchies ............................................ 42
    3.1.1 Examples of nested hierarchies ............................... 43
  3.2 Multi-scale decomposition ....................................... 45
  3.3 Separation of local contributions ............................... 48
  3.4 Cancellation property ............................................ 52
  3.5 Significance, thresholding and grid adaptation ................ 52
  3.6 Orthogonal bases .................................................. 57
  3.7 Construction of multiwavelets ................................... 60
  3.8 Choice of the local norms ....................................... 64
  3.9 Algorithms and implementation aspects ........................ 65
    3.9.1 Algorithms for the multi-scale transformations .......... 65
    3.9.2 Algorithms for thresholding and grid adaptation .......... 70
## 4 Dynamical grid adaptation for DG schemes

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Significant contributions in the semi-discrete DG solution</td>
<td>74</td>
</tr>
<tr>
<td>4.2 Significant contributions in the fully-discrete DG solution</td>
<td>76</td>
</tr>
<tr>
<td>4.3 Prediction of significant contributions</td>
<td>79</td>
</tr>
<tr>
<td>4.4 The fully adaptive DG scheme</td>
<td>80</td>
</tr>
<tr>
<td>4.5 Choice of the local threshold value</td>
<td>82</td>
</tr>
<tr>
<td>4.5.1 Uniform dyadic hierarchies</td>
<td>82</td>
</tr>
<tr>
<td>4.5.2 Non-uniform hierarchies</td>
<td>85</td>
</tr>
<tr>
<td>4.6 Adaptation and limiting</td>
<td>85</td>
</tr>
<tr>
<td>4.7 Adaptive initialization</td>
<td>86</td>
</tr>
<tr>
<td>4.8 Parallelization of the adaptive scheme</td>
<td>88</td>
</tr>
<tr>
<td>4.9 Numerical results</td>
<td>90</td>
</tr>
<tr>
<td>4.9.1 Two-dimensional inviscid Burgers’ equation</td>
<td>91</td>
</tr>
<tr>
<td>4.9.2 Compressible Euler equations</td>
<td>94</td>
</tr>
<tr>
<td>4.9.3 Shock vortex interaction with a boundary layer in a viscous shock tube</td>
<td>106</td>
</tr>
<tr>
<td>4.9.4 The bottom line of the (numerical) comparison of the wavelet-free and the classical approach</td>
<td>116</td>
</tr>
</tbody>
</table>

## 5 Grid adaptation for shallow water equations

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 The reference scheme</td>
<td>118</td>
</tr>
<tr>
<td>5.1.1 Treatment of bottom topography</td>
<td>118</td>
</tr>
<tr>
<td>5.1.2 Well-balancing</td>
<td>119</td>
</tr>
<tr>
<td>5.1.3 Positivity-preserving</td>
<td>120</td>
</tr>
<tr>
<td>5.1.4 Local projection limiting for shallow water equations</td>
<td>121</td>
</tr>
<tr>
<td>5.2 Grid adaptation for shallow water equations</td>
<td>121</td>
</tr>
<tr>
<td>5.2.1 Bottom topography and well-balancing</td>
<td>122</td>
</tr>
<tr>
<td>5.2.2 Positivity-preserving</td>
<td>123</td>
</tr>
<tr>
<td>5.2.3 Limiting in the adaptive scheme</td>
<td>130</td>
</tr>
<tr>
<td>5.2.4 Tracking of the wet/dry front</td>
<td>130</td>
</tr>
<tr>
<td>5.3 Summary of the adaptive scheme</td>
<td>131</td>
</tr>
<tr>
<td>5.4 Numerical results</td>
<td>131</td>
</tr>
<tr>
<td>5.4.1 Oscillating lake in a parabolic bowl</td>
<td>131</td>
</tr>
<tr>
<td>5.4.2 Tsunami run-up on a complex three-dimensional beach</td>
<td>135</td>
</tr>
</tbody>
</table>

## 6 Conclusion

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>141</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In this thesis we present an adaptive multiresolution discontinuous Galerkin scheme for multi-dimensional systems of conservation laws. The dynamical grid adaptation is based on a multiresolution analysis of the underlying data. Section 1.1 surveys the motivation for adaptive schemes and the background of multiresolution based grid adaptation. Then, the scope of the thesis is discussed in Section 1.2. Finally, the outline of this work is shown in Section 1.3.

1.1 Motivation and background

Nowadays numerical simulations of fluid flows play an important role in industry for designing new products, e.g., in aviation industry, as well as in forecasting natural disasters such as tsunamis. Systems of (hyperbolic) conservation laws are frequently considered to model these flows such as the shallow water equations, the compressible Euler equations or the compressible Navier-Stokes equations. Solutions of these equations often reveal a very heterogeneous structure: in some regions the solution is smooth, whereas in other regions strong gradients or even discontinuities occur.

When approximating these solutions with a numerical scheme an underlying computational grid consisting of cells or grid points is needed. Due to the presence of discontinuities in the solutions a fine resolution in the grid, i.e., small cells or condensed grid points, is locally required to realize a desired accuracy. However, in regions where the solution is smooth, a coarse resolution might be acceptable to ensure this accuracy. If the grid is considered to be uniform, the resolution requirements in regions where the solution is non-smooth causes a tremendous number of cells in the computational grid.

A possibility to overcome this remedy is to consider adaptive methods [1,21,22,24,133]. Here, the grid resolution is not uniform but is locally adjusted to the local behavior of the solution. When considering time-dependent problems, the adaptation has to dynamically take the temporal evolution of the solution into account. To this end, numerous indicators
to trigger local grid adaptation have been developed. The majority of these strategies can be categorized into the following three different paradigms:

(i) Some adaptive methods are based on local error estimators. In the context of hyperbolic problems numerous estimators have been developed, for instance, a posteriori error estimates such as Bey and Oden [24], Adjerid et al. [1], Houston et al. [101, 102, 107], Dedner et al. [66], Mavriplis et al. [173] and Gieselmann et al. [84]. All these indicators tempt to estimate the error of the adaptive solution. However, a reliable and efficient adaptation strategy must be based on an error estimator \( e_h \) which can be bounded from below and above up to a constant by the real error of the scheme, i.e., there exist \( c, C > 0 \) such that

\[
e \| u - u_h \| \leq e_h (u_h) \leq C \| u - u_h \|
\]

in a suitable norm. Here, we denote by \( u_h \) the numerical approximation and by \( u \) the analytical solution, respectively. In the aforementioned works estimators have been derived which have a bound from below, but a bound from above is missing. The existence of an upper bound requires a stable variational formulation of the underlying partial differential equation. However, the weak formulations of nonlinear systems of conservation laws are not stable in general. Thus, no rigorous error estimates are available.

(ii) A totally different paradigm are sensor-based adaptive methods, where a sensor is considered to trigger local grid adaptation. Examples for ’sensors’ are gradients, jumps or curvature of the solution, cf. [10, 109, 121, 142, 146, 147]. These methods do usually not provide any control of the error but are used frequently in practice.

(iii) Another class of adaptive methods can be considered as perturbation methods. The key idea here is to improve the efficiency of a given (reference) scheme on a (uniformly) refined (reference) grid by computing actually only on a locally refined adapted subgrid, while preserving the accuracy of the discretization on the full uniform grid, cf. [29, 35, 91, 98, 108]. This paradigm allows at least some control of the perturbation error between reference and adaptive scheme. In this context, the terms efficiency and reliability are used with different meaning than in the context of a posteriori estimator based grid adaptation, see class (i). When speaking of efficiency, we think of the reduction of the computational cost in comparison to the cost of the reference scheme. The term reliability is used here to emphasize that the adaptation process is capable of maintaining the accuracy of the reference scheme.

In this thesis we focus on multiresolution-based grid adaptation. This concept belongs to the class of perturbation methods and does not rely on error estimates. In order to decide where to coarsen the reference grid, a reliable indicator to control local grid refinement is required. To this end, a multiresolution analysis (MRA) is performed, where the data corresponding to the current solution are represented as data on some coarse level and difference information called details corresponding to successive refinement levels. This new representation of the data reveals insight into the local behavior of the solution.
1.1. MOTIVATION AND BACKGROUND

It can be shown that the details become locally small with increasing refinement level when the underlying function is locally smooth. As suggested by this so-called cancellation property, we may determine a locally refined grid performing data compression on local details using hard thresholding. This significantly reduces the complexity of the data. Based on this thresholding, local grid adaptation is performed, where we refine an element whenever there exists a significant detail. Of course, the core issue in this strategy is to realize this in such a way that at no point of the computation there is ever made use of the fully refined grid. Moreover, a repetition of time steps as frequently considered in estimator based adaptive schemes is avoided.

This concept originates from Harten’s work in the early 1990s in the framework of finite volume schemes, cf. [98]. In his work Harten was not aiming at local grid adaptation but applied the MRA only to switch between expensive and cheap flux evaluations to design a cost-effective scheme while still working on a uniformly fine grid. In the late 1990s, the MRA was also used to design fully adaptive finite volume schemes performing local grid adaptation, cf. [91, 117, 152]. In the following decade multiresolution-based grid adaptation methods have been quite successful with FV solvers, e.g., Bramkamp et al. [29], Domingues et al. [71], Bürger et al. [34], Schneider et al. [157, 158] and Müller et al. [135]. A comprehensive review can be found in [134] and [59] provides some overview on recent trends.

For the realization of a MRA Harten’s discrete framework [96] or wavelets, i.e., generators for the details characterizing difference information between successive refinement levels, can be employed. For this purpose, bi-orthogonal wavelets [55] have been used in the context of finite volume schemes. The efficiency of multiresolution-based grid adaptation schemes crucially relies on the compression rate. In the wavelet context it can be proven that the rate of decay of the details becomes larger with increasing number of vanishing moments of the wavelets, i.e., more details may be discarded in smooth regions and the adaptive grid becomes coarser. However, for bi-orthogonal wavelets to realize more vanishing moments requires to extend the support of the wavelet functions. Their construction, in particular, becomes even more complicated on unstructured grid hierarchies.

Moreover, the construction and realization of a high-order reference scheme in the context of finite volume methods is rather problematic. The discretization stencils enlarge and consequently the application to unstructured grids is complicated. Furthermore, the large stencils contradict an efficient parallelization of the scheme.

For these reasons, discontinuous Galerkin (DG) methods are nowadays applied to conservation laws. In DG schemes the solution is approximated by piecewise polynomials, i.e., the approximate solution is allowed to be discontinuous at the cell interfaces. Hence, DG methods allow to increase the order of accuracy without enlarging the discretization stencil. This locality enables an efficient parallelization, a simple application to complex geometries as well the use hanging nodes in the grids. However, the locality in the dis-
cretization space results in a large number of degrees of freedom. Thus, it is evident to combine the DG schemes with dynamical grid adaptation when applied to conservation laws. In order to realize a MRA of the DG reference scheme, suitable local generators for the difference information have to be found. The locality of the piecewise polynomials enable a straightforward use of orthogonal bases for the difference spaces. Thus, orthogonal complements can be used to identify these difference spaces. An example of local bases functions for the orthogonal complements are the so-called multiwavelets [119,167]. Multiwavelets allow for higher order vanishing moments, while being supported on a single grid element.

In order to reduce the computational complexity, the multiresolution concept has been combined with DG schemes, cf. [7,35,108,156,160]. A first attempt to extend the concept of multiresolution-based grid adaptation to DG schemes has been reported by Calle et al. [35]. They consider an implicit diffusive DG scheme (IDDG) where the DG discretization is stabilized using streamline diffusion instead of a limiter. The diffusion term is discretized implicitly whereas for the convection term an explicit discretization is applied. To indicate local grid refinement, a multiresolution analysis of the local mean-values is performed using the Haar wavelets. Hence, only a part of the (high) order DG solution is used here.

Later, an adaptive scheme for hyperbolic scalar conservation laws in one spatial dimension has been developed and analyzed in [108,156]. In this scheme the indicator for local grid refinement is based on a MRA of the complete DG solution using Alpert’s multiwavelets [2]. In particular, it has been proven that choosing an ideal threshold value, the accuracy of the adaptive solution is in the same order as the accuracy of the reference solution, but at significantly reduced computational cost. In these works, the MRA is used as an indicator for grid adaptation. The evolution of the (adaptive) solution is still carried out in the space of piecewise polynomials corresponding to the adaptive grid.

In contrast to this approach, adaptive multiresolution DG methods following the spirit of Alpert’s and Belkin’s work [3,25] have been reported in [7,160]. The basic idea in these works is to design an adaptive DG method by representing the numerical solution as well as the operators from the partial differential equation in a multiresolution representation. Thus, the evolution is carried out on the multiresolution coefficients. For that reason, their approaches strongly differ from the work of [35,108,156]. For the multiresolution representation also multiwavelets are considered in both schemes. Both works can be considered as preliminary, since the presented adaptive strategy is not analyzed and only applied to very few numerical configurations.

In all of the aforementioned works only uniform dyadic subdivisions of the cells in the hierarchy are considered, i.e., the cells are refined by dividing a cell into subcells of equal size and shape. For instance in one spatial dimension, the intervals are split in two intervals of the same length.
1.2 Scope of the thesis

In this thesis, we continue the work of [108,156]. In particular, we focus on the extension of their adaptive scheme to the multi-dimensional case as well as on the application to non-linear systems of conservation laws such as the shallow water equations, the compressible Euler equations and the compressible Navier-Stokes equations. Moreover, we extend the concept to more general grid hierarchies where the subdivision of the cells is no longer assumed to be uniform, i.e., the cells on one refinement level do not need to be of equal size. However, we assume that the diameters of the cells tend to zero for increasing refinement levels. This guarantees that the resulting nested sequence of spaces is dense in $L_2$.

A crucial part in the realization of the adaptive scheme is the construction of generators for the contributions of the orthogonal complements. Previous works, cf. [108,156], are based on the explicit construction of multiwavelets, i.e., orthogonal basis functions for these complement spaces. When considering a MRA on a hierarchy consisting of rectangular elements with a uniform dyadic subdivision, e.g., the cells are refined by dividing a coarser cell into $2^d$ cells of equal size and shape, this construction can be performed on a single reference element. The local basis is then obtained by shifting and rescaling the multiwavelets on the reference element. However, when considering the setting of more general non-uniform hierarchies, the local bases functions for the orthogonal complements have to be constructed independently on each cell in the hierarchy. In principle, this is possible, but computationally expensive and thus it is a drawback in the application of the adaptive scheme.

To overcome this issue we propose an alternative approach to realize the grid adaptation which does not rely on the construction of bases functions for the complements. The key idea thereby is to represent the local contributions from the orthogonal complements in terms of the basis function for the piecewise polynomial DG space on the next finer level. To this end, we formulate the MRA in terms of projectors to enable a realization independent of the local generators. In previous work the significance of local contributions was characterized by coefficients in the basis expansion of the detail information from orthogonal complements using multiwavelets. Here, we consider a local function norm for the characterization of significance. Thereby the construction of wavelets has become superfluous. This approach enables a very efficient realization of the adaptive concept to arbitrarily shaped elements since the explicit construction of generators for the complement spaces is avoided.

The efficiency and reliability of the adaptive scheme is strongly influenced by the choice of the local threshold values. If this threshold is chosen too large, we obtain indeed a very coarse mesh resulting in low computational cost, but the additional error introduced by the thresholding is dominating and spoils the accuracy of the solution. In contrast, if the threshold is chosen too small, the grid contains too many cells and efficiency is lost. In [108,156] attempts were made to identify the ideal threshold. However, in order to
access the ideal threshold value a computation with the reference scheme on the fully refined grid is required. For real computations this procedure is extremely expensive and not feasible. For that reason, we develop a robust and reliable strategy for choosing the local threshold values without any need of computing a solution on the (uniform) reference grid. This strategy has not yet been verified analytically. However, it has been numerically validated for numerous test cases.

Although the computational cost can be significantly reduced by local grid adaptation, applying adaptive high-order schemes to multi-dimensional systems of conservation laws is still computationally expensive. Hence, we discuss a strategy for the parallelization of the adaptive scheme. It is based on the approach for the parallelization of multiresolution-based grid adaptation in the context of finite volume schemes, cf. [31].

Many problems are subjected to constraints. For instance, the electric field in the equations of magnetohydrodynamics (MHD) must be divergence-free. Furthermore, quantities have to satisfy certain bounds such as the positivity of the water height in the shallow water equations or the density in compressible Euler equations. Another constraint originates from the discretization of inhomogeneous conservation laws which have to be well-balanced, e.g. shallow water equations. Thus, discrete constraints have to be taken into account in the numerical discretization, cf. [128, 129, 182, 188, 189]. Local grid adaptation might spoil these discrete constraints. Hence, the question is how to ensure them under grid adaptation. Exemplarily, we consider constraints in the context of the shallow water equations. These equations are frequently considered to model shallow free surface flows over non-constant bottom topographies. When solving these problems numerically, important issues must be addressed. In particular, a DG scheme for the shallow water equations must be well-balanced [23, 88, 92] and positivity-preserving [77], i.e., it must be able (i) to keep steady states over non-constant topography, and (ii) to keep positive values of depth, specially near the wet/dry front. For that purpose, the reference scheme is modified to ensure these properties. In order to accelerate the reference scheme, we present an adaptive scheme for shallow water equations. We modify the adaptive strategy to ensure that the presented grid adaptation maintains both constraints. In [73] multiresolution-based mesh adaptation for shallow water flows in the context of FV schemes is discussed. There, only well-balancing has been addressed. Here, we additionally address the problem of wetting and drying. In particular, we prove that our modified adaptive scheme maintains positivity-preserving during the adaptation process. Note, that our objective is only to accelerate the existing reference scheme by performing local grid adaptation. We are not aiming at improving the (uniform) reference scheme.

Parts of this work have already been published in the following journal articles: [80, 81, 83]. The first work regarding the extension to multi-dimensional scalar conservation laws is reported in [83]. In [81] the strategy for the choice of the thresholds and the application to the compressible Euler equations have been addressed. The adaptive scheme for the shallow water equations has been presented and analyzed in [80].
The herein presented adaptive scheme has been applied in further publications: [82, 104, 120]. A preliminary application to the shallow water equations with a constant bottom topography has been reported in [120]. In [104] the adaptive strategy has been used to accelerate a DG solver for fluid-structure coupling of a linear elastic model with the compressible Euler equations. The adaptive scheme for shallow water equations has been used for the simulation of a tsunami run-up onto a beach in [82]. Furthermore, the concept has been presented at the following international conferences:

- YIC GACM, Aachen, Germany, 2015.
- Fifth Chilean Workshop on Numerical Analysis of Partial Differential Equations, Concepción, Chile, 2016.

In all of the already published works, the adaptive scheme has been discussed and applied for multi-dimensional (hyperbolic) systems of conservation laws using a uniform dyadic hierarchy consisting of rectangular elements. The extension to non-uniform hierarchies is only considered in this thesis. Furthermore, we additionally apply the adaptive scheme to the compressible Navier-Stokes equations in this work.

### 1.3 Outline of the thesis

This thesis is structured as follows: as foundation for the adaptive scheme first we establish a (reference) discontinuous Galerkin scheme in Chapter 2. For that purpose, existing approaches from the literature are considered. We are not aiming to design a new reference scheme. To validate the implementation of the reference scheme the order of convergence is numerically confirmed for selected benchmark problems. Moreover, the capability of dealing with turbulent flows is demonstrated by considering two configurations for the compressible Navier-Stokes equations in which turbulence is observed. Then, we perform a MRA for $L^2$ based on the sequence of piecewise-polynomial spaces on a nested grid hierarchy in Chapter 3. In this chapter, the MRA is formulated in terms of orthogonal projections. In particular, the realization of the MRA using a wavelet-free approach is discussed complemented by the corresponding algorithms for the implementation. In Chapter 4 we then focus on the dynamical grid adaptation process based on the MRA from Chapter 3. At the end of this chapter, we present numerical results for the two-dimensional Burgers’ equation as well as for the compressible Euler equations to validate the adaptation strategy. In particular, we investigate whether the newly proposed approach avoiding the construction and the use of multiwavelets is a suitable alternative to the classical approach using multiwavelets. In order to demonstrate how
constraints in the reference scheme can be maintained under grid adaptation, we then present an adaptive scheme for the shallow water equations addressing well-balancing and positivity-preserving in Chapter 5. This is complemented by numerical results for two benchmark problems. We focus on an oscillating flow in a bowl as well as a run-up of a tsunami onto a complex beach. Finally, we give a short conclusion and an outlook.
Chapter 2

Runge-Kutta discontinuous Galerkin schemes

In this chapter, we focus on the discretization of systems of conservation laws

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{H}(\mathbf{u}, \nabla \mathbf{u}) = 0$$

(2.1)

in a bounded domain $\Omega \subset \mathbb{R}^d$ using a discontinuous Galerkin (DG) scheme. We are mostly interested in pure hyperbolic problems, i.e., $\mathbf{H}(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{F}(\mathbf{u})$. Well-known examples are the isentropic Euler equations as well as the compressible Euler equations and the shallow water equations. Furthermore, we are interested in mixed problems, where features of parabolic and hyperbolic equations arise, i.e., an additional viscous flux is added to the inviscid flux: $\mathbf{H}(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{F}(\mathbf{u}) - \mathbf{G}(\mathbf{u}, \nabla \mathbf{u})$. In particular, we are interested in the compressible Navier-Stokes equations. Thereby, we focus on convection-dominant systems, i.e., the influence of diffusion is small compared to the influence of convection.

For the discretization of (2.1) we follow the idea of Runge-Kutta discontinuous Galerkin (RK-DG) schemes, cf. [18, 48, 50, 51, 70], where the solution is approximated using the method of lines. For that purpose, equation (2.1) is first discretized and stabilized in space by a DG approach and then the resulting system of ordinary differential equations in time is solved by a Runge-Kutta (RK) scheme. An overview of recent developments on DG schemes for convection-dominated problems can be found in [162].

The main focus of this chapter is to establish a suitable DG scheme as foundation for the adaptive scheme in Chapter 4. For the convenience of the reader and to fix the notation, we shortly summarize here the main ideas of RK-DG schemes. This chapter is not the key part of the present thesis and we do not claim it to be complete. We first focus on the pure hyperbolic case. Then, we consider the extension to convection-dominated conservation laws, where we particularly aim at the compressible Navier-Stokes equations. Finally, we verify the implementation of the RK-DG scheme for well-known benchmark test cases. In particular, we validate the implementation with smooth test cases for
the Burgers’ equation, the compressible Euler as well as the compressible Navier-Stokes equations.

2.1 The DG space, jumps and averages

The starting point for a DG discretization is a partition of \( \Omega \) with a finite number of cells \( V_\lambda \). Thereby, we assume that each cell \( V_\lambda \) is bounded with a Lipschitz boundary. The corresponding grid is characterized by the index set \( \mathcal{I} \):

\[
\overline{\Omega} = \bigcup_{\lambda \in \mathcal{I}} V_\lambda.
\]

We denote by \( \Delta x \) the smallest diameter of the cells in the grid, i.e.,

\[
\Delta x := \inf_{\lambda \in \mathcal{I}} \text{diam}(V_\lambda).
\]

We are also interested in periodic problems, where the solution is assumed to be spatially periodic in one or more coordinate directions. For that reason, we split \( \partial \Omega \) in two non-overlapping parts:

\[
\partial \Omega = \partial \Omega_{bc} \cup \partial \Omega_{\text{per}}.
\]

Here, \( \partial \Omega_{\text{per}} \) characterizes the parts of the boundary for which periodicity is assumed. We denote the part of \( \partial \Omega \) where boundary conditions are prescribed by \( \partial \Omega_{bc} \).

The DG space \( S \)

The basic idea in DG schemes is to seek for an approximate solution in a finite-dimensional space \( S \) instead of solving the infinite dimensional problem (2.1). This space consists of piecewise polynomials defined on the partition (2.2) of \( \Omega \), i.e.,

\[
S := \{ f \in L^2(\Omega) : f|_{V_\lambda} \in \Pi_{p-1}(V_\lambda) \forall \lambda \in \mathcal{I} \}.
\]

Here, the local polynomial space \( \Pi_{p-1}(V_\lambda) \), \( \lambda \in \mathcal{I} \), is defined as usual by

\[
\Pi_{p-1}(V_\lambda) := \{ u \in L^2(V_\lambda) : u(x) = \sum_{|\alpha|_{1:p}} u_\alpha x^\alpha, x \in V_\lambda \text{ with } u_\alpha \in \mathbb{R} \text{ and } \alpha \in \mathbb{N}_0^d \},
\]

where the multivariate monomials are defined as \( x^\alpha := \prod_{i=1}^d x_i^{\alpha_i} \). Continuity at the interfaces of the grid is not enforced. Thus, functions in \( S \) may contain discontinuities and are not uniquely defined on the boundaries of the cells \( V_\lambda, \lambda \in \mathcal{I} \).
2.1. THE DG SPACE, JUMPS AND AVERAGES

Jump and average operators

For describing and analyzing DG schemes, it has become well-established to define average \{\}\ and jump \([\cdot]\) operators, cf. [9, 68]. These operators provide an easy way to deal with double valued functions in a compact notation. In order to define them, we first consider a skeleton resulting from the union of all faces by

\[
\Gamma := \bigcup_{\lambda \in \Sigma} \partial V_\lambda \quad \text{and} \quad \Gamma_I := \Gamma \setminus \partial \Omega_{\text{bc}}.
\]

Consequently, we can write \(\Gamma, \Gamma_I\) and \(\partial \Omega_{\text{bc}}\) as the (non-overlapping) union of faces \(e\), i.e.,

\[
\Gamma = \bigcup_{e \in \Sigma} e, \quad \Gamma_I = \bigcup_{e \in \Sigma_I} e, \quad \partial \Omega_{\text{bc}} = \bigcup_{e \in \Sigma_{\text{bc}}} e.
\]

Each face \(e \in \Sigma_I\) is the intersection of the boundaries from two adjacent cells \(V_{\lambda_1}\) and \(V_{\lambda_2}\), i.e., \(e = \partial V_{\lambda_1} \cap \partial V_{\lambda_2}\). Moreover, we denote by \(\mathbf{n}_{\lambda_1}\) and \(\mathbf{n}_{\lambda_2}\) the outward pointing unit normal vector corresponding to \(V_{\lambda_1}\) and \(V_{\lambda_2}\), respectively. Following [68], we define a unit normal vector \(\mathbf{n}_e\) for each face in \(\Gamma\) such that it coincides with the outward pointing normal vector on \(\partial \Omega_{\text{bc}}\). More precisely, we have to fix the orientation of the normal vector on the inner faces, since the normal vectors of two adjacent cells only differ in their sign, i.e., \(\mathbf{n}_{\lambda_1} = -\mathbf{n}_{\lambda_2}\).

Note that \(q \in S\) is not uniquely defined on an inner face \(e \in \Sigma_I\). For that reason, we define

\[
q^+(x) := \lim_{s \to 0^+} q(x + s \mathbf{n}_e), \quad q^-(x) := \lim_{s \to 0^-} q(x - s \mathbf{n}_e), \quad x \in \Gamma_I.
\]

Here, \(q^+\) is the value of the side in which the normal vector is pointing. We now define an average operator \{\cdot\} and a jump operator \([\cdot]\) on \(\Gamma_I\) by

\[
\{q\} := \frac{1}{2} (q^+ + q^-), \quad [q] = q^+ - q^-,
\]

and on \(\partial \Omega\) by \([q] := q^-\) and \([\{q\}] = q^-\), respectively. These operators can be extended to vector- or matrix-valued functions by applying them component-wise. Note that for \(p, q \in S\) there holds

\[
[[pq]] = [[[p]]q] + [p][[q]] \quad \text{on} \ \Gamma_I.
\]

Note that sometimes the alternative definition of the jump operator

\[
[[q]]^* := q|_{V_{\lambda_1}} \mathbf{n}_{\lambda_1} + q|_{V_{\lambda_2}} \mathbf{n}_{\lambda_2}, \quad [[[p]]^* := p|_{V_{\lambda_1}} \cdot \mathbf{n}_{\lambda_1} + p|_{V_{\lambda_2}} \cdot \mathbf{n}_{\lambda_2}
\]

for scalar-valued \(q\) and vector-valued \(p\) of length \(d\) is considered, for instance in [9]. This alternative jump of a scalar-valued quantity is a vector and the jump of a vector-valued quantity is a scalar. Thus, the use of this alternative is less flexible than the herein considered definition used by [39, 68]. In particular, when dealing with systems of conservations laws it is convenient to consider (vector-valued) jumps of vectors, e.g. for the vector of conserved quantities.
Cell-wise differentiation and integration by parts formula

Based on the partition (2.2), we introduce cell-wise differential operators, which are cell-wise equal to the differential operator defined on $\Omega$, i.e.,

$$
(\nabla_h \cdot \mathbf{u})|_{V_\lambda} := (\nabla \cdot \mathbf{u})|_{V_\lambda}, \quad (\nabla_h \mathbf{u})|_{V_\lambda} := (\nabla \mathbf{u})|_{V_\lambda}, \quad \forall \lambda \in \mathcal{I}.
$$

(2.9)

For $f, g \in L^2(K)$ and $\mathbf{f}, \mathbf{g} \in L^2(K, \mathbb{R}^d)$ the $L^2$-inner product on $K \subset \Gamma$ is denoted by

$$
\langle f, g \rangle_K := \int_K f g \, ds \quad \text{and} \quad \langle \mathbf{f}, \mathbf{g} \rangle_K := \int_K \mathbf{f} \cdot \mathbf{g} \, ds,
$$

(2.10)

respectively. Due to (2.7) there holds for $u \in S$ and $\mathbf{w} \in (S)^d$

$$
\sum_{\lambda \in \mathcal{I}} \int_{\partial V_\lambda} u \mathbf{w} \cdot \mathbf{n}_\lambda \, ds = \int_{\Gamma^-} [[u \mathbf{w} \cdot \mathbf{n}_\Gamma]] \, ds + \langle \mathbf{w} \cdot \mathbf{n}_\Gamma, u \rangle_{\partial \Omega_{\text{bc}}},
$$

(2.11)

where by $\mathbf{n}_\lambda$ the outward pointing unit normal vector corresponding to the cell $V_\lambda$ is denoted and $\langle \cdot, \cdot \rangle_{\partial \Omega_{\text{bc}}}$ is the $L^2$-inner product on $\partial \Omega_{\text{bc}}$. According to (2.8) this can be written as:

$$
\sum_{\lambda \in \mathcal{I}} \int_{\partial V_\lambda} u \mathbf{w} \cdot \mathbf{n}_\lambda \, ds = \langle [[u \mathbf{w} \cdot \mathbf{n}_\Gamma]], \{u\} \rangle_{\Gamma^-} + \langle \{u \mathbf{w} \cdot \mathbf{n}_\Gamma \}, [[u]] \rangle_{\Gamma^-} + \langle \mathbf{w} \cdot \mathbf{n}_\Gamma, u \rangle_{\partial \Omega_{\text{bc}}},
$$

(2.12)

Analogous expressions can be derived for vector-valued and matrix-valued functions. The $L^2$-inner product on a volume $V \subset \Omega$ is defined as usual for $f, g \in L^2(V)$ and $\mathbf{f}, \mathbf{g} \in L^2(V, \mathbb{R}^d)$ by

$$
\langle f, g \rangle_V := \int_V f g \, dx \quad \text{and} \quad \langle \mathbf{f}, \mathbf{g} \rangle_V := \int_V \mathbf{f} \cdot \mathbf{g} \, dx,
$$

(2.13)

respectively. Then, we derive an integration by parts formula for the cell-wise differential operators (2.9) based on jump and average operators (2.7) using (2.8). For $\mathbf{g} \in (S)^d$ and $v \in S$ there holds

$$
\langle \nabla_h \cdot \mathbf{g}, v \rangle_{\Omega} = -\langle \mathbf{g}, \nabla_h v \rangle_{\Omega} + \langle [[\mathbf{g} \cdot \mathbf{n}_\Gamma]], \{v\} \rangle_{\Gamma^-} + \langle \{\mathbf{g} \cdot \mathbf{n}_\Gamma \}, [[v]] \rangle_{\Gamma^-} + \langle \mathbf{g} \cdot \mathbf{n}_\Gamma, v \rangle_{\partial \Omega_{\text{bc}}},
$$

(2.14)

where the inner products are applied component-wise.

### 2.2 DG discretization of hyperbolic conservation laws

In this section we focus on the DG semi-discretization of systems of hyperbolic conservation laws

$$
\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = 0,
$$

(2.15)
in the space-time domain \((0, T) \times \Omega\) with \(\Omega \subset \mathbb{R}^d\) and \(T > 0\). These equations describe the evolution of a vector of \(m\) conserved quantities \(\mathbf{u} : [0, T] \times \Omega \to \mathcal{D}\), where by \(\mathcal{D} \subset \mathbb{R}^m\) we denote the set of all admissible states. The matrix \(\mathbf{F} : \mathcal{D} \to \mathbb{R}^{m \times d}\) comprises the fluxes \(\mathbf{f}_i : \mathcal{D} \to \mathbb{R}^m\) corresponding to the different coordinate directions, i.e.,

\[
\mathbf{F} = (\mathbf{f}_1, \ldots, \mathbf{f}_d).
\]

This problem has to be supplemented with initial data

\[
\mathbf{u}(0, \mathbf{x}) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x} \in \Omega,
\]

and suitable boundary conditions on the non-periodic part \(\partial \Omega_{bc}\) of \(\partial \Omega\). Examples for this type of equations are Burgers’ equation, shallow water equations, isentropic Euler equations as well as compressible Euler equations.

In the following we briefly summarize basic properties of hyperbolic conservation laws. For further details, we refer to classical text books such as [86]. First, we note that (2.15) can be rewritten in quasi-linear form as

\[
\frac{\partial \mathbf{u}}{\partial t} + \sum_{i=1}^{d} \mathbf{f}_i'(\mathbf{u}) \mathbf{u}_{x_i} = 0,
\]

where we denote by \(\mathbf{f}_i'(\mathbf{u}) = \frac{\partial \mathbf{f}_i}{\partial \mathbf{u}}\) the Jacobian of the fluxes \(\mathbf{f}_i\), \(1 \leq i \leq d\). By analyzing the structure of \(\mathbf{f}_i'(\mathbf{u})\), the type of (2.15) can be determined. In this thesis, we restrict ourselves to hyperbolic systems of conservation laws.

**Definition 2.1 (Hyperbolicity)** The system of conservation laws (2.15) is called hyperbolic, if for all \(\mathbf{u} \in \mathcal{D}\) and \(\omega \in \mathbb{R}^d \setminus \{0\}\) the matrix \(\mathbf{A}(\mathbf{u}, \omega)\) defined by

\[
\mathbf{A}(\mathbf{u}, \omega) := \sum_{i=1}^{d} \mathbf{f}_i'(\mathbf{u}) \omega_i,
\]

has only real eigenvalues \(\lambda_1, \ldots, \lambda_m\) and there exists a complete family of \(m\) linearly-independent right and left eigenvectors \(\mathbf{r}_1, \ldots, \mathbf{r}_m\) and \(\mathbf{l}_1, \ldots, \mathbf{l}_m\) corresponding to these eigenvalues, respectively.

Note that in the general non-linear and multi-dimensional case the eigenvalues \(\lambda_i = \lambda_i(\mathbf{u}, \omega)\) and eigenvectors \(\mathbf{r}_i = \mathbf{r}_i(\mathbf{u}, \omega)\) and \(\mathbf{l}_i = \mathbf{l}_i(\mathbf{u}, \omega)\), \(i = 1, \ldots, m\), depend on the direction \(\omega\) and the conserved quantities \(\mathbf{u}\). For ease of notation, we comprise the eigenvectors in matrices \(\mathbf{L} := (\mathbf{l}_1, \ldots, \mathbf{l}_m)^T\) and \(\mathbf{R} := (\mathbf{r}_1, \ldots, \mathbf{r}_m)\), respectively. Furthermore, we assume that the left and right eigenvectors are normalized, i.e., \(\|\mathbf{r}_i\|_2 = \|\mathbf{l}_i\|_2 = 1, 1 \leq i \leq m\). Then, for all \(\omega \in \mathbb{R}^d \setminus \{0\}\) there holds

\[
\mathbf{L}(\mathbf{u}, \omega) \mathbf{A}(\mathbf{u}, \omega) \mathbf{R}(\mathbf{u}, \omega) = \Lambda(\mathbf{u}, \omega) := \text{diag}(\lambda_1, \ldots, \lambda_m).
\]

Furthermore, \(\mathbf{L}\) coincides with the inverse of \(\mathbf{R}\), i.e., \(\mathbf{L} \mathbf{R} = \mathbf{I}\). Consequently, \(\mathbf{u}\) can be spanned by the right eigenvectors leading to local characteristic variables.
Conclusion 2.1 (Local characteristic variables) Let the system of conservation laws (2.15) be hyperbolic. Then \( \mathbf{u} \in \mathcal{D} \) can be spanned by the eigenvectors, i.e.,

\[
\mathbf{u} = \sum_{i=0}^{m} w_i \mathbf{r}_i = \mathbf{R}\mathbf{w},
\]

where the local characteristic variables \( \mathbf{w} = w(\mathbf{u}, \omega) \) in direction \( \omega \) are defined by \( \mathbf{w} = \mathbf{L}\mathbf{u} \).

In particular, for a linear system in one spatial dimension the characteristic variables (2.21) can be used to decouple the system, i.e.,

\[
\frac{\partial w_i}{\partial t} + \lambda_i \frac{\partial w_i}{\partial x} = 0, \quad 1 \leq i \leq m.
\]

Here, the solution of (2.21) is a superposition of translated initial data transformed to the characteristic variables, i.e.,

\[
\mathbf{u}(t, x) = \mathbf{R}\mathbf{w}(t, x), \quad \text{with} \quad w_i(t, x) = (\mathbf{L}\mathbf{u}_0(x - \lambda_i t))_i.
\]

Thereby, we can directly observe a typical behavior in hyperbolic conservation laws: information is transported with a finite speed of propagation. The local characteristic variables reveal insight in this behavior.

The speed of propagation remains finite in the more general non-linear case. However, the speed of propagation as well as the characteristic variables depend on the conserved quantities themselves. Thus, the solution is not only translated and even for smooth initial data discontinuities can develop within a final time. For that reason, the notion of solutions for (2.15) is extended to weak solutions, cf. [86]. In general, these weak solutions are not unique. Therefore, a particular solution is selected by augmenting (2.15) with an additional constraint; the entropy condition. This weak solution is called the entropy solution, cf. [86].

In the scalar case (\( \Omega = \mathbb{R}^d \) and \( m = 1 \)) it can be shown that there exists a unique entropy solution \( u \in L^\infty(0, T; L^1_{\text{loc}}(\mathbb{R}^d)) \) for all \( T > 0 \) provided that \( u_0 \in BV(\mathbb{R}^d) \cap L^\infty(\mathbb{R}^d) \cap L^1(\mathbb{R}^d) \) and \( f \in C^1(\mathbb{R}^d) \). Here, \( BV(\mathbb{R}^d) \) denotes the space of functions of bounded variation, cf. [86]. By assuming a small total variation of the initial data, global existence of an entropy solution has been shown for one-dimensional systems of equations, cf. [26, 30, 85]. For multi-dimensional systems, existence and uniqueness can only be shown for particular configurations, e.g. for Euler equations in [41]. No general theory is available for existence and uniqueness of entropy solutions. In fact, for the Euler equations there exist counterexamples disproving the conjecture of global uniqueness results, cf. [42, 65].

Boundary conditions

When equipping (2.15) with boundary conditions, the hyperbolic nature of the problem has to be respected, i.e., transport directions have to be taken into account: at an outflow
2.2. DG FOR HYPERBOLIC PROBLEMS

boundary no values can be prescribed. Hence, in general, Dirichlet boundary conditions
must not be prescribed, cf. [76]. If we want to prescribe values on $\partial \Omega_{bc}$, we have to
make sure that these are admissible and do not contradict the behavior of the equations.
Non-admissible boundary conditions may lead to an ill-posed problem. Due to the non-linearity
of $F$, the admissible states may depend on the solution themselves.

For this reason, we consider boundary values $u_{bc}$ depending on the (inner) boundary
value $u^-$ of the solution of (2.15), i.e., $u_{bc} : [0, T] \times \partial \Omega_{bc} \times D \to D$. Thus, the boundary
conditions become

$$u(t, x) = u_{bc}(t, x, u^-(t, x)), \quad \text{for almost every } x \in \partial \Omega_{bc}.$$  

One way to find admissible states is to consider the local characteristic decomposition
(2.21), cf. [170]. By computing the local characteristic decoupling (2.21) at the boundary,
those characteristic variables, for which values can be prescribed, can be determined. To
describe the external state $u_{ext} = u_{ext}(t, x)$ at the boundary, an admissible boundary
state $u_{bc}$ can be defined by

$$u_{bc} := R(u^-, n)\Lambda^+(u^-, n)L(u^-, n)u_{ext} + R(u^-, n)\Lambda^-(u^-, n)L(u^-, n)u^-, \quad (2.22)$$

where the diagonal matrices $\Lambda^-$ and $\Lambda^+$ are defined by $(\Lambda^\pm)_{ii} := \frac{1}{2}(\text{sign}(\lambda_i) \pm 1)$ and $n$
denotes the outward pointing unit normal vector on $\partial \Omega_{bc}$. In the following, we refer to
(2.22) as characteristic inflow boundary conditions.

Semi-discrete DG formulation

The first step in the discretization of (2.15) is to discretize in space with a DG approach.
For this purpose, we follow the ideas of Cockburn et al. [45–48]: each equation in (2.15) is
multiplied by a test function $v \in S$ and integrated over $\Omega$. Then, the cell-wise integration
by parts formula (2.14) is applied. Fluxes on the inner faces of the skeleton are substituted
by numerical fluxes $\hat{F}(u, n)$ for reasons of stability. Boundary conditions are incorporated
into the scheme in a weak sense: fluxes in the boundary integrals are evaluated with the
boundary state $u_{bc}$. This approach leads to a weak formulation defining the semi-discrete
DG solution:

**Definition 2.2 (Semi-discrete DG solution)**

We call $u_h(\cdot, t) \in S^m, t \in [0, T]$, the DG approximation of (2.15) with initial data (2.17)
and admissible boundary data $u_{bc}$, iff for all $v \in S$ there holds

$$\langle \frac{\partial u_h}{\partial t}, v \rangle_{\Omega} - \langle F(u_h), \nabla_h v \rangle_{\Omega} + \langle \hat{F}(u_h, n_G), [v] \rangle_{\Gamma_G} + \langle F(u_{bc}(t, \cdot, u_h^-)) n_G, v \rangle_{\partial \Omega_{bc}} = 0 \quad (2.23)$$

and

$$\langle u_h(0, \cdot), v \rangle_{\Omega} = \langle u_0, v \rangle_{\Omega}. \quad (2.24)$$
Here, the inner products (2.13) are applied component-wise for vectors, e.g., $\frac{\partial u_h}{\partial t}$, and row-wise for the matrix $F(u_h)$, respectively.

In order to stabilize the weak formulation (2.23), numerical viscosity is introduced by the numerical fluxes. Thus, the choice of numerical fluxes $\hat{F} = \hat{F}(u_h, n)$ in semi-discrete DG formulation affects the quality of the DG solution. We assume that the numerical flux can be written as a function $\hat{\mathbf{F}}$ depending on the inner and outer value $u_h^+$ and $u_h^-$, respectively, i.e.,

$$\hat{F}(u_h, n) = \hat{\mathbf{F}}(u_h^+, u_h^-, n).$$

The numerical flux is assumed to be consistent, i.e.,

$$\hat{\mathbf{F}}(w, w, n) = F(w)n, \quad \forall w \in \mathcal{D},$$

and Lipschitz-continuous in the first two arguments. We furthermore assume that the numerical flux is an E-flux if applied to a scalar problem ($m = 1$), i.e., for all $u^-, u^+ \in \mathcal{D}$ there holds

$$(\hat{\mathbf{F}}(u^+, u^-, n) - F(v)n)(u^- - u^+) \geq 0, \quad v \in \{s u^- + (1-s) u^+ : s \in [0, 1]\}.$$

From this, the $L^2$-stability of the semi-discrete formulation can be derived in the scalar case ($m = 1$), cf. [115].

**Property 1 (Energy estimate)** Let $m = 1$, $0 \in \mathcal{D}$, $F \in C^1(\Omega, \mathcal{D})$ and the numerical flux $\hat{F}$ be an E-flux. Then, the semi-discrete DG solution (2.23) with boundary conditions (2.22) using $u_{\text{ext}} = 0$ is $L^2$-stable, i.e.,

$$\|u_h(t, \cdot)\|_{L^2(\Omega)} \leq \|u_h(0, \cdot)\|_{L^2(\Omega)}, \quad \forall t \in (0, T).$$

For a detailed proof of Property 1 we refer to [68]. A comparably simple choice for the numerical flux, satisfying the desired properties of being consistent, Lipschitz-continuous and an E-flux, is the modified local Lax-Friedrichs flux with stabilizing parameter $0 \leq \alpha \leq 1$ [39]

$$\hat{F}(u, n) = \{F(u)n\} - \alpha \frac{\lambda_{\text{max}}}{2} \|[u]\],$$

where $\lambda_{\text{max}}$ is a local estimate of the spectral radius of $A(u, n)$. By the choice of $\alpha$, the amount of numerical viscosity can be controlled. For $\alpha = 1$ this flux coincides with the classical local Lax-Friedrichs flux. Thus, $\alpha = 1$ is a suitable choice for most situations. However, when computing turbulent flows it might be relevant to reduce the amount of numerical viscosity in the scheme. In these situations, $\alpha < 1$ might be a preferable choice.

Alternatively, approximate Riemann solvers such as Roe’s Riemann solver [151] or the HLL solver [100] can be used. Furthermore, an upwind flux based on local decoupling
via characteristic variables (2.21) has been proposed in [118]. These alternatives are computationally more expensive than the local Lax-Friedrichs flux which in our experience provides reasonable results and which we thus usually employ.

In some situations using simply the flux evaluated at the boundary state $u_{bc}$ in the boundary integrals of (2.23) might result in stability problems. For that reason, it is advisable to stabilize these boundary integrals with a numerical flux, i.e., we replace \( \langle F(u_{bc}(\cdot, u_h^{-})), n_{\Gamma}, v \rangle_{\partial \Omega_{bc}} \) by
\[
\langle \tilde{F}(u_{bc}(t, \cdot, u_{h}^{-}), u_{h}^{-}, n_{\Gamma}), v \rangle_{\partial \Omega_{bc}}.
\] (2.25)

However, for several configurations this additional modification is not necessary. Thus, we consider both approaches for the implementation of the boundary conditions in our numerical simulations.

### 2.3 Extension to convection-dominated conservation laws

In this section we are interested in applying a RK-DG scheme to the compressible Navier Stokes equations. These equations are an extension of the compressible Euler equations including additional viscous and heat conduction terms. Thus, these equations can be written in the form
\[
\frac{\partial u}{\partial t} + \nabla \cdot F(u) - \nabla \cdot G(u, \nabla u) = 0,
\] (2.26)

where by $G : \mathcal{D} \times \mathbb{R}^{mxd} \rightarrow \mathbb{R}^{mxd}$ we denote the additional viscous flux.

Since we are interested in dealing with problems revealing attributes of hyperbolic equations such as shocks, the numerical discretization has to be chosen carefully in order to obtain a stable scheme. For that purpose, we proceed similarly to the hyperbolic case and multiply each equation of (2.26) with a test function $v \in \mathcal{S}$ and integrate over $\Omega$. Then, the cell-wise integration by parts formula (2.14) is applied. Similarly to the pure hyperbolic case, the fluxes have to be stabilized.

For the stabilization of the inviscid part we proceed as described in Section 2.2 and substitute inviscid fluxes $F$ in the surface integrals by numerical fluxes $\hat{F}$. In order to stabilize the viscous fluxes $G$, several alternative stabilization techniques are available, e.g., the local discontinuous Galerkin method [50], interior penalty methods [8,74] as well as the BR2-scheme by Bassi et al. [17–19]. For a detailed review of DG approaches for elliptic and parabolic problems, we refer to [9].

Here, we employ the BR2-scheme by Bassi et al. [17–19] since it has originally been introduced for the compressible Navier-Stokes equations and has been successfully applied for the simulation of laminar and turbulent flows, cf. [17–19,39,177]. For convenience of
the reader, we briefly summarize the main ideas. The basic idea is to stabilize the scheme by penalizing jumps at interfaces. For that purpose, gradients in the viscous fluxes $\mathbf{G}$ in (2.26) are corrected by a lifting term $\mathbf{R}_{\text{br}}([\mathbf{u}])$. The lifting is defined by a global lifting operator

$$
\mathbf{R}_{\text{br}} : (L^2(\Gamma))^m \to (S)^m \times (S)^m, \quad \mathbf{w} \mapsto \mathbf{R}_{\text{br}}(\mathbf{w}),
$$

(2.27)

consisting of the sum of local lifting operators $\mathbf{R}_e$ associated to the faces $e \in \mathcal{E}$, i.e.,

$$
\mathbf{R}_{\text{br}}(\mathbf{w}) = \sum_{e \in \mathcal{E}} \mathbf{R}_e(\mathbf{w}).
$$

The local lifting $\mathbf{R}_e(\mathbf{w})$ associated to a face $e$ is defined by

$$
\langle \mathbf{R}_e(\mathbf{w}), \eta \rangle_{\Omega_e} = -\langle \mathbf{w} \otimes \mathbf{n}, \{\eta\} \rangle_e \quad \forall \eta \in (S)^d,
$$

(2.28)

where $\Omega_e$ is the union of all cells sharing the face $e$. The gradient $\nabla_h \mathbf{u}$ in the viscous flux $\mathbf{G}$ is stabilized in the volume integrals using the global lifting operator $\mathbf{R}_{\text{br}}([\mathbf{u}])$. In the surface integrals the gradient is stabilized with the local lifting operator $\mathbf{R}_e([\mathbf{u}])$ in order to guarantee locality of the scheme. Again, the flux $\mathbf{G}$ on the inner faces $e$ is substituted by viscous numerical fluxes $\mathbf{G}$ chosen as

$$
\mathbf{G}(\mathbf{u}, \nabla_h \mathbf{u} + \mathbf{R}_e([\mathbf{u}]), \mathbf{n}) = \{\mathbf{G}(\mathbf{u}, \nabla_h \mathbf{u} + \mathbf{R}_e([\mathbf{u}])) \mathbf{n}\}.
$$

In analogy to the pure hyperbolic case, boundary conditions are incorporated into the DG scheme in a weak sense. In order to retain flexibility in realizing boundary conditions for viscous problems, $\mathbf{G}$ is substituted in the boundary integrals by

$$
\mathbf{G}_{bc}(u_{bc}, u_h, \nabla u_h + \mathbf{R}_e(u_{bc} - u_h^\tau)),
$$

which realizes the particular boundary condition. Consequently, we define the semi-discrete DG solution of (2.26):

**Definition 2.3 (Semi-discrete DG solution)**

We call $\mathbf{u}_h(\cdot, t) \in S^m, t \in [0, T]$, the DG approximation of (2.26) iff for all $\mathbf{v} \in S$ there holds

$$
\langle \frac{\partial \mathbf{u}_h}{\partial t}, \mathbf{v} \rangle_\Omega - \langle \mathbf{F}(\mathbf{u}_h) - \mathbf{G}(\mathbf{u}_h, \nabla \mathbf{u}_h + \mathbf{R}_{\text{br}}([\mathbf{u}_h])), \nabla_h \mathbf{v} \rangle_\Omega
$$

$$
+ \sum_{e \in \mathcal{E}} \langle \mathbf{F}(\mathbf{u}_h(t), \mathbf{n}_e) - \mathbf{G}(\mathbf{u}_h(t), \nabla \mathbf{u}_h + \mathbf{R}_e([\mathbf{u}_h]), \mathbf{n}_e), \{\mathbf{v}\} \rangle_{\Gamma_e}
$$

$$
+ \sum_{e \in \mathcal{E}_{bc}} \langle \mathbf{F}(\mathbf{u}_{bc}(t, \cdot), \mathbf{u}_h), \mathbf{n}_e, \mathbf{v} \rangle_e
$$

$$
- \sum_{e \in \mathcal{E}_{bc}} \langle \mathbf{G}_{bc}(\mathbf{u}_{bc}(t, \cdot), \mathbf{u}_h + \mathbf{R}_e(\mathbf{u}_{bc}(t, \cdot) - \mathbf{u}_h^\tau)), \mathbf{n}_e, \mathbf{v} \rangle_e = 0
$$

(2.29)

and

$$
\langle \mathbf{u}_h(0, \cdot), \mathbf{v} \rangle_\Omega = \langle \mathbf{u}_0, \mathbf{v} \rangle_\Omega.
$$

(2.30)
2.4 Basis functions and evolution equations for the coefficients

In this section we introduce an orthogonal basis for the DG space $S$. Then, we derive an evolution equation for the coefficients in the basis expansion.

Orthonormal basis

In order to derive fully-discrete schemes from the semi-discrete DG schemes (2.23) and (2.29) we have to specify a basis for the DG space $S$, i.e.,

$$S = \text{span}_{\lambda \in \mathcal{I}, i \in \mathcal{P}} \phi_{\lambda,i},$$

where the local degrees of freedom of $\Pi_{p-1}$ are enumerated with the index set $\mathcal{P}$. For efficiency reasons we assume that the basis functions are compactly supported, i.e.,

$$\text{supp}(\phi_{\lambda,i}) = V_\lambda, \quad \forall \lambda \in \mathcal{I}, \ i \in \mathcal{P}. \quad (2.31)$$

For reasons of stability it is convenient to consider an orthonormal basis, i.e.,

$$\langle \phi_{\lambda,i}, \phi_{\lambda,i'} \rangle_\Omega = \delta_{\lambda,i} \delta_{\lambda,i'}, \quad \forall \lambda, \lambda' \in \mathcal{I}, \ i, i' \in \mathcal{P}. \quad (2.32)$$

Furthermore, we claim that the zeroth scaling basis is constant, i.e.,

$$\phi_{\lambda,0} = 1/\sqrt{|V_\lambda|}, \quad \forall \lambda \in \mathcal{I}. \quad (2.33)$$

Finally, we expand the DG solutions of (2.23) and (2.29) in this basis, respectively, i.e.,

$$u_h(t, x) = \sum_{\lambda \in \mathcal{I}} \sum_{i \in \mathcal{P}} u_{\lambda,i}(t) \phi_{\lambda,i}(x), \quad (2.34)$$

where the coefficients are defined by $u_{\lambda,i}(t) := \langle u_h(\cdot, t), \phi_{\lambda,i} \rangle_{V_\lambda}$. Since the basis is not time-dependent, the temporal evolution of the DG solution is described by the evolution of the coefficients.

Evolution equations for coefficients

Note that the weak formulations defining the semi-discrete DG solutions (2.23) and (2.29), respectively, can be written in the following form:

$$\langle \frac{\partial u_h}{\partial t}, v \rangle_\Omega = \hat{L}(u_h, v) \quad \forall \ v \in S, \quad (2.35)$$

where $\hat{L} : S^m \times S \to \mathbb{R}^m$ comprises the flux integrals. Thereby, we derive an evolution equation for the coefficients in the basis expansion of the solution (2.34). For that purpose,
we choose a single basis function $\phi_{\lambda_i}$ for $v$ in (2.35). Then, by using the orthogonality of the basis (2.32), we obtain the system of ordinary differential equations in time

$$U'(t) = \mathcal{L}(U),$$  \hspace{1cm} (2.36)

where all coefficient vectors $u_{\lambda,i}$ and right-hand sides $\hat{\mathcal{L}}$ are comprised in

$$U:= (u_{\lambda,i})_{\lambda \in I, i \in \mathbb{P}}, \quad \mathcal{L}(U) := (\hat{\mathcal{L}}(u_{h, \phi_{\lambda,i}}))_{\lambda \in I, i \in \mathbb{P}}.$$

Thus, the temporal evolution of the DG solutions (2.23) and (2.29), respectively, is specified by a system of ordinary differential equations in time for the coefficients of the basis expansion.

### 2.5 Temporal discretization and limiting

In this section we finally derive the fully-discrete DG scheme by discretizing and stabilizing the evolution equation for the coefficients (2.36). For that purpose, we discretize $[0,T]$ by discrete time levels $\{t_n\}_{n=0}^N$. At each time level the semi-discrete DG solution $u_h(t_n, \cdot)$ is approximated by $u_h^n(\cdot)$. We denote by $\overline{U^n}$ the coefficients in the basis expansion of $u_h^n(\cdot)$.

**Runge-Kutta schemes**

In order to compute these approximations, we consider strong-stability-preserving Runge-Kutta (SSP-RK) schemes [90]. A time step of an $s$-stage SSP-RK scheme applied to (2.36) can be written as

$$\begin{align*}
U^{n+1,0} &= U^n, \\
U^{n+1,i} &= \sum_{k=0}^{i-1} \alpha_{i,k} U^{n+1,k} + \Delta t \beta_{i,k} \mathcal{L}(U^{n+1,k}), & 1 \leq i \leq s, \\
U^{n+1} &= U^{n+1,s},
\end{align*}$$  \hspace{1cm} (2.37)

defined by the lower triangular coefficient matrices $(\alpha_{i,k})_{1 \leq i,k \leq s}$, and $(\beta_{i,k})_{1 \leq i,k \leq s}$. It is assumed that $\alpha_{i,k} = 0$ whenever $\beta_{i,k} = 0$.

In order to avoid backward time stepping, it is preferable that all coefficients $\beta_{i,k}$ and $\alpha_{i,k}$ are non-negative. Additionally, for reasons of consistency it is assumed that

$$\sum_{k=0}^{i-1} \alpha_{i,k} = 1, \quad 1 \leq i \leq s.$$

Based on these assumptions, it can be shown that each stage consists of convex combinations of explicit forward Euler steps. From an analytical point of view this is very
2.5. TEMPORAL DISCRETIZATION AND LIMITING

convenient: if stability has been proven for the forward Euler step, it is maintained for all stages through the complete Runge-Kutta step \([89]\). However, from a computational point of view, these methods cause disadvantages when high-order time discretizations are required: for \(p > 3\) it is not known, whether optimal schemes, i.e., \(p\) stages for \(p\)th order, with non-negative coefficients exist. For \(p < 4\) optimal SSP-RK schemes with non-negative coefficients have been derived in \([163]\). In fact, Gottlieb et al. have shown in \([89]\) that a fourth-order scheme with four stages has at least one negative \(\beta_{i,k}\). A fourth-order scheme with non-negative coefficients consisting of five stages can be found in \([154]\). Consequently, for reasons of efficiency it might be convenient to consider standard Runge-Kutta schemes if a high-order time discretization is required.

CFL condition

Using an explicit time discretization requires a limitation of the time step size \(\Delta t\). If the time step size is chosen too large, the scheme becomes unstable. In case of a purely hyperbolic problem the largest possible time step size is limited by the well-known Courant-Friedrichs-Lewy (CFL) condition \([45, 47]\):

\[ \Delta t \leq \frac{C_{\text{CFL}}}{C_{\text{hyp}}} \Delta x, \tag{2.38} \]

where the CFL number \(C_{\text{CFL}}\) depends on the polynomial degree and \(C_{\text{hyp}}\) is a bound for the spectral radius of the Jacobian of \(\mathbf{F}\). In the nonlinear case this depends on the current solution and is recomputed in each time step. The presence of viscous terms results in the additional more restrictive constraint

\[ \Delta t \leq C_{\text{visc}} \Delta x^2, \tag{2.39} \]

where \(C_{\text{visc}}\) is related to the amount of viscosity in the equations.

Local projection limiting

In order to control oscillations near discontinuities, which typically arise in high-order schemes for convection-dominated problems, we have to stabilize the fully discrete DG scheme. Therefore, either artificial viscosity is added, cf. \([12, 16, 139, 192]\), or high-order coefficients are locally modified by a projection limiter after each stage of the Runge-Kutta scheme \((2.37)\). The basic idea of limiting consists of two steps: first an indicator is used to detect troubled cells, in which the local polynomial will be corrected. Then, in a second step, polynomials in troubled cells are replaced by modified (limited) polynomials. Thereby, local mean values will remain unchanged in order to ensure conservation.

Nowadays, a wide variety of limiters is available for RK-DG methods. Originating from Finite-Volume methods a minmod-type limiter \([95]\) has been initially adopted to
DG schemes, cf. [45–48]. The key idea in this limiter is to use a modified minmod function [161]

\[
\tilde{m}(a_1, \ldots, a_m) = \begin{cases} 
  a_1, & \text{if } |a_1| \leq M \Delta x^2, \\
  \text{minmod}(a_1, \ldots, a_m), & \text{otherwise}, 
\end{cases}
\]

(2.40)

for the detection of troubled cells and the actual limiting process. In order to avoid limiting at critical points for smooth solutions, the so-called Shu constant \(M > 0\) was introduced. Both, indication of troubled cells and the following limiting process act only on the linear part of the solution and neglect all high order coefficients in a cell if it has been detected for limiting. Based on the minmod limiter, moment limiters have been developed [27, 33, 122]. In contrast to the original minmod limiter, these act on all high order coefficients, i.e., all coefficients are limited. Recently, limiters based on WENO reconstructions for the modified (limited) polynomials have been proposed [145, 190]. For a detailed overview and a comparison of several limiters, we refer to [144, 191].

We follow the idea of Cockburn and Shu [46, 48] and apply the original minmod-type limiter. We have chosen this limiter, since it is comparably simple, computationally efficient and effectively suppresses oscillations. For the description of the limiter we restrict ourselves to quadrilateral grids. An adapted version of this limiter can also be applied for triangular grids, cf. [48]. The basic idea is to consider the local \(\Pi_1\)-projection of \(u_h\), i.e.,

\[
P_{\Pi_1}(u_h)(x) = \bar{u}_\lambda + \sum_{i=0}^{d} u^i_\lambda \frac{2(x_i - (x_\lambda)_i)}{\Delta x_i}, \quad \text{on } V_\lambda,
\]

(2.41)

where by \(\bar{u}_\lambda\) we denote the mean value of \(u_h\) on \(V_\lambda\), by \(x_\lambda\) the center of \(V_\lambda\) and by \(\Delta x_i\) the width of \(V_\lambda\) in direction \(i\). The slopes \(u^i_\lambda\) in different directions are limited independently. For that purpose, the system is locally transformed for each direction to the characteristic variables by using (2.21). Subsequently, the modified minmod function (2.40) is applied component-wise. Finally, the system is transformed back to the conserved quantities. The entire algorithm for this limiter is listed in Algorithm 2.1.

When limiting has to be applied to cells adjacent to the boundary \(\partial\Omega_{bc}\), then for the corresponding direction only the difference of the mean values of the inner cells is considered in line 4 of Algorithm 2.1, i.e., \(\tilde{w}^i_\lambda = \text{minmod}(w^i_\lambda, \bar{w}_\lambda^+ - \bar{w}_\lambda)\) and \(\tilde{w}^i_\lambda = \text{minmod}(w^i_\lambda, \bar{w}_\lambda - \bar{w}_\lambda^-)\), respectively.
Algorithm 2.1 Multi-dimensional minmod limiter of Cockburn and Shu [48]

1: Determine \( \Pi_1 \)-projection of \( u_h \) on the cell \( V_\lambda \).
2: for all \( i = 1, \ldots, d \) do
3: Find the mean values \( \overline{u}_{\lambda, i} \) and \( \overline{u}_{\lambda, i} \) of the neighboring cells in direction \( i \).
4: Transform to local characteristic variables in direction \( i \) w.r.t. \( \overline{u}_\lambda \) using (2.21):
   \[
   \overline{w}_\lambda := L(\overline{u}_\lambda, n_i) u_\lambda^{(0)}, \quad \overline{w}_{\lambda, \pm} := L(\overline{u}_\lambda, n_i) \overline{u}_{\lambda, \pm}, \quad w_\lambda^i := L(\overline{u}_\lambda, n_i) u_\lambda^i,
   \]
5: Compute modified values \( \tilde{w}_\lambda^i \) using the modified minmod function:
   \[
   \tilde{w}_\lambda^i = \tilde{m}(w_\lambda^i, \overline{w}_\lambda - \overline{w}_{\lambda, -}, \overline{w}_{\lambda, +} - \overline{w}_\lambda).
   \]
6: end for
7: Transform to conserved variables using (2.21):
   \[
   \tilde{u}_\lambda(x) = \overline{u}_\lambda + \sum_{i=0}^d R(\overline{u}_\lambda, n_i) \tilde{w}_\lambda^i \frac{2(x_i - (x_\lambda)_i)}{\Delta x_i}, \quad x \in V_\lambda.
   \]
8: If one of the slopes has been modified, i.e., \( \tilde{u}_\lambda \neq P_{\Pi_1}(u_h) \), set \( u_h|_{V_\lambda} = \tilde{u}_\lambda \).

Convergence of the fully discrete DG scheme

The convergence analysis of the fully discrete RKDG schemes can be considered as incomplete. Only for the scalar case, i.e., \( m = 1 \), convergence has been shown in the mean at least for the one-dimensional case, cf. [47]. To the best of the author’s knowledge, for the more general cases considering systems and/or multiple spatial dimensions, a proof of convergence does not yet exist. For smooth solutions of scalar conservation laws optimal convergence rates of the second and third order SSP-RK schemes have been derived in [186,187].

Implementation aspects

When implementing a fully discrete RKDG scheme, integrals have to be computed in order to evaluate \( L \). In particular, \( L \) consists of volume integrals over the cells \( V_\lambda \) and surface integrals over the cell boundaries \( \partial V_\lambda \). Since the fluxes in (2.29) and (2.23) are (possibly) non-linear, we use quadrature rules for the computation of these integrals. Let \( X^\text{vol}_\lambda \) denote the set of quadrature nodes for the approximation of the volume integral over \( V_\lambda \). Analogously, we denote by \( X^\text{surf}_\lambda \) the set of quadrature nodes for the approximation of the surface integral over \( \partial V_\lambda \). We collect all quadrature nodes corresponding to the cell \( V_\lambda \) in

\[
X_\lambda := X^\text{surf}_\lambda \cup X^\text{vol}_\lambda.
\]
For the computations presented in this thesis, we use quadrature rules obtained by a
tensor-product of one-dimensional Gaussian quadratures.

2.6 Numerical results

In order to validate our implementation of the DG scheme, we focus on a few selected
well-known numerical test cases where the exact solution is known. In particular, we
consider only smooth test cases in this section to verify that our implementation reaches
the desired accuracy. For that purpose, we perform computations with different grids
and compute the empirical order of convergence (eoc). A validation for non-smooth
test cases is performed in Chapter 4 together with the validation of the adaptive scheme.
Moreover, we validate the implementation for the compressible Navier-Stokes equations by
performing direct numerical simulations of turbulent flows. To this end, we consider two
configurations; the Taylor-Green vortex and a turbulent channel flow between isothermal
walls. For these configurations reference data for the validation are available.

Simulation parameters, unless stated otherwise, are as follows. We consider quadri-lateral Cartesian grids with uniform cells. The computations are performed for different
polynomial order $p$. For the time discretization an explicit Runge-Kutta scheme with
the same order $p$ and $C_{CFL} = 0.1$ is used. In particular, for $p \leq 3$ we consider strong-stability preserving Runge-Kutta schemes and for reasons of efficiency classical Runge-
Kutta schemes are used for higher orders, i.e., $p > 3$. In order to avoid limiting in regions,
where the solution is smooth, we consider a Shu constant of $M = 50$ in the troubled cell
indicator. As foundation for the multi-dimensional quadrature rule the (one-dimensional)
Gaussian quadrature with $p+1$ nodes is used. The computations are performed in parallel
using Message Passing Interface (MPI) in order to reduce the computational cost.

2.6.1 Linear convection-diffusion equation

We start with a comparably simple scalar problem: the linear convection-diffusion equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (bu) - \mu \Delta u = 0, \quad \text{on } [0, 1]^d \times (0, T], \quad (2.43)$$

with initial data

$$u(0, x) = \prod_{i=1}^{d} \sin(2\pi x_i), \quad x \in [0, 1]^d,$$

and periodic boundary conditions. Here, $b$ denotes the direction of convection and $\mu$
specifies the magnitude of diffusion. The exact solution of (2.43) is smooth and can be
specified explicitly: \[ u(t, x) = e^{-4\pi^2 \mu t} u_0(x - bt). \] Due to the absence of boundary conditions, no boundary layer develops and consequently the initial data are merely transported and damped. Thus, we verify the accuracy of our scheme by computing the numerical order of accuracy for the pure hyperbolic case using \( \mu = 0 \) and the parabolic case using \( \mu = 10^{-2} \). For this purpose, we have computed several DG solutions up to \( T = 0.5 \) for different grid resolutions: \( [T] = 2^L d \) for \( 1 \leq L \leq 7 \), i.e., \( 2^L \) cells in each direction. Here, \( L \) characterizes the level of uniform refinement in the grid. In Figure 2.1 the error in \( L_1 \) at \( T = 0.5 \) is plotted over the level of refinement \( L \). For both cases the empirical order of convergence is reaching \( p \) for the \( p \)th order scheme with increasing refinement, \( 2 \leq p \leq 5 \), as can be seen from Figure 2.1.

![Graphs showing L-error vs L for inviscid and viscous cases](image)

(a) inviscid (\( \mu = 0 \))  (b) viscous (\( \mu = 0.01 \))

Figure 2.1: Linear convection-diffusion equation: validation of coc for \( d = 2 \) for the inviscid and the viscous case, i.e, \( \mu = 0 \) and \( \mu = 10^{-2} \), respectively. The optimal order of convergence is plotted in dashed gray for comparison.

### 2.6.2 Compressible Euler equations

In order to validate the implementation for non-linear systems, we exemplarily consider the unsteady compressible Euler equations

\[
\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p \mathbf{I} \end{pmatrix} = 0.
\]  

(2.44)

These equations express conservation of mass, momentum and total energy density. Here, \( \rho, \mathbf{v}, E = e + \frac{1}{2} \mathbf{v}^2 \) and \( e \) denote the density, the velocity, the specific total energy and the specific internal energy, respectively. The system is closed by the equation of state for the pressure \( p = \rho e (\gamma - 1) \) for a thermally and calorically perfect gas. Here \( \gamma = 1.4 \) is the ratio
of specific heats for air at standard conditions [6]. We consider a configuration where the solution is explicitly known and smooth: a moving vortex given by the initial data

\[
\begin{align*}
\rho(0, x) &= \left(1 - \frac{25(\gamma - 1)}{8\gamma \pi^2} e^{1-(x_2)} \right)^{1/(\gamma - 1)}, \\
v(0, x) &= \left(\frac{1}{1} + \frac{5 e^{0.5-(x_2)}}{2\pi} \right) \left(\frac{-x_2}{x_1}\right), \\
\rho E(0, x) &= \frac{\rho(0, x)^\gamma}{\gamma - 1} + \frac{\rho(0, x)}{2} \mathbf{v}(0, x) \cdot \mathbf{v}(0, x).
\end{align*}
\]

This vortex is transported during the time evolution, i.e., the exact (entropy) solution is specified by \( \mathbf{u}(t, x) = \mathbf{u}_0(x_1 - t, x_2 - t) \), see [183].

To avoid unphysical reflections from the boundaries we apply characteristic boundary conditions using the exact solution. The computations are performed on the domain \( \Omega = [-5, 10] \times [-5, 10] \) until \( T = 5 \). In Figure 2.2 the \( L_1 \)-error of the density is plotted over the level of refinement \( L \). The results in Figure 2.2 confirm that the desired order of convergence is achieved for \( 2 \leq p \leq 5 \).

![Graph](image)

Figure 2.2: Moving vortex for compressible Euler equations: validation of eoc. The optimal order of convergence is plotted in dashed grey for comparison.
2.6. NUMERICAL RESULTS

2.6.3 Compressible Navier-Stokes equations

Next, we consider the compressible Navier-Stokes equations in dimensionless form, cf. [176]:

\[
\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p \mathbf{I} \\ \rho \mathbf{v} (E + p/\rho) \end{pmatrix} - \frac{1}{Re} \nabla \cdot \begin{pmatrix} 0 \\ \sigma \mathbf{v} - q \end{pmatrix} = 0. \tag{2.45}
\]

The dimensionless quantities are density \( \rho \), momentum \( \rho \mathbf{v} \) and total energy density \( \rho E \).

The dimensionless pressure \( p \) and the dimensionless temperature \( T \) are defined by

\[
p := (\gamma - 1) \left( \rho E - \frac{1}{2} \rho \mathbf{v}^2 \right) \quad \text{and} \quad T := E - \frac{1}{2} \mathbf{v}^2.
\]

By \( Re \) we denote the reference Reynolds number. The shear stress tensor is determined by

\[
\sigma = \mu_{\text{vis}}(T) \left( \nabla \mathbf{v} + (\nabla \mathbf{v})^T - \frac{2}{3} \nabla \cdot \mathbf{v} I \right) \tag{2.46}
\]

with a temperature dependent dynamic viscosity \( \mu_{\text{vis}} \). The heat flux \( q \) is defined by

\[
q = -\frac{\lambda_{\text{cond}}(T) \gamma}{Pr} \nabla T
\]

with a constant Prandtl number \( Pr \), heat capacity ratio \( \gamma \) and temperature dependent thermal conductivity \( \lambda_{\text{cond}} \). The non-dimensionalization is performed such that there holds \( \lambda_{\text{cond}}(T) = \mu_{\text{vis}}(T) \). The viscosity coefficient is specified by the power law \( \mu_{\text{vis}}(T) = (T/T_{\text{ref}})^{\gamma - 1} \) where the reference temperature \( T_{\text{ref}} = 1/(Ma^2 \gamma (\gamma - 1)) \) is based on the reference Mach number \( Ma \) and the heat capacity ratio \( \gamma \).

In order to validate the implementation of the DG scheme for compressible Navier-Stokes equations we have tested the order of convergence using manufactured solutions, i.e., we enforced a solution using a suitable source term, cf. [103]. For reason of clarity we abstain from presenting these results here. Moreover, we validate our implementation by performing direct numerical simulations (DNS) of turbulent flows. It is not straightforward to perform turbulent simulations. Slight errors or bugs in the implementation can prevent the flow from becoming turbulent. Moreover, the capability of a solver to reproduce turbulence is a good indicator for the amount of numerical viscosity that is used to stabilize the scheme. If a scheme is too diffusive the flow might remain laminar and turbulence is not observed although it is to be expected physically. Moreover, for the realization of the DNS high-order schemes are required, i.e., \( p > 3 \). Thus, a DNS of a turbulent flow using compressible Navier-Stokes equations is computationally highly expensive and requires an efficient implementation as well as an efficient parallelization.
of the DG scheme. Hence, performing a DNS is in many ways a challenging task and a good alternative for the validation of numerical solvers for the compressible Navier-Stokes equations.

To this end, we perform here two direct numerical simulation of a turbulent flow to validate our implementation of the DG scheme: we consider the Taylor-Green vortex and the DNS of a turbulent channel flow to investigate wall-bounded turbulence. Both test cases are well-known benchmark test cases for turbulent flows for which reference data for comparison are available.

**Taylor-Green vortex with $Re = 1600$**

When considering the computation of turbulent flows, the Taylor-Green vortex is a simple model problem for turbulent flows: the computational domain is cube, the initial conditions are relatively simple, only periodic boundary conditions are considered and no driving force is required in the equations. Additionally, a constant viscosity $\mu = 1$ and zero bulk viscosity are assumed, i.e.,

$$\sigma = \left( \nabla \mathbf{v} + (\nabla \mathbf{v})^T \right).$$

(2.47)

This configuration is frequently used as a benchmark problem for CFD workshops such as the Third International Workshop on High-order Methods in 2015, cf. [179]. It has been computed by several groups with different numerical solvers, see for instance [32,37,40,79]. Thus, reference data are available for validation, cf. [179]. This test case was originally used for the incompressible Navier-Stokes equations but nowadays it is often used to validate compressible codes by using a small Mach number, cf. [32,69]. On $\Omega = [-\pi, \pi]^d$, the initial conditions are specified in terms of density $\rho$, velocity $\mathbf{v}$ and pressure $p$ by

$$\rho(0, \mathbf{x}) = 1,$$

$$v_1(0, \mathbf{x}) = \sin(x_1) \cos(x_2) \cos(x_3),$$

$$v_2(0, \mathbf{x}) = -\cos(x_1) \sin(x_2) \cos(x_3),$$

$$v_3(0, \mathbf{x}) = 0,$$

$$p(0, \mathbf{x}) = p_0 + \frac{1}{16} \left( \cos(2x_1) + \cos(2x_2) \right) \left( \cos(2x_3) + 2 \right),$$

where $p_0$ is chosen such that the (bulk) Mach number of the flow is $0.1$. We consider a Reynolds number of 1600.

Starting from the initial data, small scale effects are generated by vortex stretching causing the transition to a turbulent flow. For a detailed description of this flow, we refer to [28]. For validation purposes the temporal evolution of the following quantities is recorded during the simulation: the dissipation rate of kinetic energy

$$\epsilon := -\frac{\partial}{\partial t} \left( \frac{1}{\rho_0 V_0^2 |\Omega|} \int_{\Omega} \rho \mathbf{v} \cdot \mathbf{v} \frac{1}{2} d\mathbf{x} \right)$$

(2.48)
and the enstrophy

$$
\varepsilon := \frac{1}{\rho_0 V_0^2 |\Omega|} \int_\Omega \frac{\omega \cdot \omega}{2} \, dx,
$$

(2.49)

with vorticity $\omega := \nabla \times v$. For these quantities reference data are provided from a spectral code in [179].

We have performed three computations: two computations using a fourth-order scheme, i.e., $p = 4$; one on a coarse grid consisting of $32^3$ cells and another one on a finer grid consisting of $64^3$ cells. In order to investigate the influence of the polynomial order we have performed an additional computation using a sixth-order scheme on a grid consisting of $32^3$ cells. Here we have applied a tensor-product space for the local polynomial space on each cell, i.e., $|P| = p^d$. In order to compare the different results we consider the degrees of freedom of the different computations, namely $128^3$ and $256^3$ for the $p = 4$ computations and $192^3$ for the $p = 6$ computation.

![Graphs showing kinetic energy dissipation rate and enstrophy](image)

(a) kinetic energy dissipation rate $\varepsilon$

(b) enstrophy $\varepsilon$

Figure 2.3: Taylor-Green vortex: comparison with reference data.

In Figure 2.3 the time-evolution of the dissipation rate of kinetic energy and the enstrophy are plotted for the different computations. We observe that the resolution (in terms of degrees of freedom) of the coarser $p = 4$ computation is not sufficient, whereas the $p = 4$ computation with higher resolution and the $p = 6$ provide similar results. Moreover, we note that these computations show a similar behavior of the dissipation rate of the kinetic energy and the enstrophy compared to the reference data. Thus, we conclude
that our implementation of the DG scheme is in principle capable of reproducing the turbulent behavior of this flow. Moreover, we note that with increasing polynomial order the number of cells in the grid needed for adequately resolving the turbulent features is decreasing. Thus, the total number of degrees of freedom can be reduced by increasing the polynomial order.

**Turbulent channel flow with \( \text{Re}_\tau = 180 \)**

Next, we investigate whether our implementation of the DG scheme is capable of dealing with wall-bounded turbulence. For that purpose, we consider a turbulent channel flow between isothermal walls. This is a well-known test case for wall-bounded turbulent flows. For a detailed discussion of turbulent channel flows we refer to [143] and the references therein. Frequently, incompressible Navier-Stokes equations are considered for the simulations of this flow, cf. [57,110,130]. Moser et al. [130] provide a detailed database of reference data from their computations. In the following we refer to these data as MKM.

In recent years a few groups started to apply DG methods for direct numerical simulations of compressible turbulent channel flows [39,58,177]. Here, we validate our implementation of the DG method with the simplest configuration for which Moser et al. [130] have provided reference data, namely a flow with a friction Reynolds number of about 180 and investigate if our implementation is capable of dealing with this flow, i.e., whether the flow becomes turbulent.

Following Chapelier et al. [39] we choose the computational domain

\[
\Omega = [-2\pi\delta,2\pi\delta] \times [-\delta,\delta] \times [-4/3\pi\delta,4/3\pi\delta],
\]

where we denote by \( \delta = 1 \) the half-channel height. The computational domain including the turbulent flow is shown in Figure 2.4(b). Physically the flow is driven by a pressure gradient in \( x \)-direction. Following [110] the flow can alternatively be driven by a body force \( \mathbf{f} \) instead of the pressure gradient resulting in a homogeneous pressure. For that purpose, we equip (2.45) with the source term \( s(\mathbf{u},\mathbf{x}) := (0, \rho \mathbf{f}, \rho \mathbf{f} \cdot \mathbf{v})^t \) on the right-hand side.

The flow is assumed to be periodic in streamwise (\( x_1 \)-) and spanwise (\( x_3 \)-) direction. We consider isothermal no-slip boundary conditions in \( x_2 \)-direction realized by \( \rho_{bc} = \rho \gamma / ((\gamma - 1) T_{\text{Wall}}) \), \( \mathbf{v}_{bc} = 0 \) and \( \rho E_{bc} = \rho_{bc} T_{\text{wall}} \) at the wall boundary \( x_2 = \pm 1 \). The boundary condition is imposed weakly by using these values to evaluate the flux integrals at the wall boundary in (2.29). Here, we consider a dimensionless temperature at the wall of \( T_{\text{wall}} = 1/(Ma^2 \gamma (\gamma - 1)) \), cf. [177], with a Mach number of 0.2. A reference Reynolds number of \( \text{Re} = 2770 \) is used.

The turbulent behavior of the flow in the channel is characterized by the friction Reynolds number

\[
\text{Re}_\tau := \frac{\sqrt{\text{Re} \ U_\tau \delta}}{\nu} \quad (2.50)
\]
2.6. NUMERICAL RESULTS

based on the dimensionless friction velocity

\[ u_r := \sqrt{\frac{\tau_w}{\rho}}. \]  \hspace{1cm} (2.51)

The dimensionless friction velocity depends on the dimensionless wall shear stress

\[ \tau_w := \pm \mu \partial v_1 / \partial x_2. \]

Based on these quantities, we define a normalized distance from the wall and normalize the velocity by the friction velocity, i.e.,

\[ y^+ = Re \tau x_2 \quad \text{and} \quad u^+ = \frac{v_1}{\sqrt{Re} \ u_r}. \]  \hspace{1cm} (2.52)

The external force \( \mathbf{f} \) driving the flow only is non-zero in \( x_1 \)-direction and chosen as \( f_1 = \tau_w / (\delta \rho_M) \), where \( \rho_M \) is the averaged density of the flow field.

Initially, we consider a laminar flow perturbed by fluctuations. To this end, the initial values are prescribed in terms of density \( \rho \), velocity \( \mathbf{v} \) and pressure \( p \) as

\[ \rho(0, \mathbf{x}) = 1, \]
\[ v_1(0, \mathbf{x}) = 1.5 \left( 1 + r(\mathbf{x}) \right) \left( 1 - x_2^2 \right), \]
\[ v_2(0, \mathbf{x}) = r(\mathbf{x}), \]
\[ v_3(0, \mathbf{x}) = r(\mathbf{x}), \]
\[ p(0, \mathbf{x}) = (\gamma - 1) T_{\text{wall}}, \]

where the random fluctuations \( r(\mathbf{x}) \in [-0.1, 0.1] \) are computed using a C++ random function. Starting with these initial data, the flow is remaining laminar for some time. Then the flow transitions eventually to a turbulent flow. Thus, the DNS of this configuration consists of two parts: at first the simulation is performed until the turbulence has been fully developed. Then, in the second part of the DNS the turbulent flow is analyzed.

For the DNS we consider a grid of \( 24 \times 16 \times 8 \) cells. In order to appropriately resolve the boundary layer near the isothermal walls, we condense the cells towards the walls using an arctangent stretching function in \( x_2 \)-direction, i.e.,

\[ \theta(\mathbf{x}) = (x_1, 2 \delta \phi(\frac{x_2 + \delta}{2 \delta}) - \delta, x_3)^t. \]  \hspace{1cm} (2.53)

The arctangent stretching function \( \phi : [0, 1] \rightarrow [0, 1] \) is defined by

\[ \phi(t) = \left\{ \begin{array}{ll}
  d \left[ 1.0 - \arctan(\frac{t - c}{c}) \right] / \arctan(\epsilon_0) & , \quad 0 \leq t \leq c \\
  d + (1 - d) \arctan(\frac{t - c}{1 - c}) / \arctan(\epsilon_1) & , \quad c < t \leq 1
\end{array} \right\}, \]
where

\[ c = \left( 1 + \frac{\epsilon_1}{\epsilon_0} \right)^{-1} \quad \text{and} \quad d = \left( 1 + \frac{\arctan(\epsilon_1)}{\arctan(\epsilon_0)} \right)^{-1}. \]

The parameters \( \epsilon_0, \epsilon_1 \in (0, \infty) \) control the stretching at the boundaries \( t = 0 \) and \( t = 1 \), respectively. For \( \epsilon_0, \epsilon_1 \to 0 \) the stretching tends to the identity resulting in a uniform Cartesian grid. The larger these parameters become, the more the grid spacings are condensed towards 0 and 1, respectively. In \( x_1 \)- and \( x_2 \)-direction the cells are uniformly distributed. Here, we consider \( \epsilon_0 = \epsilon_1 = 4.5 \). The resulting computational grid is shown in Figure 2.4(a).

We apply a fifth-order scheme in space, i.e., \( p = 5 \), using the tensor product polynomial space. Thus, we have 125 degrees of freedom in each cell. In order to resolve all relevant features in the turbulent flow, the time step size has to be chosen smaller than the Kolmogorov time-scale, cf. [143]. Thus, \( \Delta t \) is chosen as \( 1.25 \times 10^{-5} \). Since we are dealing with a diffusive problem the time-step size is chosen as \( \Delta t = \mathcal{O}(\Delta x^2) \). Thus, it is not necessary to have the same order in the time-discretization as in the polynomial space. Moreover, we are interested here in reproducing turbulence and not in a local accuracy of the solution. Besides, the computational complexity has to be reduced as much as possible in order to be able to perform the DNS in a feasible computational time. For these reasons, low-order time-discretizations are frequently used for DNS, cf. [39,177]. Here, we consider an explicit second-order SSP-RK scheme. In order to reduce the computational time, the computations have been performed in parallel using a shared memory parallelization with MPI with 384 work units. The computations have been performed on an Intel Xeon cluster with an InfiniBand network. The overall computation has run for around 160 hours. This corresponds to 61,440 core hours.

![computational grid and turbulent flow](image)

(a) computational grid

(b) turbulent flow

Figure 2.4: Turbulent channel flow: computational grid and snapshot of the streamwise velocity in the turbulent flow. The isothermal walls are shown in gray.

At first we have run the simulation up to \( t = 100 \) by performing 8,000,000 time steps. The flow eventually becomes turbulent at \( t \approx 90 \). Then, the computation is continued up to \( T = 220 \) performing additional 9,600,000 time steps to analyze the turbulent behavior
2.6. NUMERICAL RESULTS

of the flow. Here, we focus on the average of the normalized velocity $u^+$ computed in this second part of the DNS. The averaging is performed in streamwise ($x_1$-) and spanwise ($x_3$-) direction as well as in time. To analyze the quality of the turbulent flow in our DNS, we compare the resulting profile with the reference data from Moser et al. [130] and the well-known ideal wall law by von Kármán [171]

$$u^+(y^+) = \begin{cases} 
  y^+ & \text{if } y^+ < 5, \\
  5 \log(y^+) - 3.05 & \text{if } 5 \leq y^+ \leq 30, \\
  5/2 \log(y^+) + 5.5 & \text{else}
\end{cases} \quad (\text{viscous sublayer}),$$

(2.54)

For that purpose, these profiles are plotted over $y^+$ in Figure 2.5. The averaged velocity profile of a turbulent channel flow near the boundary consists of three layers, namely the \textit{viscous sublayer} for $y^+ < 5$, the \textit{buffer layer} for $5 \leq y^+ \leq 30$ and the \textit{log layer} for $y^+ > 30$, cf. [143]. When looking at Figure 2.5, we first note that the flow clearly becomes turbulent. Moreover, we find that our results are almost identical with the results of the DNS from Moser et al. [130]. Our results as well as the results from Moser et al. agree very well with the ideal wall law by von Kármán in all three layers.

![Figure 2.5: Turbulent channel flow: averaged streamwise velocity profile in wall normal direction.](image)

Chapter 3

Multiresolution analysis

When dealing with functions \( u \in L^2(\Omega) \) on a bounded domain, we are interested in distinguishing between information from different scales and analyzing the local behavior. For this purpose we perform a multiresolution analysis [14, 44, 64, 96, 119, 126, 174] for \( L^2(\Omega) \).

The concept of multiresolution analysis (MRA) is based on a sequence of nested spaces

\[
S_0 \subset S_1 \subset \ldots \subset S_\ell \subset S_{\ell+1} \subset \ldots \subset L^2(\Omega)
\]  

(3.1)

that is dense in \( L^2(\Omega) \), i.e.,

\[
\bigcup_{\ell=0}^{\infty} S_\ell = L^2(\Omega).
\]  

(3.2)

Since we are interested in applying the MRA in the context of DG schemes, we consider spaces \( S_\ell, \ell \in \mathbb{N}_0 \), composed of piecewise polynomials defined on a hierarchy of nested grids. The concept of MRA is usually performed on a uniform dyadic hierarchy \([64, 119]\). We weaken this restriction and consider non-uniform hierarchies, where the subdivision is not necessarily uniform. The degrees of freedom of \( S_\ell, \ell \in \mathbb{N}_0 \), are locally independent since continuity of the piecewise polynomials at the cell interfaces is not enforced.

The basic idea is to focus on orthogonal complements between successive refinement levels. Consequently, \( u \) can be represented by an (infinite) sum of contributions corresponding to different orthogonal complement spaces on successive refinement levels and a contribution from the coarsest space \( S_0 \). We call this the \textit{multi-scale} decomposition of \( u \). This decomposition reveals insight into the local behavior of \( u \): it can be proven, that the contributions of the orthogonal complements become small if the underlying function is locally smooth.

Frequently, the MRA is described in terms of basis expansions for the complement spaces \([44, 119, 133, 156, 167]\). In general, it is often difficult to find compactly supported orthogonal bases for the complements. Thus, usually \textit{bi-orthogonal} Riesz bases are considered in order to realize a stable MRA \([36, 60, 133]\). In our setting, the locality of the
spaces enables the use of orthonormal bases with compact support [2, 83]. Usually, basis functions for the orthogonal complement spaces, called multiwavelets, are explicitly constructed and used for the MRA, cf. [2, 64, 119]. However, the construction and the use of multiwavelets in the general, non-uniform case is computationally expensive. Therefore, we formulate the MRA in terms of projections in the associated spaces instead of coefficients corresponding to the basis expansion of the complement spaces. Due to the locality of the spaces of piecewise polynomials, several results can be transferred directly to the functions without an explicit use of the multiwavelets. This allows for an alternative realization of the MRA avoiding the explicit construction and use of multiwavelets, respectively.

Finally, we present the algorithms needed for the implementation of the MRA and analyze the computational cost and memory requirements of these algorithms. Thereby, we compare the alternative approach which avoids the use of multiwavelets with the standard approach using multiwavelets.

### 3.1 Nested grid hierarchies

In order to generate a sequence of nested spaces as in (3.1) we start with a hierarchy of nested grids: on each refinement level \( \ell \in \mathbb{N}_0 \) of the hierarchy the bounded computational domain \( \Omega \) is partitioned by a finite number of cells and the corresponding grid is characterized by the index set \( \mathcal{I}_\ell \), i.e.,

\[
\overline{\Omega} = \bigcup_{\lambda \in \mathcal{I}_\ell} V_\lambda \quad \forall \ \ell \in \mathbb{N}_0.
\]  

(3.3)

Here, \( \lambda \) is a cell identifier containing the level as well as the number of the cell on the specific refinement level. Due to the nestedness there exists a refinement set \( \mathcal{M}_\lambda \in \mathcal{I}_{\ell+1} \) for each cell \( V_\lambda \) on level \( \ell \) such that

\[
V_\lambda = \bigcup_{\mu \in \mathcal{M}_\lambda} V_\mu \quad \forall \ \lambda \in \mathcal{I}_\ell, \ \ell \in \mathbb{N}_0.
\]  

(3.4)

On each refinement level \( \ell \in \mathbb{N}_0 \) of the hierarchy we introduce the finite-dimensional discretization space \( S_\ell \) of piecewise polynomials

\[
S_\ell := \{ f \in L^2(\Omega) : f|_{V_\lambda} \in \Pi_{p-1}(V_\lambda) \ \forall \ \lambda \in \mathcal{I}_\ell \}, \ \ell \in \mathbb{N}_0.
\]  

(3.5)

Here, the local polynomial space \( \Pi_{p-1}(V_\lambda) \) defined for the DG scheme in (2.4) is used. In order to ensure that the multiresolution sequence (3.1) is dense in \( L^2(\Omega) \), i.e., (3.2) holds, we require that with increasing refinement level the diameters of the cells tend to zero, i.e.,

\[
\lim_{\ell \to \infty} \max_{\lambda \in \mathcal{I}_\ell} \text{diam}(V_\lambda) = 0.
\]  

(3.6)
This is only a weak restriction in the choice of the hierarchy. The herein considered setting allows arbitrary shapes of elements, e.g. quadrilateral, triangular or curved elements. The computational domain can be assembled from a finite number of blocks of individual hierarchies. Each of these hierarchies may consist of elements with different shapes. In the following, a few examples of nested grid hierarchies are presented.

3.1.1 Examples of nested hierarchies

Uniform dyadic hierarchies

The concept of multiresolution analysis has its origin in classical wavelet theory, cf. [64, 119]. There dyadic hierarchies containing squared elements with uniform subdivisions are commonly considered, i.e.,

$$|\mathcal{M}_\lambda| = 2^d$$

and

$$|V_\mu| = 2^{-d}|V_\lambda|, \quad \mu \in \mathcal{M}_\lambda.$$  

In the following we refer to this hierarchy as the uniform dyadic hierarchy because all cells on a refinement level are uniform. An example of a uniform dyadic hierarchy with quadrilaterals is shown in Figure 3.1(a).

Non-uniform hierarchies via grid mappings

Non-uniform hierarchies can be generated by means of a continuous bijective grid function $\theta : \Omega_{\text{ref}} \to \Omega$ applied to the cells of a uniform grid hierarchy. For a given reference hierarchy on a reference domain, i.e.,

$$\overline{\Omega_{\text{ref}}} = \bigcup_{\lambda \in \mathcal{I}_\ell} V_{\lambda}^{\text{ref}}, \quad \forall \; \ell \in \mathbb{N}_0,$$

a nested non-uniform hierarchy is obtained by

$$V_{\lambda}^{\theta} := \theta(V_{\lambda}^{\text{ref}}), \quad \lambda \in \mathcal{I}_\ell, \; \ell \in \mathbb{N}_0.$$  

(3.7)

An important example for the use of grid mappings is the simulation of a viscous flow over a solid wall: due to the no-slip condition at the wall, a thin boundary layer develops for a large Reynolds number flow. In order to obtain an adequate numerical approximation, the boundary layer has to be appropriately resolved. For that purpose, normal to a solid wall a very fine resolution is needed, whereas in stream-wise direction moderate grid spacing is sufficient. Starting with a dyadic uniform reference hierarchy, this can be realized by a function $\theta$, which condenses the grid spacings normal to the wall.
As an example, we consider

\[
\theta(x) = \begin{pmatrix}
1 - \frac{\arctan(c - c x_1)}{\arctan(c)} \\
\frac{x_2}{x_1} \\
\vdots \\
\frac{x_d}{x_1}
\end{pmatrix},
\]

with \( c > 0 \). Then \( \Omega^{\text{ref}} = [0, 1]^d \) is mapped onto \( \Omega^{\text{ref}} \) itself, but the grid resulting from (3.7) is stretched towards \( x_1 = 0 \). The resulting hierarchy is shown in Figure 3.1(b). Furthermore, grid functions can be used to generate curved grids, e.g. grids for flows around airfoils, etc. For a detailed discussion on the choice of grid functions we refer to [125].

![Diagram of grid hierarchies](image)

**Figure 3.1**: Examples of grid hierarchies for \( d = 2 \).

**Quasi-uniform grid hierarchies**

Often *quasi-uniform* hierarchies are considered as foundation for the MRA, see for instance [133]. A hierarchy is called quasi-uniform if there exist constants \( C_{H,m}, C_{H,M} > 0 \) such that

\[
C_{H,m} 2^{-\ell} \leq \text{diam}(V_\lambda) \leq C_{H,M} 2^{-\ell} \quad \forall \lambda \in \mathcal{I}_\ell, \ell \in \mathbb{N}_0.
\]

Non-uniform hierarchies obtained by grid mappings are quasi-uniform under some regularity assumptions on \( \theta \).
3.2. MULTI-SCALE DECOMPOSITION

Proposition 3.1 Let the reference hierarchy be quasi-uniform with constants $C_{H,m}^{\text{ref}}$ and $C_{H,M}^{\text{ref}}$, i.e.,

$$C_{H,m}^{\text{ref}} 2^{-\ell} \leq \text{diam}(V^\ell_{\lambda}) \leq C_{H,M}^{\text{ref}} 2^{-\ell} \quad \forall \lambda \in \mathcal{I}_\ell, \quad \ell \in \mathbb{N}_0.$$  

Furthermore, let $\theta$ be invertible and $\theta$ and $\theta^{-1}$ be Lipschitz continuous with uniform Lipschitz constants $L_\theta$ and $L_{\theta^{-1}}$, respectively. Then, the resulting hierarchy is quasi-uniform with

$$\frac{C_{H,m}^{\text{ref}}}{L_{\theta^{-1}}} 2^{-\ell} \leq \text{diam}(V^\ell_{\lambda}) \leq C_{H,M}^{\text{ref}} L_\theta 2^{-\ell} \quad \forall \lambda \in \mathcal{I}_\ell, \quad \ell \in \mathbb{N}_0.$$  

Proof: We first note that

$$\text{diam}(V^\ell_{\lambda}) \leq L_\theta \text{diam}(V^\ell_{\lambda}^{\text{ref}}) \leq C_{H,M}^{\text{ref}} L_\theta 2^{\ell}, \quad \forall \lambda \in \mathcal{I}_\ell, \quad \ell \in \mathbb{N}_0.$$  

Analogously, we conclude from $V^\ell_{\lambda}^{\text{ref}} := \theta^{-1}(V^\ell_{\lambda})$ that

$$\text{diam}(V^\ell_{\lambda}^{\text{ref}}) \leq L_{\theta^{-1}} \text{diam}(V^\ell_{\lambda}) \quad \forall \lambda \in \mathcal{I}_\ell, \quad \ell \in \mathbb{N}_0.$$

For the example (3.8) considered in the previous section, we have

$$L_\theta = \sup_{x \in \Omega} \|D\theta(x)\|_2 = \frac{c}{\arctan(c)} \quad \text{and} \quad L_{\theta^{-1}} = \sup_{x \in \Omega} \|D\theta^{-1}(x)\|_2 = 1 + c^2 \arctan(c),$$

where by $D\theta$ and $D\theta^{-1}$ we denote the Jacobian matrices of $\theta$ and $\theta^{-1}$, respectively. For $c \to 0$ the stretching converges to the identity. For the simulation of flows with high Reynolds numbers where the boundary layers are very thin, $c$ has to be chosen very large. However, the larger $c$ becomes, the more cells are condensed towards $x_1 = 0$, thereby, $L_\theta$ and $L_{\theta^{-1}}$ are growing and thus, the worse the constants in (3.9) get. For this reason, the concept of quasi-uniform hierarchies is not well-suited for the simulation of such flows. Therefore we abstain from claiming that the hierarchy of nested grids is quasi-uniform in this work.

3.2 Multi-scale decomposition

In order to investigate the difference between two successive refinement levels we consider the orthogonal complement space $W_\ell$ of $S_\ell$ with respect to $S_{\ell+1}$ defined by

$$W_\ell := \{ d \in S_{\ell+1} : \langle d, u \rangle_\Omega = 0 \forall u \in S_\ell \},$$

where by $\langle \cdot, \cdot \rangle_\Omega$ the $L^2$-inner product on $\Omega$ is denoted. Since $S_{\ell+1}$ is finite dimensional, we thus decompose $S_{\ell+1}$ into the direct sum of $S_\ell$ and its orthogonal complement $W_\ell$, i.e.,

$$S_{\ell+1} = S_\ell \oplus W_\ell, \quad \ell \in \mathbb{N}_0.$$
Then, the space \( S_L, \, L \in \mathbb{N}, \) can be decomposed into the coarse discretization space \( S_0 \) and complement spaces \( W_\ell, \, 0 \leq \ell \leq L - 1, \) by applying (3.11) recursively:
\[
S_L = S_0 \oplus W_0 \oplus \ldots \oplus W_{L-1}. \tag{3.12}
\]
Since the multiresolution sequence is dense in \( L^2(\Omega), \) we conclude
\[
L^2(\Omega) = S_0 \oplus \bigoplus_{\ell \in \mathbb{N}} W_\ell^{14_{\ell^2}}. \tag{3.13}
\]
Thus, each function \( u \in L^2(\Omega) \) can be represented by an (infinite) multi-scale decomposition consisting of contributions from different refinement levels. For that purpose, we consider orthogonal projections \( P_{S_\ell} : L^2(\Omega) \to S_\ell \) and \( P_{W_\ell} : L^2(\Omega) \to W_\ell \) and define
\[
u^{\ell} := P_{S_\ell}(u), \quad d^{\ell} := P_{W_\ell}(u), \quad \ell \in \mathbb{N}_0. \tag{3.14}\]
In the following we use the convention that by \( u^{\ell} \) and \( d^{\ell} \) we denote the projection of \( u \in L^2(\Omega) \) to the subspaces \( S_\ell \) and \( W_\ell, \) respectively.

For convenience of the reader, we shortly summarize the definition and some basic properties of an orthogonal projection \( P_V : L^2(\Omega) \to V \) from \( L^2(\Omega) \) to the closed linear subspace \( V \subset L^2(\Omega). \) For \( u \in L^2(\Omega) \) its orthogonal projection \( P_V(u) \in V \) is defined by
\[
\{ P_V(u) - u, v \}_{\Omega} = 0 \quad \forall v \in V. \tag{3.15}\]
The orthogonal projection is a linear operator and provides the best approximation of \( u \in L^2(\Omega) \) in \( V, \) i.e.,
\[
\min_{v \in V} \| v - u \|_{L^2(\Omega)} = \| P_V(u) - u \|_{L^2(\Omega)}. \tag{3.16}\]
If \( V \) is spanned by an orthonormal basis, i.e., \( V = \operatorname{span} v_i, \) the orthogonal projection can be represented in this basis as
\[
P_V(u) = \sum_{i \in V} \langle u, v_i \rangle_{\Omega} v_i. \tag{3.17}\]
Next, we consider a closed linear subspace \( V_2 \subset V. \) Then for the projection on \( V_2 \) there holds
\[
\{ P_{V_2}(P_V(u)) - u, v \}_{\Omega} = \{ P_{V_2}(P_V(u)) - P_V(u), v \}_{\Omega} + \{ P_V(u) - u, v \}_{\Omega} = 0
\]
for all \( v \in V_2. \) Thus, the projection of \( u \) onto \( V_2 \) can be computed from the projection onto \( V \) by
\[
P_{V_2}(u) = P_{V_2}(P_V(u)). \tag{3.18}\]
3.2. MULTI-SCALE DECOMPOSITION

Due to the nestedness of the spaces (3.11), we conclude by (3.18) that contributions of two successive refinement levels $\ell$ and $\ell + 1$, $\ell \in \mathbb{N}_0$ are related to each other by

$$u^\ell \equiv P_{S_\ell}(u) = P_{S^\ell}(u^{\ell+1}), \quad d^\ell \equiv P_{W_\ell}(u) = P_{W^\ell}(u^{\ell+1}), \quad \ell \in \mathbb{N}_0.$$  \hspace{1cm} (3.19)

Consequently, $u^{\ell+1}$ may be written as

$$u^{\ell+1} = u^\ell + d^\ell, \quad \ell \in \mathbb{N}_0.$$  \hspace{1cm} (3.20)

By applying (3.20) recursively we can express $u^L \equiv P_{S_0}(u)$ for $L \in \mathbb{N}_0$ in terms of the contributions of the orthogonal complement spaces and the coarsest discretization space $S_0$, i.e.,

$$u^L = u^0 + \sum_{\ell=0}^{L-1} d^\ell.$$  \hspace{1cm} (3.21)

Due to (3.16) and (3.2), we conclude that $u^L$ converges to $u \in L^2(\Omega)$, i.e.,

$$\|u - u^L\|_{L^2(\Omega)} \to 0, \quad \text{for } L \to \infty.$$  \hspace{1cm} (3.22)

Thus, we may decompose $u \in L^2(\Omega)$ by means of the limit process in (3.22) as

$$u = u^0 + \sum_{\ell \in \mathbb{N}_0} d^\ell.$$  \hspace{1cm} (3.23)

We call the right-hand side of (3.23) the multi-scale decomposition of $u$. In the following proposition we summarize basic properties of the orthogonal complements $W_\ell$, $\ell \in \mathbb{N}_0$, and the resulting multi-scale decomposition.

**Proposition 3.2** Let $u \in L^2(\Omega)$ and $\ell, \hat{\ell} \in \mathbb{N}_0$. Then, the following statements hold true:

(i) For $d \in W_\ell$ there holds

$$\langle d, w \rangle_\Omega = 0 \quad \forall \ w \in S_\ell.$$  \hspace{1cm} (3.24)

(ii) For $d \in W_\ell$ there holds

$$\langle d, \hat{d} \rangle_\Omega = 0 \quad \forall \ \hat{d} \in W_{\hat{\ell}}, \ \hat{\ell} \neq \ell.$$  \hspace{1cm} (3.25)

(iii) For $u^\ell \equiv P_{S_\ell}(u)$ there holds

$$\langle u, w \rangle_\Omega = \langle u^\ell, w \rangle_\Omega \quad \forall \ w \in S_\ell.$$  \hspace{1cm} (3.26)
Proof: (i) follows from the definition of the complement spaces $W_\ell$ in (3.10). (ii) is a consequence of (3.11). In order to prove (iii) we first approximate $u$ by $u^L$ and then, due to (3.21) there holds for $L \geq \ell$ that

\[
\langle u^\ell, w \rangle_\Omega = \langle u^\ell, w \rangle_\Omega + \sum_{\ell=\ell}^{L-1} \langle d^\ell, w \rangle_\Omega, \quad \forall \ w \in S_\ell.
\]

Furthermore, the nestedness (3.1) implies that $w \in S_{\ell}$ for $\ell \geq \ell$. Hence, we deduce from (i) that

\[
\sum_{\ell=\ell}^{L-1} \langle d^\ell, w \rangle_\Omega = 0, \quad \forall \ w \in S_\ell.
\]

Then, we conclude by using the Cauchy-Schwarz inequality in combination with (3.22) that

\[
\langle u^\ell, w \rangle_\Omega = \langle u^L, w \rangle_\Omega \rightarrow \langle u, w \rangle_\Omega \quad \text{for } L \rightarrow \infty,
\]

and, thus, (iii) follows immediately. □

The above Proposition implies the following Conclusion.

Conclusion 3.1 The average in the multi-scale decomposition is conserved in $u^0$, i.e.,

\[
\int_\Omega u(x)dx = \int_\Omega u^0(x)dx, \quad \int_\Omega d^\ell(x)dx = 0 \quad \forall \ \ell \in \mathbb{N}_0.
\]

Proof: Note that $1 \in S_\ell$ for all $\ell \in \mathbb{N}_0$. Thus, (3.27) follows from Proposition 3.2(i) and (iii) with $w = 1$. □

3.3 Separation of local contributions

In (3.23) we separate contributions corresponding to the different refinement levels. The spaces of piecewise polynomials $S_\ell$ as well as the orthogonal complement spaces $W_\ell$ have a local structure. The degrees of freedom are locally independent, since discontinuities at cell interfaces are admissible. This allows for a local computation of the orthogonal projections in (3.19). Furthermore, the local structure can be used to estimate the approximation error locally and conservation properties of the orthogonal projections (3.14). For that purpose, we spatially separate the local contributions in the multi-scale decomposition (3.23), i.e., for $\ell \in \mathbb{N}_0$ we split $u^\ell$ and $d^\ell$ into compactly supported local contributions defined by

\[
u^\ell_\lambda := u^\ell 1_{V_\lambda}, \quad d^\ell_\lambda := d^\ell 1_{V_\lambda},
\]
3.3. SEPARATION OF LOCAL CONTRIBUTIONS

where by $1_{V_{\lambda}}$ we denote the characteristic function of $V_{\lambda}$, i.e.,

$$1_{V_{\lambda}}(x) := \begin{cases} 
1 & \text{if } x \in V_{\lambda}, \\
0 & \text{else},
\end{cases}$$

Consequently, we may decompose $u^\ell$ and $d^\ell$ into the sum of their local contributions, i.e.,

$$u^\ell = \sum_{\lambda \in I_\ell} u^\ell_{\lambda}, \quad d^\ell = \sum_{\lambda \in I_\ell} d^\ell_{\lambda}. \quad (3.29)$$

Based on (3.23), we formulate a localized multi-scale decomposition

$$u = \sum_{\lambda \in I_0} u^0_{\lambda} + \sum_{\ell \in N_0} \sum_{\lambda \in I_\ell} d^\ell_{\lambda}. \quad (3.30)$$

In order to derive relations between local contributions on successive refinement levels, we define

$$u^\ell+1_{\lambda} := \sum_{\mu \in M_{\lambda}} u^\ell_{\mu}, \quad \lambda \in I_\ell, \ell \in N_0. \quad (3.31)$$

In the following proposition basic properties of the local contributions (3.28) and (3.31) are summarized. The bottom line of these properties is that localization and projection to the subspaces are commutative as depicted in Figure 3.2.

**Theorem 3.1** For the local contributions (3.28) of the localized multi-scale decomposition (3.30) the following statements hold true for $\lambda \in I_\ell, \ell \in N_0$:

(i) $u^\ell_{\lambda} \in S_{\ell}$ and $d^\ell_{\lambda} \in W_{\ell}$,

(ii) $u^\ell_{\lambda} = P_{S_{\ell}}(u 1_{V_{\lambda}})$,

(iii) $u^\ell_{\lambda} = P_{S_{\ell}}(u^{\ell+1}_{\lambda})$ and $d^\ell_{\lambda} = P_{W_{\ell}}(u^{\ell+1}_{\lambda})$,

(iv) $u^{\ell+1}_{\mu} = u^\ell_{\lambda} 1_{V_{\lambda}} + d^\ell_{\lambda} 1_{V_{\lambda}}, \quad \mu \in M_{\lambda}$.

Proof: First of all, we note that due to the definition (3.5) of $S_{\ell}$ there holds $w_{\lambda} := w 1_{V_{\lambda}} \in S_{\ell}$ for all $w \in S_{\ell}$ and $\lambda \in I_{\ell}$. Thus, we conclude $u^\ell_{\lambda} \in S_{\ell}$. Furthermore, due to (3.10) there holds

$$\langle d^\ell_{\lambda}, w \rangle_\Omega = \langle d^\ell_{\lambda}, w \rangle_{V_{\lambda}} = \langle d^\ell_{\lambda}, w_{\lambda} \rangle_\Omega = 0 \quad \forall \ w \in S_{\ell}.$$ 

Therefore, we deduce from (3.10) that $d^\ell_{\lambda} \in W_{\ell}$. Next, we note by (i), (3.14) and (3.15) that

$$\langle u^\ell_{\lambda} - u 1_{V_{\lambda}}, v \rangle_\Omega = \langle P_{S_{\ell}}(u) 1_{V_{\lambda}} - u 1_{V_{\lambda}}, v \rangle_\Omega = \langle P_{S_{\ell}}(u) - u, v 1_{V_{\lambda}} \rangle_\Omega = 0$$

for all $v \in S_{\ell}$. This completes the proof.
holds for arbitrary \( v \in S_\ell \). Thus, \( u^{\lambda}_{\tilde{\lambda}} \) coincides with the projection \( P_{S_\ell}(u 1_{V_\lambda}) \in S_\ell \) of \( u 1_{V_\lambda} \in L^2(\Omega) \), i.e., (ii) holds.

In order to prove (iii), we first consider \( w := P_{S_\ell}(u^{\lambda+1}) \in S_\ell \). As we have discussed above, we conclude that \( w^{\lambda} := w 1_{V_\lambda} \in S_\ell \) for all \( \tilde{\lambda} \in \mathcal{I}_\ell \). From the definition of the orthogonal projection, we then conclude for all \( \tilde{\lambda} \neq \lambda \) that

\[
0 = \langle w - u^{\lambda+1}, w^{\lambda} \rangle_{V_\lambda} = \langle w^{\lambda}, w^{\lambda} \rangle_{V_\lambda}.
\]

Hence, we deduce \( w^{\lambda} = 0 \) for all \( \tilde{\lambda} \neq \lambda, \tilde{\lambda} \in \mathcal{I}_\ell \). Thus, by definition of \( w \), we conclude that \( \text{supp } w = \text{supp } \left( P_{S_\ell}(u^{\lambda+1}) \right) \subset V_\lambda \). Due to the linearity of the orthogonal projection we deduce by (3.29) and (3.31) that

\[
u^{\lambda}_\ell = P_{S_\ell}(u^{\lambda+1}) = \sum_{\tilde{\lambda} \in \mathcal{I}_\ell} P_{S_\ell}(u^{\lambda+1})_{\tilde{\lambda}}.
\]

Combining these observations leads to the first equation. The second equation can be proven analogously: by using (i), we conclude \( \left( P_{W_\ell}(u^{\lambda+1}) \right) 1_{V_\lambda} \in W_\ell \) and proceed in the same way as before.

Finally, we conclude (iv) by multiplying (3.20) with \( 1_{V_\lambda} \) and \( 1_{V_{\mu}} \) and using the definition of the local contributions (3.28).

\[\begin{array}{c|c|c}
\begin{array}{c}
u^{\lambda+1} \\downarrow P_{S_\ell} \\downarrow \nu^{\lambda+1} \\
\begin{array}{c}
\downarrow P_{S_\ell} \\
u^{\lambda} \\downarrow P_{W_\ell} \\downarrow d^{\lambda+1} \\
\downarrow P_{W_\ell}
\end{array}
\end{array}
\end{array}\]

Figure 3.2: Commutation of projection and localization in the two-scale transformation.

The third statement in Theorem 3.1 allows a local computation of the multi-scale decomposition: the projections to \( W_\ell \) and \( S_\ell \) can be computed level-wise proceeding from fine to coarse recursively. The decomposition of \( u^{\lambda+1} \) into \( u^\ell \) and \( d^\ell \) can be computed cell-wise by local projections. In Figure 3.3 the local decomposition according to Theorem 3.1(iii) is shown for a uniform dyadic hierarchy. Moreover, we note that for the localized multi-scale decomposition (3.30) there holds

\[
\|u\|_{L^2(\Omega)} = \sum_{\lambda \in \mathcal{I}_0} \|u^{\lambda}_{\lambda}\|_{L^2(V_\lambda)}^2 + \sum_{\ell \in \mathcal{N}_0} \sum_{\lambda \in \mathcal{I}_\ell} \|d^{\lambda}_{\lambda}\|_{L^2(V_\lambda)}^2.
\]

(3.32)

As a consequence of the separation of local contributions (3.28) we can derive a local equivalent of Conclusion 3.1:
3.3. SEPARATION OF LOCAL CONTRIBUTIONS

Figure 3.3: Example of local multi-scale decomposition.

**Conclusion 3.2 (Local conservation)** For \( u \in L^2(\Omega) \), \( u^\ell \equiv P_{S_\ell}(u) \), \( d^\ell \equiv P_{W_\ell}(u) \), \( \ell \in \mathbb{N}_0 \) there holds

\[
\int_{V_\lambda} u(x) dx = \int_{V_\lambda} u^\ell(x) dx, \quad \int_{V_\lambda} d^\ell(x) dx = 0, \quad \forall \; \lambda \in \mathcal{I}_\ell. \tag{3.33}
\]

**Proof:** Note that \( 1_{V_\lambda} \in S_\ell \) for all \( \lambda \in \mathcal{I}_\ell, \ell \in \mathbb{N}_0 \). Thus, (3.33) follows from Proposition 3.2 (i) and (iii) with \( w = 1_{V_\lambda} \).

Thus, we note that the multi-scale decomposition is locally conservative.

Furthermore, we can use the separation of local contributions (3.29) to estimate the local approximation error of the orthogonal projections. The estimate is based on the local regularity of the underlying function \( u \in L^2(\Omega) \): if \( u \) is locally smooth, then the local approximation error of \( u^\ell \) is decreasing with order \( p \).

**Property 2 (Local approximation error)** Let \( \lambda \in \mathcal{I}_\ell \), \( \ell \in \mathbb{N}_0 \), \( V_\lambda \) be convex and \( u \in L^2(\Omega) \) be locally smooth, i.e., \( u|_{V_\lambda} \in H^p(V_\lambda) \). Then, the error of \( u^\ell \) is locally bounded by

\[
\| u - u^\ell_{\lambda} \|_{L^2(V_\lambda)} \leq \text{diam}(V_\lambda)^p \sum_{|\alpha|_{1^p} = p} \frac{1}{\alpha!} \| \partial^\alpha u \|_{L^2(V_\lambda)}. \tag{3.34}
\]

This result is a familiar consequence of Whitney’s theorem. For details we refer to literature on classical approximation theory, see for instance [67].
3.4 Cancellation property

In this section we derive an estimate for the $L^2$-norm of the local contributions $d^\ell_\lambda \in W_\ell$, \( \lambda \in \mathcal{I}_\ell \) and \( \ell \in \mathbb{N}_0 \). This estimate is based on the local regularity of the underlying function \( u \in L^2(\Omega) \) and results from the estimate of the local approximation error in Property 2.

**Proposition 3.3** Let \( \lambda \in \mathcal{I}_\ell, \ \ell \in \mathbb{N}_0 \). If \( V_\lambda \) is convex and \( u|_{V_\lambda} \in H^p(V_\lambda) \), then it holds

\[
\|d^\ell_\lambda\|_{L^2(V_\lambda)} \leq \text{diam}(V_\lambda)^p \sum_{|\alpha|=p} \frac{1}{\alpha!} \|D^\alpha u\|_{L^2(V_\lambda)}.
\] (3.35)

**Proof:** Since \( d^\ell_\lambda \in W_\ell \) we conclude from (3.14) and (3.15) that

\[
\langle d^\ell_\lambda - u, d^\ell_\lambda \rangle_{V_\lambda} = \langle d^\ell - u, d^\ell_\lambda \rangle_{\Omega} = 0.
\]

Together with (3.28) and (3.24) we obtain

\[
\|d^\ell_\lambda\|_{L^2(V_\lambda)} = \langle u, d^\ell_\lambda \rangle_{V_\lambda} = \langle u - u^\ell, d^\ell \rangle_{V_\lambda}.
\]

By applying the Cauchy-Schwarz inequality we finally conclude (3.35) using Property 2. \( \square \)

The local smoothness implies that the local contributions of \( d^\ell \) decrease with order \( p \) for increasing \( \ell \). In previous works, e.g. [83, 108], it was shown that the coefficients of a bi-orthogonal basis expansion of \( d^\ell_\lambda \) are becoming small if the underlying function is smooth. Here, we derive an estimate for \( \|d^\ell_\lambda\|_{L^2(V_\lambda)} \) independent of the choice of the bases for the complement spaces \( W_\ell \). Thus, Proposition 3.3 is a generalization of the results in [83, 108].

3.5 Significance, thresholding and grid adaptation

A function \( u \in L^2(\Omega) \) might have a very heterogeneous structure. It may exhibit local discontinuities and elsewhere it is smooth. In Section 3.4 we have observed that \( d^\ell_\lambda \) is locally decaying fast for \( \ell \to \infty \) if the underlying function is locally smooth. Thus, in these regions contributions from the orthogonal complements are comparably small, whereas in non-smooth regions the contributions are comparably large.

By neglecting small contributions from the orthogonal complement spaces, we may locally sparsify the projection \( u^L \) corresponding to an arbitrary but fixed refinement level \( L \in \mathbb{N}_0 \) and define an approximation \( u^{L,\varepsilon} \). For that purpose, we have to find a suitable norm to measure the local significance of \( d^\ell_\lambda \) in order to decide if the contribution can be neglected or has to be maintained. For this purpose we define for \( \lambda \in \mathcal{I}_\ell, \ \ell \in \mathbb{N}_0 \), a local complement space

\[
W_{\ell,\lambda} := \{ d|_{V_\lambda} : d \in W_\ell \} \in L^2(V_\lambda).
\] (3.36)
3.5. SIGNIFICANCE, THRESHOLDING AND GRID ADAPTATION

Next, we introduce for each of the local complement spaces a local norm \( \| \cdot \|_{\lambda} : W_{\ell,\lambda} \to \mathbb{R} \), \( \lambda \in \mathcal{I}_\ell \) and \( \ell \in \mathbb{N}_0 \). In order to be flexible in its choice and to cover the results from previous works [83, 108], we solely assume that it is uniformly equivalent to \( \| \cdot \|_{L^2(V_\lambda)} / \sqrt{\| V_\lambda \|} \), i.e., there exist constants \( C_{\text{loc},m}, C_{\text{loc},M} > 0 \) that depend neither on the level \( \ell \in \mathbb{N}_0 \) nor on the local position \( V_\lambda \) such that

\[
C_{\text{loc},m} \| d_{\lambda}^\ell \|_\lambda \leq \frac{\| d_{\lambda}^\ell \|_{L^2(V_\lambda)}}{\sqrt{\| V_\lambda \|}} \leq C_{\text{loc},M} \| d_{\lambda}^\ell \|_\lambda \quad \forall \lambda \in \mathcal{I}_\ell, \ell \in \mathbb{N}_0.
\] (3.37)

By introducing local threshold values \( \varepsilon_{\lambda,L} \geq 0 \) we distinguish between significant and non-significant contributions. These threshold values are chosen level dependent and might differ from cell to cell on the same level. Moreover, they depend on the maximum level \( L \). In order to retain flexibility, we require only a weak constraint on the choice of the local thresholds. More precisely, we claim that there exists an \( \varepsilon_{\max} > 0 \) such that for all \( L \in \mathbb{N} \) there holds

\[
\sum_{\ell=0}^{L-1} \max_{\lambda \in \mathcal{I}_\ell} \varepsilon_{\lambda,L} \leq \varepsilon_{\max}.
\] (3.38)

**Definition 3.1** For \( \lambda \in \mathcal{I}_\ell \), we call a local contribution on level \( \ell \) \( \| \cdot \|_\lambda \)-significant iff

\[
\| d_{\lambda}^\ell \|_\lambda > \varepsilon_{\lambda,L}.
\] (3.39)

For a uniform dyadic hierarchy a straightforward choice of the local thresholds is given by \( \varepsilon_{\lambda,L} = 2^{\ell-L} \varepsilon_L \) with \( \varepsilon_L \geq 0 \) as considered in previous works, cf. [80, 81, 83, 108]. In contrast to these works, we generalize the definition of significance of local contributions that allows us to formulate the concept without specifying a basis. In the aforementioned works, uniform dyadic hierarchies are considered and the notion of significance is related to the \( \infty \)-norm applied to the vector of detail coefficients from a wavelet expansion. Here, we extend the concept to non-uniform hierarchies.

If a local contribution \( d_{\lambda}^\ell \) is not \( \| \cdot \|_\lambda \)-significant, we will discard it and thus, sparsify the multi-scale decomposition. For this purpose, we define the *sparse approximation* \( u^{L,\varepsilon} \in S_L \) of \( u^L \) by

\[
u^{L,\varepsilon} := \sum_{\lambda \in \mathcal{I}_0} u_0^\lambda + \sum_{\ell=0}^{L-1} \sum_{\lambda \in \mathcal{D}(\varepsilon,\ell) \cap \mathcal{I}_\ell} d_{\lambda}^\ell;
\] (3.40)

where the index set \( \mathcal{D}_\varepsilon \subset \bigcup_{\ell=0}^{L-1} \mathcal{I}_\ell \) is defined as the smallest set containing the indices of \( \| \cdot \|_\lambda \)-significant contributions, i.e.,

\[
\left\{ \lambda \in \bigcup_{\ell=0}^{L-1} \mathcal{I}_\ell : \| d_{\lambda}^\ell \|_\lambda > \varepsilon_{\lambda,L} \right\} \subset \mathcal{D}_\varepsilon.
\] (3.41)
and being a tree, i.e.,
\[ \mu \in \mathcal{D}_\epsilon \Rightarrow \lambda \in \mathcal{D}_\epsilon \ \forall \lambda \text{ with } V_{\mu} \not\subset V_{\lambda}. \] (3.42)

The set \( \mathcal{D}_\epsilon \) can be determined by adding additional indices of non-significant contributions to \( \{ \lambda \in \bigcup_{l=0}^{L-1} \mathcal{I}_l : \| d^l_\lambda \| > \varepsilon_{\lambda,l} \} \) such that the tree condition (3.42) is fulfilled.

The error introduced by the thresholding procedure can be estimated by proceeding in analogy to classical wavelet analysis, cf. [52].

**Theorem 3.2 (Thresholding error)** Let the local threshold \( \varepsilon_{\lambda,l} \) fulfill (3.38). Then, the threshold error of \( u^{L,\varepsilon} \), \( L \in \mathbb{N}_0 \), can be estimated by
\[ \| u^L - u^{L,\varepsilon} \|_{L_q(\Omega)} \leq |\Omega|^{\frac{1}{q}} C_{\text{loc},M} \varepsilon_{\text{max}}, \] (3.43)
for \( q \in \{1,2\} \).

**Proof:** First of all, we note by (3.40) and (3.41) that
\[ u^L - u^{L,\varepsilon} = \sum_{l=0}^{L-1} \sum_{\lambda \in \mathcal{I}_l \setminus (\mathcal{D}_\epsilon \cap \mathcal{I}_l)} d^l_\lambda. \]

For \( q = 1 \) we deduce using the Cauchy-Schwartz inequality
\[ \| u^L - u^{L,\varepsilon} \|_{L_1(\Omega)} \leq \sum_{l=0}^{L-1} \sum_{\lambda \in \mathcal{I}_l \setminus (\mathcal{D}_\epsilon \cap \mathcal{I}_l)} \sqrt{|V_{\lambda}|} \| d^l_\lambda \|_{L^2(V_{\lambda})}. \]

Alternatively, for \( q = 2 \) we obtain by the orthogonality of contributions corresponding to the complement spaces on different refinement levels (3.25) and compactness of the support (3.28) that
\[ \| u^L - u^{L,\varepsilon} \|_{L_2(\Omega)}^2 = \sum_{l=0}^{L-1} \sum_{\lambda \in \mathcal{I}_l \setminus (\mathcal{D}_\epsilon \cap \mathcal{I}_l)} \sum_{\lambda \in \mathcal{I}_l \setminus (\mathcal{D}_\epsilon \cap \mathcal{I}_l)} \langle d^l_\lambda, d^l_\lambda \rangle_{\Omega} = \sum_{l=0}^{L-1} \sum_{\lambda \in \mathcal{I}_l \setminus (\mathcal{D}_\epsilon \cap \mathcal{I}_l)} (\| d^l_\lambda \|_{L^2(V_{\lambda})})^2. \]

Together, we obtain
\[ \| u^L - u^{L,\varepsilon} \|_{L_q(\Omega)}^q \leq \sum_{l=0}^{L-1} \sum_{\lambda \in \mathcal{I}_l \setminus (\mathcal{D}_\epsilon \cap \mathcal{I}_l)} |V_{\lambda}| \left( \frac{\| d^l_\lambda \|_{L^2(V_{\lambda})}}{\sqrt{|V_{\lambda}|}} \right)^q \]
for \( q \in \{1,2\} \). Using the norm equivalence (3.37) we conclude
\[ \| u^L - u^{L,\varepsilon} \|_{L_q(\Omega)}^q \leq (C_{\text{loc},M})^q \sum_{l=0}^{L-1} \sum_{\lambda \in \mathcal{I}_l \setminus (\mathcal{D}_\epsilon \cap \mathcal{I}_l)} |V_{\lambda}| (\| d^l_\lambda \|)^q. \]
Then, due to (3.39) and (3.41) we can estimate the error by
\[ \|u^L - u^{L,\varepsilon}\|_{L^q(\Omega)}^q \leq (C_{\text{loc},M})^q \sum_{t=0}^{L-1} \sum_{\lambda \in \mathcal{D}_t \cap \mathcal{I}_{t}} |\mathcal{V}_\lambda| (\varepsilon_{\lambda,t})^q. \]

Next, we conclude that
\[ \|u^L - u^{L,\varepsilon}\|_{L^q(\Omega)}^q \leq (C_{\text{loc},M})^q |\Omega| \sum_{t=0}^{L-1} \max_{\lambda \in \mathcal{D}_t} (\varepsilon_{\lambda,t})^q. \]

Using (3.38), we conclude that (3.43) holds for \( q = 1 \). Equation (3.38) implies that \( \varepsilon_{\lambda,t} \leq \varepsilon_{\max} \) and, thus, we deduce for \( q = 2 \)
\[ \sum_{t=0}^{L-1} \max_{\lambda \in \mathcal{D}_t} (\varepsilon_{\lambda,t})^2 \leq \varepsilon_{\max}^2 \sum_{t=0}^{L-1} \max_{\lambda \in \mathcal{D}_t} \varepsilon_{\lambda,t} \leq (\varepsilon_{\max})^2. \]

Thus, (3.43) holds for for \( q = 2 \) as well. \( \square \)

The proof of the estimate of the threshold error (3.43) implies that \( \sum_{t=0}^{L-1} \max_{\lambda \in \mathcal{D}_t} \varepsilon_{\lambda,t} \) should tend to zero for increasing \( L \) to ensure that the threshold error tends to zero for increasing \( L \). Next, we conclude that the thresholding procedure is stable.

**Proposition 3.4 (Stability of the thresholding procedure)** The thresholding of local contributions from the orthogonal complements is stable in \( L^2 \), i.e.,
\[ \|u^{L,\varepsilon}\|_{L^2(\Omega)} \leq \|u^L\|_{L^2(\Omega)}. \]

**Proof:** From (3.40) we conclude by (3.32) that
\[ \|u^{L,\varepsilon}\|_{L^2(\Omega)}^2 = \sum_{\lambda \in \mathcal{D}_0} \|u^0_\lambda\|_{L^2(\Omega)}^2 + \sum_{t=0}^{L-1} \sum_{\lambda \in (\mathcal{D}_t \cap \mathcal{I}_t)} \|d^t_\lambda\|_{L^2(\Omega)}^2. \]
Since there holds
\[ \sum_{t=0}^{L-1} \sum_{\lambda \in (\mathcal{D}_t \cap \mathcal{I}_t)} \|d^t_\lambda\|_{L^2(\Omega)}^2 \leq \sum_{t=0}^{L-1} \sum_{\lambda \in \mathcal{D}_t} \|d^t_\lambda\|_{L^2(\Omega)}^2, \]
equation (3.44) holds true. \( \square \)

Since \( \mathcal{D}_t \) has a tree structure, we can identify \( u^{L,\varepsilon} \) with the orthogonal projection to a piecewise polynomial space corresponding to an adaptive grid. Hence, \( u^{L,\varepsilon} \) can be written as
\[ u^{L,\varepsilon} = \sum_{t=0}^{L} \sum_{\lambda \in \mathcal{D}_t} u^t_\lambda, \] (3.45)
where the adaptive grid is defined by the index set
\[
\mathcal{I}_\varepsilon := \left\{ \mu \in \bigcup_{\ell=0}^{L} \mathcal{I}_\ell : \exists \lambda \in \mathcal{D}_\varepsilon \text{ such that } \mu \in \mathcal{M}_\lambda \text{ and } \mu \notin \mathcal{D}_\varepsilon \right\}
\]  
(3.46)
satisfying \( \mathring{\Omega} = \bigcup_{\lambda \in \mathcal{I}_\varepsilon} \mathcal{V}_\lambda \) with \( \mathcal{V}_\lambda \cap \mathcal{V}_{\lambda'} = \emptyset \) for \( \lambda \neq \lambda' \in \mathcal{I}_\varepsilon \). Then, the sparsified representation \( u^{L,\varepsilon} \) can be characterized equivalently by either the index set \( \mathcal{I}_\varepsilon \) corresponding to the cells of the adaptive grid (3.46) or the index set \( \mathcal{D}_\varepsilon \) corresponding to the significant local contributions of the orthogonal complements, cf. Figure 3.4.

![Diagram of adaptive grid and index set](image)

**Figure 3.4:** Relation between the adaptive grid \( \mathcal{I}_\varepsilon \) and the index set of significant contributions \( \mathcal{D}_\varepsilon \). Cells corresponding to indices in \( \mathcal{I}_\varepsilon \) and \( \mathcal{D}_\varepsilon \) are highlighted in orange (on the left) and in blue (on the right), respectively.

Since in the sparse approximation (3.40) only contributions of the orthogonal complement spaces are modified, we conclude by Conclusion 3.2 that the thresholding procedure is locally conservative:

**Conclusion 3.3 (Local conservation of the thresholding procedure)** The thresholding procedure (3.40) is conservative, i.e.,
\[
\int_{\mathcal{V}_\lambda} u^{L^*}(x) dx = \int_{\mathcal{V}_\lambda} u^{L,\varepsilon}(x) dx, \quad \lambda \in \mathcal{I}_\varepsilon.
\]
(3.47)

We give the relevant algorithms for the implementation of thresholding and the corresponding grid adaptation in Section 3.9.
3.6 Orthogonal bases

In order to implement the multiresolution analysis basis functions or rather generators for the spaces $S_\ell$ and $W_\ell$, $\ell \in \mathbb{N}_0$ are required. For that purpose, we consider the general framework of stable bi-orthogonal bases for multiresolution representations that has been established in [36,61]. The choice of the basis determines how to compute the projections (3.14). Thus, it strongly influences the efficiency of the implementation. In the following, we refer to the basis functions $\phi_i$ for $S_\ell$ as scaling functions and to the basis functions $\psi_i$ for $W_\ell$ as multiwavelets, i.e.,

$$S_\ell = \text{span}_{i \in \mathcal{I}_\ell^S} \phi_i, \quad W_\ell = \text{span}_{i \in \mathcal{I}_\ell^W} \psi_i,$$

with index sets $\mathcal{I}_\ell^S$ and $\mathcal{I}_\ell^W$ corresponding to the global degrees of freedom of $S_\ell$ and $W_\ell$, respectively. The local degrees of freedom of $S_\ell$ and $W_\ell$ on a cell $V_\lambda$ are enumerated with index sets $\mathcal{P}$ and $\mathcal{P}_\lambda^*$, respectively. Due to the definition of the spaces there holds

$$|\mathcal{P}| = \dim(\Pi_{p-1}) = \binom{p+d-1}{d}, \quad |\mathcal{P}_\lambda^*| = |\mathcal{P}| \cdot (|\mathcal{M}_\lambda| - 1).$$

The local degrees of freedom of the orthogonal complement spaces can be enumerated by

$$\mathcal{P}_\lambda^* := \mathcal{P} \times \{ e \in \mathbb{N} : 1 \leq e < |\mathcal{M}_\lambda| \}.$$

Hence, we encode the spatial and polynomial degrees of freedom, i.e.,

$$\mathcal{I}_\ell^S := \{ i = (\lambda, i) | \lambda \in \mathcal{I}_\ell \text{ and } i \in \mathcal{P} \},$$

$$\mathcal{I}_\ell^W := \{ i = (\lambda, i) | \lambda \in \mathcal{I}_\ell \text{ and } i \in \mathcal{P}_\lambda^* \}.$$

For reasons of stability it is convenient to consider orthonormal bases, i.e.,

$$\langle \phi_i, \phi_j \rangle_\Omega = \delta_{i,j}, \quad \langle \psi_i, \psi_j \rangle_\Omega = \delta_{i,j}, \quad \forall i, j \in \mathcal{I}_\ell^S \text{ and } j, j' \in \mathcal{I}_\ell^W. \quad (3.51)$$

Hence, the herein considered setting is a special case of [36,61]. Moreover, for efficiency reasons we require that the basis functions are compactly supported, i.e.,

$$\text{supp}(\phi_{\lambda,i}) = \text{supp}(\psi_{\lambda,j}) = V_\lambda, \quad \forall \lambda \in \mathcal{I}_\ell, i \in \mathcal{P}, j \in \mathcal{P}_\lambda^*.$$

Additionally, we require that the zeroth scaling function is constant, i.e.,

$$\phi_{\lambda,0} = 1/\sqrt{|V_\lambda|}, \quad \forall \lambda \in \mathcal{I}_\ell.$$

(3.53)

Since the multiwavelets form a basis of $W_\ell$ they are orthogonal to the scaling functions

$$\langle \phi_i, \psi_j \rangle_\Omega = 0, \quad i \in \mathcal{I}_\ell^W, i \in \mathcal{I}_\ell^S.$$  

(3.54)
CHAPTER 3. MULTiresOLUTION ANALYSIS

We expand the global functions $u^\ell$ and $d^\ell$, $\ell \in \mathbb{N}_0$ in the bases (3.48)

$$u^\ell = \sum_{i \in \mathbb{Z}_\ell^d} u_i \phi_i, \quad d^\ell = \sum_{i \in \mathbb{Z}_\ell^d} d_i \psi_i. \quad (3.55)$$

Since $u^\ell \equiv P_{S_{\ell}}(u)$ and $d^\ell \equiv P_{W_{\ell}}(u)$ are defined by orthogonal projections of $u \in L^2(\Omega)$, the coefficients are given due to (3.17) by

$$u_i = \langle u, \phi_i \rangle_\Omega, \quad d_j = \langle u, \psi_j \rangle_\Omega. \quad (3.56)$$

Next, we make use of the locality of the bases (3.52) and expand the local contributions $u^\ell_\lambda$, $u^{\ell + 1}_\lambda$ and $d^\ell_\lambda$ as

$$u^\ell_\lambda = \sum_{i \in \mathcal{P}_\lambda} u_{\lambda,i} \phi_{\lambda,i}, \quad d^\ell_\lambda = \sum_{i \in \mathcal{P}_\lambda} d_{\lambda,i} \psi_{\lambda,i}, \quad u^{\ell + 1}_\lambda = \sum_{\mu \in \mathcal{M}_\lambda} \sum_{i \in \mathcal{P}_\lambda} u_{\mu,i} \phi_{\mu,i}. \quad (3.57)$$

For ease of notation, we comprise all local scaling functions and multiwavelets corresponding to a cell $V_\lambda$ in vectors, i.e.,

$$\Phi_\lambda := (\phi_{\lambda,i})_{i \in \mathcal{P}_\lambda} \in (S_{\ell})^{||\mathcal{P}_{\lambda}||}, \quad \Psi_\lambda := (\psi_{\lambda,i})_{i \in \mathcal{P}_\lambda} \in (W_{\ell})^{||\mathcal{P}_{\lambda}||}, \quad \lambda \in \mathcal{I}_\ell, \ell \in \mathbb{N}_0. \quad (3.58)$$

Additionally, we merge the vectors of scaling functions $\Phi_\mu$ of all subcells in the refinement set $\mathcal{M}_\lambda = \{\mu_1, \ldots, \mu_{||\mathcal{M}_\lambda||}\}$ in

$$\Phi_{\mathcal{M}_\lambda} := \left(\Phi_{\mu_1}^T, \ldots, \Phi_{\mu_{||\mathcal{M}_\lambda||}}^T\right)^T \in (S_{\ell+1})^{||\mathcal{M}_\lambda||}, \quad \lambda \in \mathcal{I}_\ell, \ell \in \mathbb{N}_0. \quad (3.59)$$

Thus, we may rewrite (3.57) in a more compact way as

$$u^\ell_\lambda = u_\lambda \cdot \Phi_\lambda, \quad d^\ell_\lambda = d_\lambda \cdot \Psi_\lambda, \quad u^{\ell + 1}_\lambda = u_{\mathcal{M}_\lambda} \cdot \Phi_{\mathcal{M}_\lambda}, \quad (3.60)$$

where the coefficients are comprised in vectors $u_\lambda \in \mathbb{R}^{||\mathcal{P}_{\lambda}||}$, $d_\lambda \in \mathbb{R}^{||\mathcal{P}_{\lambda}||}$ and $u_{\mathcal{M}_\lambda} \in \mathbb{R}^{||\mathcal{M}_\lambda||}$, respectively. Due to (3.58) and (3.56), these vectors are given by

$$u_\lambda = \langle u, \Phi_\lambda \rangle_\Omega, \quad d_\lambda = \langle u, \Psi_\lambda \rangle_\Omega, \quad u_{\mathcal{M}_\lambda} = \langle u, \Phi_{\mathcal{M}_\lambda} \rangle_\Omega, \quad (3.61)$$

where the inner products are applied component-wise.

Based on the nestedness of the spaces, we derive two-scale relations between the scaling functions and multiwavelets of successive refinement levels. For that purpose, we rewrite $\phi_i, i \in I^S_{\ell}$, and $\psi_j, j \in I^W_{\ell}$, in terms of scaling functions on level $\ell + 1$ by

$$\phi_i = \sum_{k \in I^S_{\ell+1}} \langle \phi_i, \phi_k \rangle_\Omega \phi_k, \quad \psi_j = \sum_{k \in I^W_{\ell+1}} \langle \psi_j, \phi_k \rangle_\Omega \phi_k.$$

Due to the locality of the bases (3.52), this reduces to

$$\phi_{\lambda,i} = \sum_{\mu \in \mathcal{M}_\lambda} \sum_{k \in \mathcal{P}_\lambda} \langle \phi_{\lambda,i}, \phi_{\mu,k} \rangle_\Omega \phi_{\mu,k}, \quad \psi_{\lambda,j} = \sum_{\mu \in \mathcal{M}_\lambda} \sum_{k \in \mathcal{P}_\lambda} \langle \psi_{\lambda,j}, \phi_{\mu,k} \rangle_\Omega \phi_{\mu,k}. \quad (3.62)$$
3.6. ORTHOGONAL BASES

where \( \mathbf{i} = (\lambda, i) \) and \( \mathbf{j} = (\lambda, j) \). Then, we comprise the inner products in (3.62) in matrices \( \mathbf{M}_{\lambda,0} \in \mathbb{R}^{P|k|P||\mathcal{M}_{\lambda}|} \) and \( \mathbf{M}_{\lambda,1} \in \mathbb{R}^{P|k|P||\mathcal{M}_{\lambda}|} \), respectively. Thereby, we can write (3.62) equivalently in matrix-vector notation as

\[
\Phi_\lambda = \mathbf{M}_{\lambda,0} \Phi_{\mathcal{M}_\lambda}, \quad \Psi_\lambda = \mathbf{M}_{\lambda,1} \Phi_{\mathcal{M}_\lambda}.
\] (3.63)

Since the bases are assumed to be orthonormal, the matrix

\[
\mathbf{M}_\lambda := \begin{pmatrix} \mathbf{M}_{\lambda,0} \\ \mathbf{M}_{\lambda,1} \end{pmatrix} \in \mathbb{R}^{P|k|P||\mathcal{M}_{\lambda}|}
\] (3.64)

is an orthogonal matrix. Thus, \( \mathbf{M}_\lambda \) is invertible and consequently scaling functions on a finer level can be represented by the sum of scaling functions and multiwavelets on the next coarser level, i.e.,

\[
\Phi_{\mathcal{M}_\lambda} = \mathbf{M}^T_{\lambda,0} \Phi_\lambda + \mathbf{M}^T_{\lambda,1} \Psi_\lambda.
\] (3.65)

Furthermore, we deduce from the orthogonality of \( \mathbf{M}_\lambda \) that

\[
\mathbf{M}^T_{\lambda,1} \mathbf{M}_{\lambda,1} = \mathbf{I}_{\mathcal{M}_{\lambda}\times P} - \mathbf{M}^T_{\lambda,0} \mathbf{M}_{\lambda,0},
\] (3.66)

where by \( \mathbf{I}_{\mathcal{M}_{\lambda}\times P} \in \mathbb{R}^{\mathcal{M}_{\lambda} \times P} \) we denote the identity matrix.

Based on (3.63) and (3.65) we derive the following relations linking the coefficients of successive refinement levels.

**Proposition 3.5 (Two-scale relations for the coefficients)** For the coefficient vectors (3.61) of the basis expansions corresponding to the multi-scale decomposition (3.60) the following relations hold true for \( \lambda \in \mathcal{I}_\ell, \ell \in \mathbb{N}_0 \):

\[
\begin{align*}
\varphi_\lambda &= \mathbf{M}_{\lambda,0} \varphi_{\mathcal{M}_\lambda}, \\
\phi_\lambda &= \mathbf{M}_{\lambda,1} \varphi_{\mathcal{M}_\lambda}, \\
\varphi_{\mathcal{M}_\lambda} &= \mathbf{M}^T_{\lambda,0} \varphi_\lambda + \mathbf{M}^T_{\lambda,1} \phi_\lambda
\end{align*}
\] (3.67) (3.68) (3.69)

with local transformation matrices \( \mathbf{M}_{\lambda,0} \) and \( \mathbf{M}_{\lambda,1} \) defined by (3.62) and (3.63).

**Proof:** Starting from the definition of the coefficient vectors (3.61), we conclude by (3.63) that

\[
\varphi_\lambda = \mathbf{M}_{\lambda,0} \langle u, \Phi_{\mathcal{M}_\lambda} \rangle_\Omega, \quad \phi_\lambda = \mathbf{M}_{\lambda,1} \langle u, \Phi_{\mathcal{M}_\lambda} \rangle_\Omega.
\]

Hence, (3.67) and (3.68) hold true. Equation (3.69) can be shown analogously using (3.65) instead of (3.63).

\( \square \)

Proposition 3.5 provides an explicit representation of the local projections in Theorem 3.1(iii), i.e.,

\[
\begin{align*}
u_\lambda^\ell &\equiv P_{\mathcal{S}_\ell} (u_{\lambda}^{\ell+1}) = (\mathbf{M}_{\lambda,0} \varphi_{\mathcal{M}_\lambda}) \cdot \Phi_\lambda, \\
d_\lambda^\ell &\equiv P_{\mathcal{W}_\ell} (u_{\lambda}^{\ell+1}) = (\mathbf{M}_{\lambda,1} \phi_{\mathcal{M}_\lambda}) \cdot \Psi_\lambda.
\end{align*}
\] (3.70)
Thus, in the discrete setting of coefficient vectors and basis expansions, cf. (3.60), these projections can be realized with matrix-vector multiplications. Due to the orthogonality of $M_\lambda$ there holds $\|M_\lambda\|_2 = \|M_\lambda^T\|_2 = 1$. Hence, the two-scale transformation and its inverse, i.e.,

$$u_{M_\lambda} = M_\lambda \begin{pmatrix} y_\lambda \\ d_\lambda \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} y_\lambda \\ d_\lambda \end{pmatrix} = M_\lambda^T u_{M_\lambda}$$

(3.71)

are well-conditioned.

### 3.7 Construction of multiwavelets

We now determine bases with the desired properties of Section 3.6, i.e., orthonormal and compactly supported, where in particular the construction can be performed efficiently and stable. The starting point is the construction of the scaling functions. These can be easily computed by orthogonalization and a following normalization of a local monomial basis. A well-known example in one spatial dimension are the shifted and normalized Legendre polynomials. The multiwavelets are not uniquely defined by the conditions in Section 3.6, cf. [64]. For the construction of multiwavelets, i.e., the bases for the orthogonal complement spaces, several approaches are possible.

Following the ideas of [2,3,185] leads to an algebraic algorithm for an explicit construction of multiwavelets: in [2] Alpert proposed an algorithm for multiwavelets on uniform dyadic hierarchies in one spatial dimension. Yu et al. applied this idea in [185] for the construction of multiwavelets on a triangle with a uniform subdivision. The idea can easily be extended to the general multi-dimensional case, cf. [83]. The algorithm is performed locally, i.e., the multiwavelets can be generated on a single cell independently of the multiwavelets on the other cells. For each cell $V_\lambda$, $\lambda \in \mathcal{I}_\ell$ and $\ell \in \mathbb{N}_0$, the multiwavelets can be constructed by applying Algorithm 3.1. Examples of multiwavelets are shown in Figure 3.5. The algorithm starts with a local basis $\psi_{\lambda,i}$, $\lambda \in \mathcal{I}_\ell$ and $i \in \mathcal{P}$, for $S_{\ell+1} \setminus S_\ell$ and thus, it is ensured that $W_{\ell+1} \subset \text{span } \psi_{\lambda,i}$, $\lambda \in \mathcal{I}_\ell$. For this purpose, we might choose

$$\psi_{\lambda,i,e}(x) := \begin{cases} x^i & \text{if } x \in V_{\mu_e} \\ -x^i & \text{if } x \in V_\lambda \setminus V_{\mu_e} \\ 0 & \text{else} \end{cases}, \quad (i,e) \in \mathcal{P}_\lambda^*,$$

where the subcells $V_\mu$ have the enumeration $M_\lambda = \{\mu_1, ..., \mu_{|M_\lambda|}\}$. Then, in step (ii) the contribution of $S_\ell$ in the basis functions are filtered by an orthogonalization with respect to the coarser scaling functions. Afterwards they are orthogonalized among each other in step (iii). This can be done using a $QR$-decomposition, e.g., via Householder transformations. Finally, the multiwavelets are normalized in step (iv).
Algorithm 3.1 Construction of multiwavelets

From $\phi_{\lambda,j}, \ j \in \mathcal{P}$, the multiwavelets $\psi_{\lambda,i}, i \in \mathcal{P}_{\lambda}$, can be constructed by the following steps:

(i) Initialize $\psi_{\lambda,i}, i \in \mathcal{P}_{\lambda}$, such that the following conditions are satisfied:
   - $\psi_{\lambda,i} \in S_{\ell+1} \setminus S_{\ell}$,
   - $\text{span}_{i \in \mathcal{P}_{\lambda}} \psi_{\lambda,i} \cap S_{\ell} = \emptyset$,
   - $\{\psi_{\lambda,i}\}_{i \in \mathcal{P}_{\lambda}}$ is a linearly independent family,
   - $\text{supp} \psi_{\lambda,i} = V_{\lambda}$.

(ii) Orthogonalize $\psi_{\lambda,i}$ with respect to $\phi_{\lambda,j}, j \in \mathcal{P}$.

(iii) Orthogonalize $\psi_{\lambda,i}, i \in \mathcal{P}_{\lambda}$.

(iv) Normalize $\psi_{\lambda,i}, i \in \mathcal{P}_{\lambda}$, with respect to $L^2(V_{\lambda})$.

It is possible to enforce additional properties of the multiwavelets, because there are fewer necessary conditions for the multiwavelets being an orthonormal basis for $W_{\ell}$, than degrees of freedom in the construction, cf. [64]. In Alpert’s original algorithm [2] these remaining degrees of freedom are used to increase the order of vanishing moments $M_{p,i} \geq p - 1$, i.e.,

\[ \langle \psi_{\lambda,i}, P \rangle = 0 \quad \forall P \in \Pi_{M_{p,i}}. \]

Therefore, an additional step between step (ii) and (iii) was included. Nevertheless, the order of vanishing moments $M_{p,i}$ can only be increased partially, i.e., on each cell $V_{\lambda}$ a multiwavelet with $M_{p,i} = p - 1$ is remaining. Since our interest is on the local contribution $d_{\ell}^i$ in terms of a function rather than in a single coefficient we omit this step. These degrees of freedom can also be used to increase the zero entries in the local mask matrices $M_{\lambda,1}$ occurring in the two-scale relations for the coefficients (3.71). This has been analyzed and realized for the one-dimensional case with uniform and non-uniform dyadic hierarchies in [137].

Another possibility for the construction of multiwavelets comes from a discrete point of view. The basic idea is to find a completion of $M_{\lambda}$, i.e., for a given $M_{\lambda,0}$ find $M_{\lambda,1} \in \mathbb{R}^{p^{\star} \times p^{\star} \Pi M_{\lambda,1}}$ such that

\[ M_{\lambda} = \begin{pmatrix} M_{\lambda,0} \\ M_{\lambda,1} \end{pmatrix} \]

is an orthogonal matrix. Then, due to (3.63) the multiwavelets can be determined by

\[ \Psi_{\lambda}(x) = M_{\lambda,1} \Phi_{M_{\lambda}}(x). \]
Figure 3.5: Examples of multiwavelets constructed with Alpert’s construction principle for uniform dyadic hierarchies and $p = 2$.

This idea originates from [36], where a very general setting is considered. It has been carried out in the particular setting of DG spaces for the one-dimensional uniform dyadic hierarchy in [20] and for the one-dimensional non-uniform dyadic case in [5]. In both works an explicit formula for $M_{\lambda,1}$ is derived. The drawback of this idea is that it cannot be easily realized in the multi-dimensional case. To find a completion for the general non-uniform multi-dimensional case is not trivial and it is open whether an explicit formula similar to the one-dimensional case can be derived. Furthermore, the computation of the completions from [5, 20] for the one-dimensional case is still costly since it requires the inversion of a matrix.

With increasing polynomial order $p$ and spatial dimension $d$ the local degrees of $W_{\ell}$ are increasing dramatically, e.g., in case of the dyadic hierarchy we have

$$|\mathcal{P}| = (2^d - 1)|P|.$$  

Hence, to perform Algorithm 3.1 or the completion of the matrix separately for all cells in the hierarchy, is computationally expensive. Therefore, we are interested in reducing the computational cost for the construction of multiwavelets. In some situations, $V_{\lambda}$ and its subcells $V_{\mu}$, $\mu \in \mathcal{M}_{\lambda}$, can be mapped by the same affine linear transformation $\gamma_{\lambda}$ to a reference cell $V_{\lambda}^{\text{ref}}$ and its subcells $V_{\mu}^{\text{ref}}$, respectively, i.e.

$$V_{\lambda}^{\text{ref}} = \gamma_{\lambda}(V_{\lambda}) \quad \text{and} \quad V_{\mu}^{\text{ref}} = \gamma_{\lambda}(V_{\mu}) \quad \forall \mu \in \mathcal{M}_{\lambda}. \quad (3.72)$$

Then, the construction can be performed on the reference element $V_{\lambda}^{\text{ref}}$. The multiwavelets on $V_{\lambda}$ can be obtained by shifts and rescaling of the multiwavelets constructed on the
3.7. **CONSTRUCTION OF MULTIWAVELETS**

reference element. If multiple cells can be mapped to the same reference element, the construction of the multiwavelets for all these cells can be done once on the reference element. Thus, the computational cost for the construction can be reduced significantly. A well-known example is the uniform dyadic hierarchy. There, the construction has to be performed only once on $[0, 1]^d$ and by shifts and normalization all multiwavelets can be obtained. The multiwavelets on $[0, 1]^d$ with a centered subdivision, constructed with Algorithm 3.1, are listed for $d = 2, 3$ in [136]. For a comparable triangular reference element multiwavelets are constructed in [185] using barycentric coordinates.

If the hierarchy of nested grids is very heterogeneous, i.e., only a few cells have the same reference element, the construction via Alpert’s algorithm or matrix completion has to be performed for several reference elements. An example for this situation is given by the stretched Cartesian grid hierarchy in Figure 3.1(b) using the grid function (3.8). Since here the subdivision of the cells in direction $x_1$ is not uniform, it is not possible to map all cells in this hierarchy to the same reference element by means of (3.72).

In order to avoid performing the construction of multiwavelets on all of these reference elements, the construction with Alpert’s algorithm has been performed algebraically for the one-dimensional case on $[0, 1]$ with an arbitrary subdivision of this reference cell in [141]. The conclusion of this work is that this construction becomes extremely expensive for increasing polynomial degree. Moreover, the explicit representation of the constructed multiwavelets is non-linearly depending on the position of the subdivision $\alpha \in (0, 1)$. Thus, the evaluation of these multiwavelets becomes computationally costly. In addition, an algebraic construction of multiwavelets with an arbitrary subdivision of the cells in multiple spatial dimensions is not straightforward. Actually, it turned out that this approach is not practicable for the general (multi-dimensional) non-uniform case.

For reasons of efficiency it is thus of major interest to avoid the explicit construction of a basis for the complement spaces. To this end, we propose an alternative approach avoiding the explicit use of multiwavelets: since $W_\ell \subset S_{\ell+1}$, we might express $d^\ell_\lambda$ in terms of scaling functions on level $\ell + 1$ using (3.63) by

$$d^\ell_\lambda = d^{\ell}_{M, \lambda} \cdot \Phi_{M, \lambda}, \tag{3.73}$$

where due to (3.60) and (3.63) the coefficients are given by

$$d^{\ell}_{M, \lambda} := \langle d^\ell_\lambda, \Phi_{M, \lambda} \rangle_\Omega = (M_{\lambda, 1})^T d_\lambda. \tag{3.74}$$

Applying (3.68) and (3.66) leads to

$$d^{\ell}_{M, \lambda} = (I - (M_{\lambda, 0})^T M_{\lambda, 0}) d_{\lambda}. \tag{3.75}$$

Since $(I - (M_{\lambda, 0})^T M_{\lambda, 0}) \in \mathbb{R}^{k(M, \lambda)}$, the computation of (3.75) is more costly than the classical approach using (3.68) with multiwavelets. Since $d_{\lambda}^{\ell}$ has to be computed from
\[ \mathbf{u}_{\mathcal{M}_\lambda} \text{ by (3.67) in the two-scale decomposition, we might compute } d_{\mathcal{M}_\lambda} \text{ in a more efficient } \\
\text{way by} \\
\quad \quad d_{\mathcal{M}_\lambda} = \mathbf{u}_{\mathcal{M}_\lambda} - (\mathbf{M}_{\lambda,0})^T \mathbf{u}_\lambda. \] 

Note that the multiresolution analysis in this chapter is based merely on the local contributions \( d^\ell_\lambda \) and is \textbf{not} influenced by the choice of the local basis functions. Due to (3.73) and (3.76), we can represent \( d^\ell_\lambda \) without an explicit use of multiwavelets \( \Psi_{\mathcal{M}_\lambda} \) or the matrix \( \mathbf{M}_{\lambda,1} \). Thus, the construction of the multiwavelets can be avoided using (3.73) with (3.76). From the orthogonality of \( \mathbf{M}_\lambda \) we conclude that this alternative representation of \( d^\ell_\lambda \) is well-conditioned:

\textbf{Proposition 3.6} \textit{The task of computing } \( d_{\mathcal{M}_\lambda} \text{ from } \mathbf{u}_{\mathcal{M}_\lambda} \text{ is well-conditioned, i.e.,} \\
\quad \|d_{\mathcal{M}_\lambda}\|_2 \leq \|\mathbf{u}_{\mathcal{M}_\lambda}\|_2. \}

\textbf{Proof:} First of all, by (3.64) and (3.74) we note that \\
\quad \|d_{\mathcal{M}_\lambda}\|_2 \leq \|\mathbf{M}_\lambda^T\|_2 \left\| \begin{pmatrix} 0_{|\mathcal{P}|} \\ d_\lambda \end{pmatrix} \right\|_2, \\
\text{where } 0_{|\mathcal{P}|} \in \mathbb{R}^{|\mathcal{P}|} \text{ is a vector with } |\mathcal{P}| \text{ vanishing entries. Then, we deduce from (3.71) that} \\
\quad \left\| \begin{pmatrix} 0_{|\mathcal{P}|} \\ d_\lambda \end{pmatrix} \right\|_2 \leq \left\| \begin{pmatrix} \mathbf{u}_\lambda \\ d_\lambda \end{pmatrix} \right\|_2 \leq \|\mathbf{M}_\lambda\|_2 \|\mathbf{u}_{\mathcal{M}_\lambda}\|_2. \\
\text{By combining these two observations and twice using the orthogonality of } \mathbf{M}_\lambda \text{ we finally conclude} \\
\quad \|d_{\mathcal{M}_\lambda}\|_2 \leq \|\mathbf{u}_{\mathcal{M}_\lambda}\|_2. \] 

\textbf{□}

\section{3.8 Choice of the local norms}

In order to implement the thresholding procedure (3.40), we have to specify the local norms (3.39) characterizing the significance of local contributions. In previous works [80, 81, 83, 108, 156]

\[ \|d^\ell_\lambda\|_{\lambda} = \frac{\|d_\lambda\|_{\infty}}{\sqrt{|V_\lambda|}} \] 

has been used, being equivalent to \( \|\cdot\|_{L^2(V_\lambda)}/\sqrt{|V_\lambda|} \) by means of (3.37). In order to compute this norm, \( \mathbf{M}_{\lambda,1} \) is explicitly needed according to (3.68). As discussed in Section 3.7 the
construction of \( \mathbf{M}_{\lambda,1} \) as well as the realization of the MRA and the grid adaptation using \( \mathbf{M}_{\lambda,1} \) is computationally expensive. Thus, we propose an alternative choice for the local norm:

\[
\| d^i_\lambda \|_\lambda = \frac{\| d^i_\lambda \|_{L^2(\Omega)}}{\sqrt{|V_\lambda|}}.
\] (3.78)

Due to Parseval’s identity, (3.78) can be computed equivalently by either \( \| d^i_\lambda \|_2/\sqrt{|V_\lambda|} \) or \( \| d^i_\lambda \|_2/\sqrt{|V_\lambda|} \). Thus, the use of \( \mathbf{M}_{\lambda,1} \) can be avoided. Hence, this choice enables the realization of the thresholding procedure while avoiding the costly explicit construction of multiwavelets or rather \( \mathbf{M}_{\lambda,1} \).

The two presented norms are equivalent when \( \sup_\lambda |\mathcal{M}_\lambda| \) is bounded since

\[
\| d_\lambda \|_\infty \leq \| d_\lambda \|_2 \leq \sqrt{|\mathcal{M}_\lambda| - 1} \sqrt{|P|} \| d_\lambda \|_\infty.
\]

In the following we refer to (3.77) together with the use of multiwavelets in the multi-scale decomposition as the \textit{classical} approach. The newly proposed choice for the local norm (3.78) together with (3.73) avoiding the explicit use of multiwavelets is referred to as the \textit{wavelet-free} approach.

### 3.9 Algorithms and implementation aspects

In this section we present the algorithms needed to realize the multiresolution analysis and the thresholding procedure.

First, we focus on the implementation of the multiresolution analysis. In the previous sections two different approaches were presented. The difference between both approaches is the choice of bases in which the function \( d^i_\lambda \) is represented. The \textit{classical} approach is based on an explicit construction of the multiwavelets, cf. (3.60) and (3.68). By (3.73) and (3.76) we proposed the alternative \textit{wavelet-free} approach avoiding the costly construction of multiwavelets or rather the completion of the matrix \( \mathbf{M}_\lambda \). For both approaches we present the discrete algorithms for the two-scale transformation and its inverse. Furthermore, we compare the efficiency in terms of memory requirement and computational cost.

Secondly, we present the algorithms for the implementation of the grid adaptation. Thereby, we focus on the generation of the sets \( \mathcal{D}_\epsilon \) and \( \mathcal{L}_\epsilon \) characterizing the sparsified representation \( u^{h,\epsilon} \).

#### 3.9.1 Algorithms for the multi-scale transformations

Usually, the starting point for a multiresolution analysis is some given function \( u^L \in S_L \subset L^2(\Omega) \) on a fixed refinement level \( L \in \mathbb{N}_0 \). This might come from a numerical scheme
or from a projection of an $L^2$-function. The task then is to compute the multi-scale decomposition (3.21), i.e., compute $u^0, d^0, \ldots, d^{L-1}$. In order to analyze the computational structure of the transformation we define a multi-scale operator

$$T : S_L \to S_0 \times W_0 \times \cdots \times W_{L-1}, \quad T(u^L) = (u^0, d^1, \ldots, d^{L-1})^T.$$  

(3.79)

Due to (3.19), the projections in (3.79) can be computed recursively from fine to coarse, cf. Figure 3.6. Consequently, the computation of the multi-scale decomposition consists

$$u^L \xrightarrow{P_{S_{L-1}}} u^{L-1} \xrightarrow{P_{S_{L-2}}} u^{L-2} \xrightarrow{P_{S_{L-2}}} \cdots \xrightarrow{P_{S_1}} u^1 \xrightarrow{P_{S_0}} u^0$$


Figure 3.6: Level-wise realization of the multi-scale decomposition.

of repeated two-scale transformations. Due to Theorem 3.1, the two-scale transformation, i.e., the projections to the next coarser spaces, can be performed cell-wise. The resulting algorithm is listed in Algorithm 3.2. The inversion of $T$ is specified by

**Algorithm 3.2** Two-scale transf.

Input: $u^{\ell+1}$.

Compute:

1: for all $\lambda \in \mathcal{I}_\ell$ do
2: $u^\ell_\lambda = P_{S_\lambda}(u^{\ell+1}_\lambda)$,
3: $d^\ell_\lambda = P_{W_\lambda}(u^{\ell+1}_\lambda)$,
4: end for

Output: $u^\ell$ and $d^\ell$.

**Algorithm 3.3** Inverse two-scale transf.

Input: $u^\ell$ and $d^\ell$.

Compute:

1: for all $\lambda \in \mathcal{I}_\ell$ do
2: for all $\mu \in \mathcal{M}_\lambda$ do
3: $u^{\ell+1}_\mu = u^\ell_\lambda 1_{\nu_\mu} + d^\ell_\lambda 1_{\nu_\mu}$,
4: end for
5: end for

Output: $u^{\ell+1}$.

Figure 3.7: Algorithms for the two-scale transformations.

$$T^{-1}(u^0, d^1, \ldots, d^{L-1}) = u^0 + \sum_{\ell=0}^{L-1} d^\ell.$$  

(3.80)

The inverse can be computed recursively from coarse to fine by summation of the contributions of the subspaces, i.e.,

$$u^{\ell+1} = u^\ell + d^\ell, \quad 0 \leq \ell \leq L-1.$$
This can be performed locally for each cell. The resulting algorithm is listed in Algorithm 3.3.

In order to implement the multi-scale transformation, one usually employs basis expansions for the different spaces. The choice of bases defines the discrete algorithms for the transformations. The difference between the classical and the wavelet-free approach is the choice of the generating set for the orthogonal complement spaces \( W_\ell \). In the classical approach the multiwavelets are used, whereas in the wavelet-free approach the scaling functions of \( S_{\ell+1} \) are used. Thus, the implementation of the local two-scale transformation

<table>
<thead>
<tr>
<th>Algorithm 3.4 classic</th>
<th>Algorithm 3.5 wavelet free</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> ( u^{\ell+1}<em>\lambda = \underline{u}</em>{M_\lambda} \cdot \Phi_{M_\lambda} ).</td>
<td></td>
</tr>
<tr>
<td><strong>Compute:</strong></td>
<td></td>
</tr>
<tr>
<td>1: ( \underline{u}<em>\lambda = M</em>{\lambda,0} \underline{u}<em>{M</em>\lambda} )</td>
<td></td>
</tr>
<tr>
<td>2: ( \underline{d}<em>\lambda = M</em>{\lambda,1} \underline{u}<em>{M</em>\lambda} )</td>
<td></td>
</tr>
<tr>
<td><strong>Output:</strong> ( u^\ell_\lambda = \underline{u}<em>\lambda \cdot \Phi</em>\lambda ) and ( d^\ell_\lambda = \underline{d}<em>\lambda \cdot \Psi</em>\lambda ).</td>
<td></td>
</tr>
<tr>
<td><strong>Input:</strong> ( u^{\ell+1}<em>\lambda = \underline{u}</em>{M_\lambda} \cdot \Phi_{M_\lambda} ).</td>
<td></td>
</tr>
<tr>
<td><strong>Compute:</strong></td>
<td></td>
</tr>
<tr>
<td>1: ( \underline{u}<em>\lambda = M</em>{\lambda,0} \underline{u}<em>{M</em>\lambda} )</td>
<td></td>
</tr>
<tr>
<td>2: ( \underline{d}<em>\lambda = M</em>{\lambda,1}^{T} \underline{u}<em>{M</em>\lambda} )</td>
<td></td>
</tr>
<tr>
<td><strong>Output:</strong> ( u^\ell_\lambda = \underline{u}<em>\lambda \cdot \Phi</em>\lambda ) and ( d^\ell_\lambda = \underline{d}<em>\lambda \cdot \Phi</em>{M_\lambda} ).</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.8: Algorithms for the local two-scale transformation.

in Algorithm 3.2 and its inverse in Algorithm 3.3 is different for the two approaches. In Algorithm 3.4 and 3.5 we summarize the discrete algorithms of the local-two scale transformation for the classical and the wavelet-free approach, respectively. The algorithms for the inverse transformation are listed in Algorithms 3.6 and 3.7, respectively.

<table>
<thead>
<tr>
<th>Algorithm 3.6 classic</th>
<th>Algorithm 3.7 wavelet free</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> ( u^\ell_\lambda = \underline{u}<em>\lambda \cdot \Phi</em>\lambda ) and ( d^\ell_\lambda = \underline{d}<em>\lambda \cdot \Psi</em>\lambda ).</td>
<td></td>
</tr>
<tr>
<td><strong>Compute:</strong></td>
<td></td>
</tr>
<tr>
<td>1: ( \underline{u}<em>{M</em>\lambda} = M^{T}<em>{\lambda,0} \underline{u}</em>\lambda + M^{T}<em>{\lambda,1} \underline{d}</em>\lambda )</td>
<td></td>
</tr>
<tr>
<td><strong>Output:</strong> ( u^{\ell+1}<em>\lambda = \underline{u}</em>{M_\lambda} \cdot \Phi_{M_\lambda} ).</td>
<td></td>
</tr>
<tr>
<td><strong>Input:</strong> ( u^\ell_\lambda = \underline{u}<em>\lambda \cdot \Phi</em>\lambda ) and ( d^\ell_\lambda = \underline{d}<em>\lambda \cdot \Phi</em>{M_\lambda} ).</td>
<td></td>
</tr>
<tr>
<td><strong>Compute:</strong></td>
<td></td>
</tr>
<tr>
<td>1: ( \underline{u}<em>{M</em>\lambda} = M^{T}<em>{\lambda,0} \underline{u}</em>\lambda + \underline{d}<em>{M</em>\lambda} )</td>
<td></td>
</tr>
<tr>
<td><strong>Output:</strong> ( u^{\ell+1}<em>\lambda = \underline{u}</em>{M_\lambda} \cdot \Phi_{M_\lambda} ).</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.9: Algorithms for the inverse of the local two-scale transformation.

Next, we compare the computational cost and memory requirements of both approaches. First of all, we note that the construction of the multiwavelets needed in the classical approach is very costly. The wavelet-free approach avoids the explicit construction. In the following we compare the two approaches neglecting the construction
issue. Therefore, we focus on a local two-scale transformation on a single element described by Algorithms 3.4 – 3.7. By this comparison the computational cost and memory requirements of the full transformations can be estimated.

In Table 3.1 the memory requirement, i.e., number of coefficients for the storage of $d_\lambda^{t}$, $u_\lambda^{t}$ and $u_\lambda^{t+1}$ for $\lambda \in \mathcal{I}_\ell$ and the number of matrix entries needed for the transformations, are compared.

<table>
<thead>
<tr>
<th>Ansatz</th>
<th># coefficients</th>
<th># matrix entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>classical</td>
<td>$2</td>
<td>P</td>
</tr>
<tr>
<td>wavelet free</td>
<td>$2</td>
<td>P</td>
</tr>
</tbody>
</table>

Table 3.1: Comparison of memory requirement for a two scale decomposition on a single cell.

In order to quantify the overhead in storing more coefficients in the wavelet-free approach, we consider the ratio of the number of coefficients in both approaches:

$$\text{ratio}_{\text{coeff}} := \frac{\text{# coefficients for wavelet-free approach}}{\text{# coefficients for classical approach}} = 1 + \frac{1}{2|\mathcal{M}_\lambda|}. \quad (3.81)$$

The reduction of memory requirement resulting from storing less matrix entries in the wavelet-free approach can be quantified by analyzing the ratio of matrix entries:

$$\text{ratio}_{\text{mat}} := \frac{\text{# matrix entries for wavelet-free approach}}{\text{# matrix entries for classical approach}} = \frac{1}{|\mathcal{M}_\lambda|}. \quad (3.82)$$

For dyadic hierarchies the ratios (3.81) and (3.82) are listed in Table 3.2 for $d = 1, 2, 3$. On

<table>
<thead>
<tr>
<th>$d$</th>
<th>ratio$_{\text{coeff}}$</th>
<th>ratio$_{\text{mat}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.25</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>1.125</td>
<td>0.25</td>
</tr>
<tr>
<td>3</td>
<td>1.0625</td>
<td>0.125</td>
</tr>
</tbody>
</table>

Table 3.2: Ratio of number of coefficients and matrix entries in case of a dyadic hierarchy for different spatial dimensions $d$.

On the one hand, avoiding the explicit construction of multiwavelets has the advantage that $\mathcal{M}_{\lambda,1}$ is not needed anymore, cf. (3.76), and therefore $|\mathcal{M}_\lambda|$-times less matrix entries have to be stored. On the other hand, there is more memory needed for the storage of the coefficients, since $d_\lambda^{t}$ is expanded in the basis of the larger spaces $S_{t+1}$.

Next, we compare the computational cost for the two-scale transformation and its inverse in Table 3.3. For that purpose, we compare the number of multiplications and additions for the computation of $d_\lambda^{t}$, $u_\lambda^{t}$ from $u_\lambda^{t+1}$ in Algorithms 3.4 and 3.5 and of its
Table 3.3: Comparison of the number of operations for a two-scale decomposition on a single cell.

inverse in Algorithms 3.6 and 3.7. In order to quantify the difference of the number of operations needed for both approaches, we consider the ratio of number of total operations for both approaches

$$\text{ratio}_{\text{op}} := \frac{\text{total \# operation wavelet-free approach}}{\text{total \# operation classical approach}} = \frac{6|\mathcal{P}| |M_\lambda|-1}{4|\mathcal{P}| |M_\lambda|^2 - 2|\mathcal{P}| |M_\lambda|}$$ (3.83)

Figure 3.10: ratio of number of operations according to (3.83) for dyadic sub-division, i.e. $|M_\lambda| = 2^d$.

position using the wavelet-free approach avoids the multiplication with $M_{\lambda,1}$ and $M_{\lambda,1}^T$, respectively. With increasing polynomial degree $p$ and space dimension $d$ the number of operations needed for the transformation is reduced significantly compared to the classical approach using multiwavelets.

In summary, we conclude that the wavelet-free approach not only avoids the costly construction of multiwavelets, but also is less expensive in terms of computational cost in comparison with the classical approach based on multiwavelets. The only advantage of the classical approach over the wavelet-free approach is that less coefficients in the basis expansions have to be stored. However, in the classical approach more matrix entries, i.e., storage of submatrix $M_{\lambda,1}$, have to be stored. Consequently, one may benefit from this
advantage whenever it is not necessary to store \( \mathbf{M}_\lambda \) on each cell separately, e.g., in the case of a uniform dyadic hierarchy. Apart from that, the reduction of memory requirements for storing less matrix entries compensates for the overhead of storing more coefficients in the wavelet-free approach. Thus, even in terms of memory requirement, the wavelet-free approach is more efficient than the classical approach. Hence, it seems to be preferable to use the wavelet-free approach for the realization of the multiresolution analysis.

### 3.9.2 Algorithms for thresholding and grid adaptation

Finally, we summarize the algorithms for the realization of the thresholding procedure (3.40) and the resulting adaptive grid (3.46). Starting with the local multi-scale decomposition (3.30), local contributions \( d_\lambda^\ell \) of the orthogonal complements \( W_\ell, \ell = 0, \ldots, L - 1 \), are thresholded. The thresholding is specified by the index set \( \mathcal{D}_\varepsilon \) of significant contributions corresponding to a threshold value \( \varepsilon \). A detailed algorithm for the generation of \( \mathcal{D}_\varepsilon \) is presented in Algorithm 3.8.

#### Algorithm 3.8 Thresholding procedure

Input: multi-scale decomposition (3.30) of \( u^L \) and threshold values \( \varepsilon_{\lambda,L} \).

Step 1: determine significant contributions:
1. for all \( \ell = 0, \ldots, L - 1 \) do
2. \hspace{1em} for all \( \lambda \in \mathcal{I}_\ell \) do
3. \hspace{2em} if \( \|d_\lambda^\ell\|_\lambda > \varepsilon_{\lambda,L} \) then
4. \hspace{3em} insert \( \lambda \) into \( \mathcal{D}_\varepsilon \),
5. \hspace{2em} end if
6. \hspace{1em} end for
7. end for

Step 2: extend \( \mathcal{D}_\varepsilon \) to be a tree:
1. for all \( \ell = L - 1, \ldots, 1 \) do
2. \hspace{1em} for all \( \mu \in \mathcal{D}_\varepsilon \cap \mathcal{I}_\ell \) do
3. \hspace{2em} determine \( \lambda \in \mathcal{I}_{\ell-1} \) such that \( \mu \in \mathcal{M}_\lambda \),
4. \hspace{2em} if \( \lambda \notin \mathcal{D}_\varepsilon \) then
5. \hspace{3em} insert \( \lambda \) into \( \mathcal{D}_\varepsilon \),
6. \hspace{2em} end if
7. \hspace{1em} end for
8. end for

Output: \( \mathcal{D}_\varepsilon \).

According to (3.45) the thresholding procedure can be identified with the orthogonal
projection corresponding to the adaptive grid (3.46). The construction of this adaptive grid from the index set $\mathcal{D}_\varepsilon$ is described in Algorithm 3.9. Thereby, the orthogonal projection to the adaptive grid is realized simultaneously by using the inverse of the local two-scale transformation in Algorithm 3.6 and 3.7, respectively. For both, the classical as well as the wavelet-free approach, thresholding and adaptation are realized in a similar way. The only difference is the computation of the local norms $\| \cdot \|_\lambda$, e.g. for

$$\| d^\ell_\lambda \|_\lambda = \frac{\| d^\ell_\lambda \|_{L^2(\nu_\lambda)}}{\sqrt{\nu_\lambda}}$$

these norms are computed by $\| d^\ell_\lambda \|_\lambda = \frac{1}{\sqrt{\nu_\lambda}}$ and $\| d^\ell'_\lambda \|_\lambda = \frac{1}{\sqrt{\nu_\lambda}}$ for the classical and the wavelet-free approach, respectively.

**Algorithm 3.9** $\mathcal{D}_\varepsilon$ (significant contributions) $\rightarrow$ $\mathcal{I}_\varepsilon$ (adaptive grid)

**Input:** multi-scale decomposition (3.30) of $u^L$ and the index set $\mathcal{D}_\varepsilon$.

**Determine $\mathcal{I}_\varepsilon$ from $\mathcal{D}_\varepsilon$:**

1. for all $\ell = 0, \ldots, L - 1$ do
2.  for all $\lambda \in \mathcal{D}_\varepsilon \cap \mathcal{I}_\ell$ do
3.    for all $\mu \in \mathcal{M}_\lambda$ do
4.      compute $u^\ell_{\mu+1}$ from $u^\ell_\lambda$ and $d^\ell_\lambda$,
5.    if $\mu \notin \mathcal{D}_\varepsilon$ then
6.      insert $\mu$ into $\mathcal{I}_\varepsilon$,
7.    end if
8.  end for
9. end for
10. end for

**Output:** $\mathcal{I}_\varepsilon$ and $u^{L,\varepsilon} = \sum_{\ell=0}^L \sum_{\lambda \in \mathcal{I}_\varepsilon \cap \mathcal{D}_\varepsilon} u^\ell_\lambda$. 
Chapter 4

Dynamical grid adaptation for DG schemes

Solutions of hyperbolic conservation laws and convection-dominated problems often reveal a heterogeneous structure: in some regions solutions are smooth whereas in other regions strong gradients or even discontinuities occur. In smooth regions a relatively coarse resolution is sufficient to realize a certain target accuracy. In contrast, in non-smooth regions a fine resolution is required to obtain the same accuracy. Thus, considering uniform grids for non-smooth solutions results in high computational cost. In order to reduce the computational complexity, we perform dynamical local grid adaptation. Thereby, we are interested in maintaining the accuracy of a uniformly refined discretization while reducing the computational cost.

For that purpose, we consider the concept of multi-scale based grid adaptation. This concept has originally been developed for finite volume schemes, cf. [56, 59, 72, 91, 116, 117, 133, 134, 152, 153]. It is motivated by pioneering work of A. Harten [96–99] on the construction of cost-effective hybrid finite volume schemes for conservation laws using a MRA. The underlying idea of this adaptation strategy is to perform a MRA of the reference scheme and evolve only significant local contributions in time. Later on, the concept has been extended to DG schemes in [35, 108, 156]. In [108, 156] an adaptive DG scheme for one-dimensional scalar conservation laws has been derived and analyzed. Here, we generalize the concept to nonlinear systems of equations in multiple space dimensions. Parts of this work have already been published in [80, 81, 83]. In previous works [80, 81, 83, 108, 156] the adaptation process was directly linked to the choice of the basis for the orthogonal complement spaces, i.e., the multiwavelets. Here, we present an alternative procedure which does not rely on the explicit construction of multiwavelets and can be applied without explicitly specifying a basis for the complement spaces and, thus, is applicable to any nested grid hierarchy.

This chapter is structured as follows: we first investigate how significance of local information is evolved in time for the semi-discrete and fully-discrete solution in Section 4.1
and 4.2, respectively. Then, we present a heuristic strategy for the prediction of significance of information on the next time level in Section 4.3. Afterwards, we summarize the resulting adaptive scheme in Section 4.4 and then present a strategy for the choice of the local thresholds in Section 4.5. The local projection limiting is intertwined with the adaptive scheme in Section 4.6. Moreover, we address implementation aspects such as efficient initialization and parallelization in Section 4.7 and 4.8, respectively. Finally, we present numerical results to validate the adaptive scheme in Section 4.9.

4.1 Significant contributions in the semi-discrete DG solution

The starting point for the grid adaptation is a hierarchy of nested grids as in (3.3) and (3.4). For practical reasons we fix a maximum refinement level $L$. Corresponding to the grid on the finest refinement level $L$, we consider a MRA of the semi-discrete DG solution $u_L$ defined by (2.23) and (2.29), respectively. According to (2.35) the weak formulations defining the evolution of these solutions can be written in a compact form as

$$
\langle \frac{\partial u_L}{\partial t}, v \rangle_\Omega = \hat{\mathcal{L}}(u_L, v) \quad \forall \ v \in S_L,
$$

(4.1)

where the inner product is applied component-wise and $\hat{\mathcal{L}} : (S_L)^m \times S_L \to \mathbb{R}^m$ is defined in (2.35). Note that $\hat{\mathcal{L}}(\cdot, \cdot)$ is linear in the second argument. Due to (3.30) we can rewrite the semi-discrete DG solutions as

$$
u_L(t, x) = \sum_{\lambda \in \mathcal{I}_0} u_L^0(t, x) + \sum_{\lambda \in \mathcal{I}_\ell} \sum_{t=0}^{L-1} d_L^\lambda(t,x), \quad x \in \Omega \text{ and } t \in (0, T].
$$

(4.2)

In the following we refer to this scheme as well as to its fully-discrete counterpart as the reference scheme. Next, we derive weak formulations for the local contributions. By choosing test functions $v \in S_0$ and $w \in W_\ell$, $1 \leq \ell \leq L - 1$, with compact support and using the orthogonality (3.24) and (3.25), we conclude that for $\lambda \in \mathcal{I}_0$ the evolution of $u_L^\lambda$ is specified by

$$
\langle \frac{\partial u_L^\lambda}{\partial t}, v_\lambda \rangle_{V_\lambda} = \hat{\mathcal{L}}(u_L, v_\lambda) \quad \forall \ v \in S_0,
$$

(4.3)

and for $\lambda \in \mathcal{I}_\ell$, $1 \leq \ell \leq L - 1$, the evolution of the local contribution $d_L^\lambda$ is specified by

$$
\langle \frac{\partial d_L^\lambda}{\partial t}, w_\lambda \rangle_{V_\lambda} = \hat{\mathcal{L}}(u_L, w_\lambda) \quad \forall \ w \in W_\ell.
$$

(4.4)

Here, the local contributions of the test functions are defined similarly to (3.28) by $w_\lambda := w 1_{V_\lambda}$ and $v_\lambda := v 1_{V_\lambda}$. 


4.1. SIGNIFICANCE IN THE DG SOLUTION

While dealing with systems of equations, we have to account for the fact, that individual quantities might have different orders of magnitude. Thus, we augment the definition of significance (3.39) with a rescaling factor for vector-valued \( \mathbf{u} \in (S_L)^m \): we call a local contribution \( \mathbf{d}_\lambda^i \) \( \lambda \)-significant iff

\[
\max_{1 \leq i \leq m} \frac{\| (\mathbf{d}_\lambda^i) \|_\lambda}{c_i} > \varepsilon_{\lambda, L}, \tag{4.5}
\]

where the scaling factor is defined by

\[
c_i := \max \left( 1, \frac{1}{|\Omega|} \int_{\Omega} u_i(x) dx \right).
\]

Comparable results as in Theorem 3.2 and Conclusion 3.3 hold true for the thresholding for a vector-valued quantity using the modified definition of significance.

**Proposition 4.1** For \( \mathbf{u}_L \in (S_L)^m \) let \( \mathbf{u}_{L, \varepsilon} \in (S_L)^m \) be the sparsified representation using the modified definition of significance (4.5) by means of (3.40). Then, there holds:

(i) The threshold error of \( \mathbf{u}_{L, \varepsilon}, L \in \mathbb{N}_0 \), can be estimated component-wise by

\[
\| (\mathbf{u}_L - \mathbf{u}_{L, \varepsilon})_i \|_{L_q(\Omega)} \leq |\Omega|^{1/q} c_i C_{\text{loc}, M} \varepsilon_{\max}, \quad 1 \leq i \leq m,
\]

for \( q \in \{1, 2\} \).

(ii) The thresholding procedure is locally conservative in each component, i.e.,

\[
\int_{V_\lambda} (\mathbf{u}_L(x))_i dx = \int_{V_\lambda} (\mathbf{u}_{L, \varepsilon}(x))_i dx, \quad \lambda \in \mathcal{I}_\varepsilon, \quad 1 \leq i \leq m.
\]

The proof of Proposition 4.1 is similar to the proofs of Theorem 3.2 and Conclusion 3.3. Here, the index set of significant contributions is slightly inflated since in (4.5) the significance of all components is intertwined. However, this does not influence the arguments in the proof of Proposition 3.2 and Conclusion 3.3.

In order to reduce the computational cost of the reference scheme (4.1), we only consider significant contributions \( \mathbf{d}_\lambda^i \) in (4.4) characterized by (4.5). For this purpose, we proceed analogously to (3.40) and define a sparsified representation of \( \mathbf{u}_h(t, x) \) by

\[
\mathbf{u}_{L, \varepsilon}(t, x) := \sum_{\lambda \in \mathcal{I}_0} \mathbf{u}_h^0(t, x) + \sum_{l=0}^{L-1} \sum_{\lambda \in \mathcal{D}_l(t)} \mathbf{d}_h^l(t, x), \quad x \in \Omega \text{ and } t \in (0, T], \tag{4.6}
\]

where evolution equations for the local contributions of the complements are partially neglected. Note that the significance of local contributions \( \mathbf{d}_\lambda^i \) may change in time. Thus, the index set \( \mathcal{D}_\varepsilon \) is not constant in time. In particular, when choosing (3.78) for the local norm, how significance is evolved in time is linked to the evolution of the local \( L_2 \)-norm of the local contributions \( \| \mathbf{d}_\lambda^i \|_{L_2(V_\lambda)} \). Based on the evolution equations for the local contributions (4.4) we derive an evolution equation for their local \( L_2 \)-norm.
Proposition 4.2  Let \( t_n \neq t_{n+1} \in (0,T) \), then, there holds for \( 1 \leq i \leq m \) and \( \lambda \in \mathcal{I}_\ell \), \( 0 \leq \ell \leq L-1 \):

\[
\left\| (d^\lambda_i(t_{n+1},\cdot))_i \right\|_{L^2(\mathcal{V}_\lambda)}^2 = \left\| (d^\lambda_i(t_n,\cdot))_i \right\|_{L^2(\mathcal{V}_\lambda)}^2 + 2 \int_{t_n}^{t_{n+1}} \left( \Delta (u_L,(d^\lambda_i(t,\cdot))_i) \right)_i dt.
\]  

(4.7)

Proof: For \( 1 \leq i \leq m \) we conclude from (4.4) with \( w = (d^\lambda_i(t,\cdot))_i \), that

\[
\frac{d}{dt}\left( (d^\lambda_i(t,\cdot))_i \right)_{L^2(\mathcal{V}_\lambda)} = 2 \left( \frac{\partial (d^\lambda_i(t,\cdot))_i}{\partial t},(d^\lambda_i(t,\cdot))_i \right)_{\mathcal{V}_\lambda} = 2 \left( \Delta (u_L,(d^\lambda_i(t,\cdot))_i) \right)_i.
\]

Integration over \((t_n,t_{n+1})\) leads to (4.7). \( \square \)

4.2 Significant contributions in the fully-discrete DG solution

As a next step for the derivation of an adaptive scheme, we migrate the idea of only considering significant contributions to the fully-discrete scheme (2.37). At each time step \( t_n \) we consider only significant contributions in the local multi-scale decomposition (3.30) of the solution of the fully-discrete scheme \( u^n_L \), i.e.,

\[
u^n_{L,x}(x) = \sum_{\lambda \in \mathcal{I}_0} u^{0,n}_{\lambda}(x) + \sum_{\ell=0}^{L-1} \sum_{\lambda \in \mathcal{I}_\ell \cap \mathcal{D}_\ell} d^\ell_{\lambda}(x), \quad x \in \Omega,
\]  

(4.8)

where the associated index set corresponding to significant local contributions \( \mathcal{D}_\ell = \mathcal{D}_\ell(t_n) \) approximates the index set of local contributions in the semi-discrete DG formulation. According to (3.45) and (3.46) this sparsified representation corresponds to a projection of \( u^n_L \) to an adaptive grid.

By means of (3.60) we can expand each component of the local contributions in (4.8) at time step \( t_n \) in the bases (3.58), i.e.,

\[
u^{0,n}_{\lambda}(x) = (\Phi_{\lambda}(x))^T U^n_{\lambda}, \quad \lambda \in \mathcal{I}_0,
\]  

(4.9)

\[
u^\ell_{\lambda}(x) = (\Phi_{\lambda}(x))^T D^n_{\lambda}, \quad \lambda \in \mathcal{I}_\ell \text{ and } 0 \leq \ell \leq L-1,
\]  

(4.10)

where coefficients of different components are comprised altogether in matrices \( U^n_{\lambda} \in \mathbb{R}^{p_{\lambda} \times m} \) and \( D^n_{\lambda} \in \mathbb{R}^{p_{\lambda} \times m} \), respectively.

Similarly to previous works [83, 108, 156], we derive evolution equations for significant contributions in (4.8) under the temporal evolution using an explicit Euler step. For ease of notation, we define a matrix-valued evolution operator \( \mathcal{L}(u_L,v) \) by

\[
(\mathcal{L}(u_L,v))_{i,j} := (\mathcal{L}(u_L,(v_i))_j,
\]  

(4.11)
where \( \mathbf{v} \) denotes a vector of arbitrary length. In particular, we use \( \Phi_\lambda \) and \( \Psi_\lambda \) for \( \mathbf{v} \) in order to write the (discrete) evolution equations for \( U^n_\lambda, D^n_\lambda \) in a compact way.

Next, we derive evolution equations for the local contributions of the complements based on an explicit Euler discretization in the reference scheme. According to (2.36) and (2.37) with \( s = 1 \), a time step of the reference scheme using an explicit Euler time discretization can be locally written as

\[
U^{n+1}_\lambda = U^n_\lambda + \Delta t \mathcal{L}(u_L, \Phi_\lambda), \quad \lambda \in \mathcal{I}_L. \tag{4.12}
\]

**Proposition 4.3 (Evolution of the multi-scale coefficients)** If the temporal evolution of the reference scheme \( U^K_\lambda \) is performed with the explicit Euler step (4.12), the following evolution equations for the coefficients (4.9) and (4.10) hold true

\[
U^{n+1}_\lambda = U^n_\lambda + \Delta t \mathcal{L}(u_L, \Phi_\lambda), \quad \lambda \in \mathcal{I}_{\ell}, 0 \leq \ell \leq L - 1, \tag{4.13}
\]

\[
D^{n+1}_\lambda = D^n_\lambda + \Delta t \mathcal{L}(u_L, \Psi_\lambda), \quad \lambda \in \mathcal{I}_{\ell}, 0 \leq \ell \leq L - 1. \tag{4.14}
\]

**Proof:** First, we prove (4.13) and (4.14) for arbitrary \( \lambda \in \mathcal{I}_{L-1} \). For that purpose, we rewrite (4.12) as

\[
U^{n+1}_\mu = U^n_\mu + \Delta t \mathcal{L}(u_L, \Phi_\mu), \quad \mu \in \mathcal{M}_{\lambda} \subset \mathcal{I}_L. \tag{4.15}
\]

Next, we apply (3.59) to comprise all coefficients corresponding to subcells \( \mu, ..., \mu_{|\mathcal{M}_{\lambda}|} \) of \( V_\lambda \) in one matrix \( U^{n+1}_{\mathcal{M}_{\lambda}} \), i.e.,

\[
U^{n+1}_{\mathcal{M}_{\lambda}} := \left( (U^{n+1}_{\mu_1})^T, ..., (U^{n+1}_{\mu_{|\mathcal{M}_{\lambda}|}})^T \right)^T.
\]

Thereby, we can rewrite (4.15) using (4.11) as

\[
U^{n+1}_{\mathcal{M}_{\lambda}} = U^n_{\mathcal{M}_{\lambda}} + \Delta t \mathcal{L}(u_L, \Phi_{\mathcal{M}_{\lambda}}). \tag{4.16}
\]

Next, by applying (3.67) and (3.68) column-wise, we conclude

\[
\tilde{U}^{\tilde{n}}_\lambda = M_{\lambda,0} U^{n+1}_{\mathcal{M}_{\lambda}}, \quad \tilde{D}^{\tilde{n}}_\lambda = M_{\lambda,1} U^{n+1}_{\mathcal{M}_{\lambda}}, \quad \tilde{n} \in \{n, n + 1\}. \tag{4.17}
\]

Then, we multiply (4.16) from the left with \( M_{\lambda,0} \) and \( M_{\lambda,1} \), respectively and deduce by (4.17) that

\[
\tilde{U}^{\tilde{n}+1}_\lambda = \tilde{U}^{\tilde{n}}_\lambda + \Delta t M_{\lambda,0} \mathcal{L}(u_L, \Phi_{\mathcal{M}_{\lambda}}),
\]

\[
\tilde{D}^{\tilde{n}+1}_\lambda = \tilde{D}^{\tilde{n}}_\lambda + \Delta t M_{\lambda,1} \mathcal{L}(u_L, \Phi_{\mathcal{M}_{\lambda}}).
\]

Since \( \mathcal{L} \) is linear in the second argument, we conclude by (4.11) and (3.63) that

\[
M_{\lambda,0} \mathcal{L}(u_L, \Phi_{\mathcal{M}_{\lambda}}) = \mathcal{L}(u_L, \Phi_\lambda),
\]

\[
M_{\lambda,1} \mathcal{L}(u_L, \Phi_{\mathcal{M}_{\lambda}}) = \mathcal{L}(u_L, \Psi_\lambda).
\]
Eventually, (4.13) and (4.14) hold true for arbitrary \( \lambda \in \mathcal{I}_{L-1} \). Finally, by proceeding inductively from \( \ell = L - 1 \) to \( \ell = 0 \), we deduce that (4.13) and (4.14) hold true for cells on all levels, i.e., \( \lambda \in \mathcal{I}_\ell, \ 0 \leq \ell \leq L - 1 \).

By means of Proposition 4.3, we derive a discrete version of Proposition 4.2 for the discrete time evolution using an explicit Euler time step.

**Conclusion 4.1 (Discrete evolution of significance)** If the temporal evolution of the reference scheme \( u_L^\ell \) is performed with an explicit Euler step, then there holds

\[
\| (d_{n+1}^\lambda)_{-i} \|^2_{L_2(V_{\lambda})} = \| (d_{n}^\lambda)_{-i} \|^2_{L_2(V_{\lambda})} + 2\Delta t \hat{L}(u_L, (d_{n}^\lambda)_{-i}) + (\Delta t)^2 \| \mathcal{L}(u_L, \Psi_{\lambda})_{-i} \|^2_2
\]

and

\[
\| (d_{n+1}^\lambda)_{-i} \|^2_{L_2(V_{\lambda})} = \| (d_{n}^\lambda)_{-i} \|^2_{L_2(V_{\lambda})} + 2\Delta t \hat{L}(u_L, (d_{n}^\lambda)_{-i}) + (\Delta t)^2 \| \mathcal{L}(u_L, (1 - (M_{\lambda,0})^T M_{\lambda,0}) \Phi_{M_{\lambda}})_{-i} \|^2_2.
\]

Here, we denote by \( \mathcal{L}(u_L, \Psi_{\lambda})_{-i} \) the \( i \)-th column of \( \mathcal{L}(u_L, \cdot) \).

**Proof:** First, we note that according to (4.10) the \( i \)-th component of the function \( d_{n+1}^\lambda \) can be written as

\[
(d_{n+1}^\lambda)_{-i} = (D_{n}^\lambda)_{-i} \cdot \Psi_{\lambda}.
\]

Thus, due to Parseval’s identity, we can compute \( \| (d_{n+1}^\lambda)_{-i} \|^2_{L_2(V_{\lambda})} \) by

\[
\| (d_{n+1}^\lambda)_{-i} \|^2_{L_2(V_{\lambda})} = \| (D_{n}^\lambda)_{-i} \|^2_{L_2(V_{\lambda})} = \left\| (D_{n}^\lambda)_{-i} \right\|_2^2.
\]

Then, by plugging in the evolution equation (4.14) we deduce

\[
\| (d_{n+1}^\lambda)_{-i} \|^2_{L_2(V_{\lambda})} = \| (D_{n}^\lambda)_{-i} \|^2_{L_2(V_{\lambda})} + 2\Delta t \left\| \mathcal{L}(u_L, \Psi_{\lambda})_{-i} \right\|_2^2 + (\Delta t)^2 \| \mathcal{L}(u_L, \Psi_{\lambda})_{-i} \|^2_2.
\]

Next, we conclude by (4.20), (4.11) and the linearity of \( \hat{L} \) in the second argument that

\[
\left\| \mathcal{L}(u_L, \Psi_{\lambda})_{-i} \right\|_2 = \hat{L}(u_L, (d_{n}^\lambda)_{-i}).
\]

Hence, (4.18) holds true. Due to the orthogonality of \( M_{\lambda} \) there holds

\[
\| \mathcal{L}(u_L, \Psi_{\lambda})_{-i} \|^2_2 = \| M_{\lambda}^T \left( \mathcal{L}(u_L, \Psi_{\lambda})_{-i} \right) \|^2_2 = \| M_{\lambda,1}^T \mathcal{L}(u_L, \Psi_{\lambda})_{-i} \|^2_2.
\]

Using the linearity of \( \hat{L} \) in the second argument, we conclude

\[
\| M_{\lambda,1}^T \mathcal{L}(u_L, \Psi_{\lambda})_{-i} \|^2_2 = \| \mathcal{L}(u_L, (M_{\lambda,1})^T \Psi_{\lambda})_{-i} \|^2_2.
\]

By (3.63), we deduce

\[
\| \mathcal{L}(u_L, (M_{\lambda,1})^T \Psi_{\lambda})_{-i} \|^2_2 = \| \mathcal{L}(u_L, (M_{\lambda,1})^T (M_{\lambda,1}) \Phi_{M_{\lambda}})_{-i} \|^2_2.
\]

Finally, we conclude (4.19) using (3.66). □
4.3 Prediction of significant contributions

To determine which local contributions become significant in the next time step, in principle the solution of the reference scheme $\mathbf{u}_t^{n+1}$ is required. However, the computation of the reference scheme has to be avoided. Thus, we have to predict the index set $\mathcal{D}_\varepsilon^{n+1}$ of significant contributions for the next time step $t_{n+1}$ by an index set $\mathcal{D}_\varepsilon^n$. The prediction of significant contributions at the new time step $t_{n+1}$ can only be based on available information corresponding to the old time step $t_n$, i.e., the set $\mathcal{D}_\varepsilon^n$. In order to ensure the reliability of the prediction, we have to avoid that future significant contributions are lost. For that purpose we want to ensure

$$\mathcal{D}_\varepsilon^n \cup \mathcal{D}_\varepsilon^{n+1} \subset \mathcal{D}^{n+1},$$

where $\mathcal{D}^{n+1}$ is as small as possible. Otherwise the prediction results in excessive grid refinement causing unnecessary computational overhead. Thus, the ‘overprediction’ has significant influence on the efficiency of the adaptive scheme. For scalar (hyperbolic) conservation laws in one spatial dimension a prediction strategy has been proposed in [108, 156]. This strategy is intertwined with the limiting process in order to ensure the reliability condition (4.21). For the proof of reliability of this strategy properties of the reference scheme that are only available for the scalar case in one spatial dimension are essential and therefore the extension to the multidimensional case is not trivial. Thus, we modify the idea of Harten’s heuristic strategy [98] and define $\mathcal{D}^{n+1}$ as the smallest superset of $\mathcal{D}_\varepsilon^n$ fulfilling the following constraints:

(i) significant contributions may remain significant within one time step, i.e.,

$$\mathcal{D}_\varepsilon^n \subset \mathcal{D}^{n+1},$$

(ii) contributions in a local neighborhood of a significant contribution may also become significant within one time step, i.e.,

$$\lambda \in \mathcal{D}_\varepsilon^n \Rightarrow \{\hat{\lambda} : V_{\hat{\lambda}} \text{ is neighbor of } V_\lambda\} \subset \mathcal{D}^{n+1},$$

(iii) new discontinuities may develop causing significant contributions on higher refinement levels, i.e.,

$$\max_{1 \leq i \leq m} \| \left( \mathbf{d}^i \right)_\lambda \|_\lambda > 2^M \varepsilon_{\lambda, L} \Rightarrow \mathcal{M}_\lambda \subset \mathcal{D}^{n+1},$$

where $M = p + 1$,

(iv) $\mathcal{D}^{n+1}$ is a tree, i.e.,

$$\mu \in \mathcal{D}^{n+1} \Rightarrow \lambda \in \mathcal{D}^{n+1} \forall \lambda \text{ with } V_\mu \notin V_\lambda.$$
This type of strategy has been originally developed for finite volume schemes. Although the reliability of this heuristic strategy has never been proven to hold, it turned out that it gives satisfactory results in the context of finite volume schemes \([29, 133, 134]\) as well as for DG schemes \([38, 80, 81, 83]\).

### 4.4 The fully adaptive DG scheme

By the prediction set \(\mathcal{D}^n\) an adaptive grid characterized by the index set \(\mathcal{I}^n\) is defined at each time step \(t_n\) according to (3.46). Associated to this locally refined grid there exists a discretization space \(S^n \subset S_L\). Analogously, we denote by \(\mathcal{I}^n\) the index set characterizing the adaptive grid defined by \(\mathcal{D}^n\) at time step \(t_n\). Furthermore, we denote by \(S^n \subset S^n \subset S_L\) the associated discretization space. Thus, the adaptive fully-discrete DG solution \(u_{\ell,\varepsilon}^n \in (S^n)^m\) can be written equivalently in two different ways, i.e.,

\[
 u_{\ell,\varepsilon}^n = \sum_{\lambda \in 0} u_{\lambda,\varepsilon}^{n} + \sum_{\ell=0}^{L-1} \sum_{\lambda \in 0} d_{\lambda,\varepsilon}^{n} = \sum_{\ell=0}^{L} \sum_{\lambda \in 0} u_{\lambda,\varepsilon}^{n}. \tag{4.26}
\]

A time step of the adaptive scheme consists of three parts:

First, before computing the time evolution the grid is refined via prediction of significant contributions using the multi-scale representation. Consequently, \(u_{\ell,\varepsilon}^n\) is represented in the basis of the richer space \(S^{n+1} \supset S^n\) corresponding to the refined grid \(\mathcal{I}^{n+1}\) of the next time step \(t_{n+1}\), i.e.,

\[
 u_{\ell,\varepsilon}^n = \sum_{\ell=0}^{L} \sum_{\lambda \in 0} u_{\lambda,\varepsilon}^{n}. \tag{4.27}
\]

Then, the evolution of the DG scheme with a RK scheme including local projection limiting is computed using the single-scale representation (4.27). The outcome of the evolution is denoted by \(\tilde{u}_{\ell,\varepsilon}^{n+1}\).

After the evolution, significant contributions might turn out to be non-significant, i.e., \(\tilde{u}_{\ell,\varepsilon}^{n+1}\) might contain non-significant contributions. Thus, we apply the thresholding procedure (3.40) using the rescaled definition of significance in (4.5) in order to compute

\[
 u_{\ell,\varepsilon}^{n+1} = \sum_{\ell=0}^{L} \sum_{\lambda \in 0} u_{\lambda,\varepsilon}^{n}. \tag{4.28}
\]

corresponding to a coarser grid.

In Algorithm 4.1 we summarize the procedure for one time step of the adaptive scheme. Note, when performing multi-scale transformations and the thresholding procedure for an already sparsified function, only local contributions \(u_{\lambda}^{t}\) and \(d_{\lambda}^{t}\) have to be considered which are present in the sparsified function. For that purpose, the corresponding index sets \(\mathcal{I}_{\ell}\) are replaced by \(\mathcal{I}_{\ell} \cap \mathcal{I}_{\varepsilon}^{t}\) in Algorithms 3.2 and 3.8, respectively.
Algorithm 4.1 Time step of the adaptive scheme

Input: \( u^n_{L,\varepsilon} \).

(i) **Grid refinement:**

   (a) apply locally multi-scale transformations (3.79) to determine the multi-scale decomposition of \( u^n_{L,\varepsilon} \) using Algorithm 3.2 recursively,

   (b) compute \( D^{n+1} \) from \( D^n \) by performing the prediction strategy according to (4.22), (4.23), (4.24) and (4.25),

   (c) determine the adaptive grid \( I^{n+1} \) from \( D^{n+1} \) and compute the single-scale coefficients in (4.27) by using Algorithm 3.9.

(ii) **Time evolution:**

    perform Runge-Kutta time evolution on the single-scale coefficients using (2.37) in order to compute \( \tilde{u}^{n+1}_{L,\varepsilon} \). Thereby, the limiter described in Algorithm 2.1 is applied after each stage to all cells of the adaptive grid on the finest level, i.e., \( \lambda \in I^{n+1} \cap I_L \).

(iii) **Grid coarsening:**

    (a) apply locally the multi-scale transformation (3.79) to determine the multi-scale decomposition of \( \tilde{u}^{n+1}_{L,\varepsilon} \) using Algorithm 3.2 recursively,

    (b) compute \( D^{n+1}_L \) from \( D^{n+1} \) by performing thresholding according to (4.5) and (3.40) using Algorithm 3.8,

    (c) determine the adaptive grid \( I^{n+1}_L \) from \( D^{n+1}_L \) and compute the single-scale coefficients in (4.28) by using Algorithm 3.9.

Output: \( u^{n+1}_{L,\varepsilon} \).
4.5 Choice of the local threshold value

In order to implement the adaptive scheme, we have to specify the local threshold values $\varepsilon_{\lambda,L}$. The efficiency of the adaptive scheme is essentially influenced by this choice: if $\varepsilon_{\lambda,L}$ is (locally) chosen too large, we obtain indeed a very coarse mesh resulting in low computational cost, but the thresholding error is dominating and spoils the accuracy of the solution. In contrast, if $\varepsilon_{\lambda,L}$ is chosen too small, the grid contains too many cells and efficiency is lost. For the choice of $\varepsilon_{\lambda,L}$ we have some flexibility as long as the constraint (3.38) is fulfilled to ensure uniform boundedness of the threshold error. We note that if there exist constants $a > 1$ and $C > 0$ such that for $L \in \mathbb{N}$ there holds

$$
\varepsilon_{\lambda,L} \leq C a^{\ell-L}, \quad \lambda \in \mathcal{I}_\ell, \quad 0 \leq \ell \leq L - 1,
$$

then (3.38) is satisfied because $\sum_{\ell=0}^{L-1} a^{\ell-L} \leq 1/(a-1)$. In the following we present a strategy for choosing $\varepsilon_{\lambda,L}$ for uniform and non-uniform hierarchies, respectively. For these strategies we verify (4.29).

4.5.1 Uniform dyadic hierarchies

For the choice of a suitable local threshold we first focus on the particular case of a uniform dyadic hierarchy. Here, introducing a threshold parameter $\varepsilon_L > 0$ as used in previous works, cf. [80, 81, 83, 108], results in a straightforward choice

$$
\varepsilon_{\lambda,L} = \frac{h_L}{h_\ell} \varepsilon_L, \quad \lambda \in \mathcal{I}_\ell,
$$

where $h_\ell$ and $h_L$ denote the (uniform) diameters of the cells on level $\ell$ and $L$, respectively. In order to derive a strategy for the choice of the parameter $\varepsilon_L$, we consider the error of the adaptive scheme at the final time $T = T_N$ resulting from $N$ time steps. This error can be estimated by the sum of the discretization error (error of the reference scheme) and the perturbation error (difference between reference scheme and adaptive scheme), i.e.,

$$
\|u_{L,\varepsilon}^N - u(t_N, \cdot)\| \leq \|u_{L,\varepsilon}^N - u_L^N\| + \|u_L^N - u(t_N, \cdot)\|,
$$

where for the norm $\| \cdot \|$ usually the $L_1$-norm is considered. In order to maintain the accuracy of the reference scheme, the perturbation error should not be dominant. Thus, we demand that the perturbation error is in the same order of magnitude as the discretization error of the reference scheme, i.e.,

$$
\|u_{L,\varepsilon}^N - u_L^N\| \approx \|u_L^N - u(t_N, \cdot)\|.
$$

For reasons of efficiency a computation of the reference scheme on the fully refined grid has to be avoided, otherwise the use of the adaptive scheme would be redundant. Thus,
4.5. CHOICE OF THE LOCAL THRESHOLD VALUE

we have to find a strategy for the choice of $\varepsilon_L$ avoiding the explicit use of the solution of the reference scheme.

For that purpose, we incorporate some 'knowledge' of the reference scheme. More precisely, we consider the order of convergence $\beta > 0$ of the reference scheme, i.e.,

$$
\left\| u_L^N - u(t_N, \cdot) \right\| = O(h_L^{\beta}),
$$

where $h_L$ is the (uniform) diameter of the cells on level $L$. The order $\beta$ can be identified using an a priori estimate or, alternatively, estimated from numerical computations of the reference scheme for well-known benchmark problems. The threshold $\varepsilon_L$ has to be chosen such that the perturbation error is decaying with the same order, i.e.,

$$
\left\| u_h^N,\varepsilon - u_h^N \right\| = O(h_L^{\beta}).
$$

To this end, we consider the approach $\varepsilon_L = C_{\text{thr}} h_L^\gamma$ with a constant $C_{\text{thr}}$ and an exponent $\gamma$ to be determined, i.e., the local thresholds are chosen as

$$
\varepsilon_{\lambda,L} = C_{\text{thr}} \frac{h_L^{1+\gamma}}{h_\ell}, \quad \lambda \in \mathcal{I}_\ell.
$$

We note that for a uniform dyadic hierarchy there holds $h_{\ell} = C 2^{-\ell}$ with a constant $C > 0$. Hence, we conclude that for all $\gamma > 0$ there holds $\varepsilon_{\lambda,L} = C_{\text{thr}} C_\gamma 2^{-(1+\gamma) \ell} \leq C_{\text{thr}} C_\gamma 2^{-\ell}$. Thus, by using (4.29) we conclude that (3.38) is fulfilled. In the following we present a strategy for choosing $\gamma$ and discuss the influence and the choice of $C_{\text{thr}}$.

The choice of $\gamma$

First we focus on the choice of the exponent $\gamma$. For scalar one-dimensional conservation laws it was proven in [108, 156] for a particular prediction strategy and limiter that the $L_1$-norm of the perturbation error in the mean can be estimated by a term proportional to $(\varepsilon_L + h_L^{\alpha})/\Delta t$, where $\alpha > 1$ is a constant coming from stability assumptions on the reference scheme and $\Delta t$ the temporal step size. The division by $\Delta t$ in this estimate accounts for the fact that we might introduce an error of order $\varepsilon_L$ in each time step and in the worst case this error accumulates over all time steps. Due to the CFL condition of the explicit time-stepping, $\Delta t$ is proportional to $h_L$ and therefore $N \sim T/h_L$. Thus, we may choose an a priori threshold parameter by

$$
\gamma_{\text{a priori}} = \beta + 1
$$

for $\alpha \geq \beta + 1$, where the $\beta + 1$ compensates the $h_L$ in the denominator. In several numerical examples it turned out that using this strategy often results in over-refined grids, cf. [81]. Since it is based on a priori error estimates and the assumption that in each time step
an error of order $\varepsilon_L$ is made the strategy is too pessimistic. For that reason a heuristic strategy was proposed in [81], where the power of $h_L$ is reduced by one, i.e.,

$$\gamma_{\text{heuristic}} = \beta.$$  \hfill (4.34)

This strategy has not yet been verified analytically. However, a numerical comparison of both strategies in [81] led to the conclusion that for test cases where discontinuities are present, i.e., $\beta \leq 1$, the heuristic strategy (4.34) gives satisfactory results meaning that the discretization error of the adaptive computation is asymptotically in the same order of magnitude as the discretization error of the reference scheme. In contrast, the a priori strategy (4.33) seems to the preferable choice for smooth test cases. If the solution is smooth, the reference scheme is expected to converge with $p$th order accuracy, whereas the reference scheme is at most first order accurate if discontinuities such as shocks or contact discontinuities are present. Thus, we suggest the following strategy for the choice of $\gamma$:

- if the solution is expected to be non-smooth, choose $\gamma = 1$,
- if the solution is expected to be smooth, choose $\gamma = p + 1$,

The choice of $C_{\text{thr}}$

First of all, we note that the choice of $C_{\text{thr}}$ does not influence the asymptotic behavior of the adaptive scheme: since $\varepsilon_L$ decays with $h_L^2$ for $L \rightarrow \infty$, the convergence behavior of the reference scheme is maintained. Nevertheless, the efficiency of the adaptive scheme is influenced by $C_{\text{thr}}$ because the thresholding procedure acts as a filter: depending on the choice of $\varepsilon_L$ very small features are not resolved. For instance, if we consider a discontinuity of strength $\delta \ll 1$, e.g.,

$$u(x) := \begin{cases} 1 & \text{if } x_1 < 1, \\ 1 - \delta & \text{else}, \end{cases}$$

then, this discontinuity will be resolved in the adaptive scheme only if the threshold parameter $\varepsilon_L$ is at least in the same order of magnitude as $\delta$. With increasing maximum refinement level the threshold value will become sufficiently small in order to resolve this small feature. If we are interested in resolving the grid up the finest refinement level near such a small feature even for a coarser maximum refinement level, we have to adjust $C_{\text{thr}}$ in order to ensure a sufficiently small $\varepsilon_L$. In numerous computations it turned out that $C_{\text{thr}} = \delta$ is a good choice if refinement of the weak discontinuity is desired. However, in most cases $C_{\text{thr}} = 1$ is a suitable choice.
4.5.2 Non-uniform hierarchies

In this section, we extend the above derived strategy for choosing the local threshold values to non-uniform hierarchies. The key question is how to interpret \( h_\ell \) and \( h_L \) in (4.32) for a non-uniform hierarchy. When considering a non-uniform hierarchy the accuracy of the reference scheme locally varies. Since the main goal of our adaptive scheme is to maintain the accuracy of the reference scheme, we consider the local accuracy when choosing the local threshold \( \varepsilon_{\lambda,L} \). To this end, we propose a localized strategy, where \( h_L \) as well as \( h_\ell \) in (4.32) are substituted with local equivalents. Thus, we define

\[
\varepsilon_{\lambda,L} = C_{\text{thr}} \frac{h_{L,\lambda}^{1+\gamma}}{h_{\lambda}},
\]

where \( \gamma > 0 \) is chosen as discussed above. The local grid ‘sizes’ \( h_{L,\lambda} \) and \( h_{\lambda} \) are defined as

\[
h_{L,\lambda} := \min_{\mu \in \mathcal{L}, \lambda \in \mathcal{V}_\lambda} |V_\mu|^{1/d}, \quad h_{\lambda} := |V_{\lambda}|^{1/d}.
\]

To verify that this choice fulfills (3.38), we first conclude from (3.6) that for each \( V_\lambda \) in the hierarchy there exists only a finite number of sub-cells \( V_\mu \) for which \( |\mathcal{M}_{\mu}| = 1 \), i.e., cells that are not refined. We denote the maximum of these numbers by \( N_{\text{max}} \). Next, we observe that if \( |\mathcal{M}_{\lambda}| \geq 2 \) then there holds \( \min_{\mu \in \mathcal{M}_{\lambda}} |V_\mu| \leq |V_{\lambda}|/2 \). Combining these two observations, we conclude that for all \( L \in \mathbb{N} \) there holds

\[
\varepsilon_{\lambda,L} = C_{\text{thr}} \frac{h_{L,\lambda}^{1+\gamma}}{h_{\lambda}} \leq C_{\text{thr}} (2^{1/d})^{L-N_{\text{max}}} h_{L,\lambda}^{1+\gamma}, \quad \lambda \in \mathcal{I}_{\ell}, \quad 0 \leq \ell \leq L-1.
\]

Due to the nestedness of the hierarchy (3.4) this can be estimated by

\[
\varepsilon_{\lambda,L} \leq C_{\text{thr}} 2^{N_{\text{max}}/d} \max_{\lambda \in \mathcal{I}_0} h_{\lambda}^{1+\gamma} (2^{1/d})^{L-\ell},
\]

and, thus, by using (4.29) we conclude that equation (3.38) holds true.

Note that this strategy is consistent with the canonical strategy (4.32) for a uniform-dyadic hierarchy. In the presented strategy, we have chosen the \( d \)th root of the volume to characterize the ‘size’ of the cells. This is not the only possible choice. Here, we have additional flexibilities. Alternatively, one might consider the diameter to quantify the ‘size’ of the cell. This is admissible whenever for the resulting thresholds (3.38) holds. For instance, this holds true if the diameter decays in an exponential rate similarly to the \( d \)th root of the volume as discussed above. Typically this is a constraint in the choice of the hierarchy.

4.6 Adaptation and limiting

The indicator in the limiting procedure should only detect cells, where the solution is non-smooth. According to (3.35) local contributions from the orthogonal complement
spaces are small whenever the underlying solution is locally smooth. Provided that we have chosen a suitable threshold value $\varepsilon_{\lambda,L}$, in smooth regions the grids are for that reason not refined up to finest level, whereas the grid is refined near discontinuities up to the finest refinement level. This cannot be proven rigorously but has been observed in numerous computations. Hence, the multiresolution analysis can be considered as an additional indicator for troubled cells in the limiting procedure, cf. [108, 172]. In [172] a MRA is used as a pure troubled cell indicator, where the magnitude of contributions from orthogonal complements characterizes troubled cells. There, no grid adaptation is considered.

Since the adaptation is based on a MRA, adaptation and limiting can be combined: we add the MRA as an additional indicator for the detection of troubled cells, i.e., the mimmod indicator and limiter described in Algorithm 2.1 are only applied to cells $\lambda$ on the finest refinement level, i.e., $\lambda \in \mathcal{I}^n \cap \mathcal{I}_L$. In the one-dimensional adaptive scheme proposed in [108] limiting and prediction were intertwined: whenever a cell is indicated for limiting, the prediction enforces a refinement of this cell up to the finest level. Thereby, the stability of the scheme considering the additional troubled cell indicator can be ensured. Since our prediction strategy is not intertwined with the limiting process, we cannot prove the reliability of the additional troubled cell indicator. However, it turned out that if the threshold value is chosen carefully this strategy gives satisfactory results in practice, cf. [38, 80, 83].

### 4.7 Adaptive initialization

To start the computation we need to generate the initial adaptive grid by means of the initial data $\mathbf{u}^0$. The simplest but not very efficient way for the initialization is to start on the highest refinement level $L$: we project the initial function $\mathbf{u}^0$ to $S^n_L$ and apply the thresholding procedure (4.5). This projection can be performed locally cell by cell, i.e.,

$$\mathbf{u}^0_\lambda := P_{S_L}(\mathbf{u}^0 1_{V_\lambda}), \quad \lambda \in \mathcal{I}_L. \tag{4.37}$$

When a discontinuity in the initial data is located exactly on a face of the grid it might not be refined up to the finest level. However, for accuracy and stability reasons it should be refined up to finest level. To ensure this, we (optionally) add a constraint to the thresholding procedure: if the maximum of the difference between the mean values of $\mathbf{u}^0_\lambda$ and the mean values of the neighboring cells is larger than $\varepsilon_{\lambda,L}$ the corresponding cell is kept, i.e.,

$$\max_{\lambda \in \mathcal{J}_{\lambda,\mathcal{I}_L}} \left| \frac{1}{V_\lambda} \int_{V_\lambda} \mathbf{u}^0_\lambda - \mathbf{u}^0_\mathcal{I} \right| > \varepsilon_{\lambda,L},$$

where the set $\mathcal{J}_{\lambda,\mathcal{I}} := \{ \tilde{\lambda} \in \mathcal{I} : V_{\tilde{\lambda}} \text{ is neighbor of } V_\lambda \}$ contains the neighboring cells belonging to the index set $\mathcal{I}$. By this strategy we are able to guarantee that all significant
features of the initial data are captured by the adaptivity. However, this approach leads to the computational complexity of the reference scheme for the initialization. This is not desirable since it is our goal to avoid the computational complexity of the reference scheme by grid adaptation. For that reason we consider an alternative approach based on a greedy-procedure, which has been originally proposed for the initialization in adaptive finite volume schemes in [133]. Here, we adjust this strategy to the framework of adaptive DG schemes; the resulting strategy is presented in Algorithm 4.2. The memory requirement of this strategy is in the same order of magnitude as for the resulting adaptive representation. Moreover, the computational cost is reduced significantly since an initial projection to the fully refined grid is avoided.

Algorithm 4.2 Bottom-up initialization

Input: $u_0$.

1: initialize $\mathcal{I}_0 = \mathcal{I}_0$,
2: project $u^0$ and $(S_0)^m$, i.e., $u^0_\lambda = P_{S_0}(u^0 1_\lambda)$, $\lambda \in \mathcal{I}_0$,
3: for $\ell = 0, \ldots, L - 1$ do
4:   for all $\lambda \in \mathcal{I}_{\ell}^0 \cap \mathcal{I}_\ell$ do
5:     for $\mu \in \mathcal{M}_\lambda$ compute $u^{\ell+1,0}_\mu = P_{S_{\ell+1}}(u^0 1_{V_\mu})$,
6:     thereby compute $d^{\ell,0}_\lambda \equiv P_{W_{\ell}}(\sum_{\mu \in \mathcal{M}_\lambda} u^{\ell+1,0}_\mu)$ using Algorithm 3.4 or rather 3.5.
7:   end for
8: for all $\lambda \in \mathcal{I}_{\ell}^0 \cap \mathcal{I}_\ell$ do
9:   if $d^{\ell,0}_\lambda$ is significant by means of (4.5) or $\max_{\mu \in \mathcal{M}_\lambda} \max_{\mu \in \mathcal{I}_{\ell+1}} |\overline{u}^0_\mu - \overline{u}^0_{\mu_\ell}| > \varepsilon_{\mu,\ell}$ then
10:      remove $\lambda$ from $\mathcal{I}_{\ell}^0$, insert $\mathcal{M}_\lambda$ into $\mathcal{I}_{\ell}^0$ and delete $d^{\ell,0}_\lambda$ and $u^{\ell,0}_\lambda$,
11:    else
12:      delete $d^{\ell,0}_\lambda$ and $u^{\ell+1,0}_\mu$ for all $\mu \in \mathcal{M}_\lambda$,
13: end if
14: end for
15: end for

Output: $u^0_{L,\ell} = \sum_{\ell=0}^L \sum_{\lambda \in \mathcal{I}_{\ell}^0 \cap \mathcal{I}_\ell} u^{\ell,0}_\lambda$.

This strategy has to be applied very carefully since it cannot be guaranteed that all significant contributions are captured. However, the initial data are often very simple and only consist of piece-wise constant data or piece-wise polynomial data. It turned out that for these cases this strategy works efficiently and reliably. Moreover, knowledge of the initial data can be incorporated into the bottom-up strategy to ensure that certain regions are refined.
4.8 Parallelization of the adaptive scheme

With increasing size of the considered problems computational cost and memory requirements of the RKDG scheme rise dramatically. In particular, three-dimensional problems with need for high resolution are computationally challenging. Although the computational cost can be significantly reduced by local grid adaptation, this reduction might not be sufficient in order to obtain results in an affordable time. Therefore, a parallelization strategy to distribute the work load of the adaptive scheme on several work units, i.e., processors, is required. We aim at a distributed memory parallelization using MPI [93, 165].

Due to the locality of the evolution operator of the DG scheme, these schemes are well suited for parallelization, cf. [4, 11, 15, 105]. Since the adaptive grid in our scheme is dynamically changing in time, the distribution of the work load needs to be rebalanced during the computation. The quality of the load-balancing crucially influences the efficiency of the parallel scheme. The adaptive grids resulting from multi-scale based grid adaptation are unstructured since hanging nodes are present. Thus, the task of finding an efficient load-balancing strategy is not trivial. Here, we use space-filling curves [155, 193] for the distribution of the adaptive grid and the rebalancing of the workload. In this section we briefly summarize the basic ideas of distribution and load balancing of adaptive grids using space-filling curves. For a detailed description we refer to [31].

![Figure 4.1: First three iterates of Hilbert's space-filling curve in two spatial dimensions.](image)

The basic idea is to map the identifiers of the cells in the hierarchy to a one-dimensional line using space-filling curves in order to obtain a global enumeration of the cells in the adaptive grid. An example of a space-filling curve in two spatial dimensions is shown in Figure 4.1. Each cell identifier $\lambda$ is mapped to a point on the curve, the SFC index. This line is split into segments each containing approximately the same number of cells. The segments are defined by separators, which can be locally computed with little communication. Then, each segment defines the work load for one of the work units. In Figure 4.2 the distribution of an adaptive grid on two work units is exemplarily shown.
4.8. PARALLELIZATION OF THE ADAPTIVE SCHEME

The space-filling curve is constructed such that a cell identifier \( \lambda \) is mapped to the same SFC index as one of its subcells’ identifier \( \mu \in \mathcal{M}_\lambda \). Thereby, it is ensured that \( \lambda \) and its subcells \( \mu \in \mathcal{M}_\lambda \) are assigned to the same work unit in most cases. Hence, interprocessor communication is minimized while performing a parallel MRA and grid adaptation, respectively. Rebalancing of the adaptive grid is needed whenever the evolution of the adaptive scheme leads to non-balanced distribution, i.e., the number of cells per processor differs strongly. For that purpose new separators defining the new distribution have to be computed and cells have to be transferred to their new work unit if this has changed during the rebalancing process.

![Space filling curve and distribution](image)

Figure 4.2: Example of distribution of an adaptive grid using space filling curves on two processors. On the left the space filling curve is shown. On the right, the resulting distribution is shown: cells for processor one and two are highlighted in red and blue, respectively.

When parallelizing multi-level methods, the efficiency of the parallelization strategy is influenced by the parallel efficiency of the computational load related to the coarsest level. In our scheme, during the realization of the MRA, the solution is projected to coarser grids on descending level \( \ell \). In particular, a projection on the grid on level \( \ell = 0 \) has to be performed. If the grid on level 0 consists only of a few cells, the efficiency of the parallelization might be damaged, since these projections are realized only by a few work units, whereas many other work units are running idly. Thus, we suggest to consider a sufficiently large number of cells on level 0 in order to avoid this kind of problems, i.e., the grid on level 0 should contain at least as many cells as work units (cores) considered in the parallelization. The total number of cells in the adaptive grid is mostly triggered by the refinement on higher levels. Thus, increasing the number of cells on level \( \ell = 0 \)
does not increase the total number of cells in the adaptive grid significantly.

### 4.9 Numerical results

In [38, 80, 81, 83, 120] numerical results are presented for numerous test cases using the herein presented adaptive scheme. There, the classical approach (3.60) has been applied with the local norms (3.77) based on the explicit use of multiwavelets. Thus, the main focus of this section is on the validation of the newly developed wavelet-free approach (3.73) with the alternative choice for the local norms (3.78).

More precisely, we investigate whether the wavelet-free approach is an efficient alternative to the classical approach using multiwavelets. To this end, we perform adaptive computations for several well-known numerical test cases and compare the accuracy and efficiency of both approaches. Moreover, we explore the difference between the heuristic strategy (4.34) and the a priori strategy (4.33) for test cases where the exact (entropy) solution is known. For the classical approach this has been already done in [80, 81]. There the comparison verified for non-smooth solutions that the heuristic strategy is the right choice. The a priori strategy leads to too dense grids without significantly improving the accuracy. Here, we investigate if the conclusions are still valid for the wavelet-free approach.

The outline of the section is as follows: first, we consider several test cases with a uniform subdivision, since here we can compare both approaches. Then, we present numerical results for a test case where we apply a non-uniform subdivision of the cells in the hierarchy. In this situation it is mandatory to apply the wavelet-free approach in order to perform grid adaptation. Finally, we shortly discuss the findings from this comparison.

To measure the efficiency of the adaptive scheme we apply the following criterion: first of all we note that at each time step $t_n$ the efficiency of the adaptive scheme can be estimated by the number of cells in the adaptive grid characterized by the size of $\mathcal{T}^n$. Since this number changes from time step to time step, we consider the maximum number appearing during the computation. Thus, we estimate the efficiency of the adaptive scheme by comparing

$$
N_{\text{max}} := \max_{\text{all time steps } t_n} |\mathcal{T}^n|
$$

with the number of cells of the reference scheme. For the reference scheme $N_{\text{max}}$ coincides with $|\mathcal{L}_{\text{ref}}|$.

Simulation parameters, unless stated otherwise, are the following. We consider quadrilateral Cartesian grids with uniform cells on each level. The subdivision is dyadic, i.e., $|\mathcal{M}_l| = 2^l$. The computations have been performed using a third-order scheme, i.e., $p = 3$. For the time discretization an explicit third-order SSP-RK method with three stages and
4.9. NUMERICAL RESULTS

$C_{\text{CFL}} = 0.1$ is used. To compute the integrals in the scheme for a linear problem exactly, the quadrature rule must be capable of integrating polynomials of degree $2p - 2$ exactly. However, in the general non-linear case the integrand is not a polynomial but the product of a polynomial with a polynomial mapped with the non-linear flux function $F$. For these reasons, we compute the integrals in the DG scheme using a tensor product based Gaussian quadrature formula with $(p + 1)^d$ points. In order to avoid limiting in regions, where the solution is smooth, we consider a Shu constant of $M = 50$. In order to reduce the computational cost, the two-dimensional computations have been performed in parallel using the parallelization strategy presented in Section 4.8.

4.9.1 Two-dimensional inviscid Burgers’ equation

In the first instance, we consider a classical two-dimensional Riemann problem for the inviscid Burgers’ equation

$$\frac{\partial u}{\partial t} + \frac{1}{2} \nabla \cdot \left(u^2 \left(\begin{array}{c} 1 \\ 1 \end{array}\right)\right) = 0$$

on $\Omega = [0,1]^2$ that is subject to the initial data

$$u(0,x) = \begin{cases} 
-0.2 & \text{if } x_1 < 0.5 \text{ and } x_2 > 0.5 \\
-1 & \text{if } x_1 \geq 0.5 \text{ and } x_2 > 0.5 \\
0.5 & \text{if } x_1 < 0.5 \text{ and } x_2 \leq 0.5 \\
0.8 & \text{if } x_1 \geq 0.5 \text{ and } x_2 \leq 0.5
\end{cases} \, .$$

The exact entropy solution can be determined using characteristics, cf. [184]. Since the exact solution is explicitly known for this test case, it is a well-known benchmark test case for numerical schemes dealing with nonlinear conservation laws, cf. [43, 94]. In order to avoid an ill-posed problem caused by inadmissible boundary conditions, we pose the exact solution as boundary condition at the inflow boundaries using the characteristic boundary conditions (2.22). From the initial states a two-dimensional wave structure consisting of shocks and a rarefaction wave emerges, see Figure 4.3.

We utilize this test case for the validation of our adaptive strategy and a comparison of the strategies for the choice of the threshold value, i.e., the a priori strategy (4.33) and the heuristic strategy (4.34). Since the solution of this test contains discontinuities, we choose $\beta = 1$ in both strategies with $C_{\text{thr}} = 1$. Furthermore, we compare the classical approach using (3.77) with the wavelet-free approach using (3.78). For that purpose, we compute adaptive solutions with the classical and the wavelet-free approach. Both strategies for choosing $\varepsilon_L$ are considered. Computations have been performed up to $T = 0.5$ for different maximum refinement levels $1 \leq L \leq 10$ based on $10 \times 10$ cells on level 0. Thus, the grid of the reference scheme on level $L = 10$ consists of 104,857,600 cells. Exemplarily, we show the adaptive solution for $L = 6$ using the wavelet-free approach together with the heuristic
strategy in Figure 4.3. Here, we observe that the grid is only refined up to the finest level in regions where the solution is discontinuous, whereas the grid is coarse in regions where the solution is smooth.

Figure 4.3: Burgers’ equation: adaptive solution and grid at $T = 0.5$ using the wavelet-free approach with the heuristic strategy and $L = 6$. 

In order to evaluate the accuracy of the adaptive scheme, we compare the discretization errors of the adaptive schemes with the uniform results of the reference scheme. To this end, we compute uniform solutions with the reference scheme for $1 \leq L \leq 9$. A computation with $L = 10$ was unrealizable in an acceptable time due to the huge amount of degrees of freedom. In order to evaluate the accuracy of the different approaches, we compare the $L_1$-error of the adaptive scheme with the error of the reference scheme. For that purpose, the error is plotted over the maximum refinement level of the corresponding computation in Figure 4.4(a) and Figure 4.5(a) for the heuristic strategy and the a priori strategy, respectively. From these figures, it can be seen that for both the classical and the wavelet-free approach the accuracy of the reference scheme is maintained. The accuracy of all approaches and strategies is almost identical with the accuracy of the reference scheme.

Next, we analyze the efficiency of the different approaches and strategies, respectively. To this end, we focus on the maximum number of cells $N_{\text{max}}$ of the different computations. Thereby, we compare on the one hand both strategies for the choice of $\varepsilon_L$. On the other hand, we investigate if the wavelet-free approach is an efficient alternative to the classical approach using multiwavelets. In order to draw a quantitative conclusion, we plot the $L_1$-error over $N_{\text{max}}$ in Figure 4.4(b) and Figure 4.5(b) for the heuristic strategy and the a priori strategy, respectively. Thereby, we note that the heuristic strategy is more efficient
4.9. **Numerical Results**

![Graph](image1.png)

(a) accuracy

![Graph](image2.png)

(b) efficiency

Figure 4.4: Burgers’ equation: accuracy and efficiency of the adaptive scheme using the **heuristic** strategy for choosing the threshold value. Each symbol denotes the results from a computation with a different maximum refinement level.

![Graph](image3.png)

(a) accuracy

![Graph](image4.png)

(b) efficiency

Figure 4.5: Burgers’ equation: accuracy and efficiency of the adaptive scheme using the **a priori** strategy for choosing the threshold value. Each symbol denotes the results from a computation with a different maximum refinement level.
than the a priori strategy. This holds true for both the wavelet-free as well as the classical approach. In order to quantify the overhead of the a priori strategy, we consider the ratio of maximum cells in the grid of both strategies, i.e.,

\[
\text{ratio}_{\text{strategy}} := \frac{N_{\text{max}} \text{ using the a priori strategy}}{N_{\text{max}} \text{ using the heuristic strategy}}.
\]

In Figure 4.6(a) this ratio is plotted for the different maximum refinement levels for the classical as well as the wavelet-free approach. We find that the overhead of the a priori strategy is significant, i.e., the a priori strategy is too pessimistic. In fact, the overhead of the a priori strategy is getting worse with increasing maximum refinement level. Moreover, we note that there is a slight difference between the wavelet-free and the classical approach.

In order to quantify the difference between the classical and the wavelet-free approach, we consider in Figure 4.6(b) the ratio of \(N_{\text{max}}\) using the wavelet-free approach over \(N_{\text{max}}\) using the classical approach, i.e.,

\[
\text{ratio}_{\text{approach}} := \frac{N_{\text{max}} \text{ using the wavelet-free approach}}{N_{\text{max}} \text{ using the classical approach}}.
\]

We focus on this ratio for the a priori as well as the heuristic strategy. Thereby, we conclude that using the heuristic strategy results in an overhead of the wavelet-free approach of less than 10\%. For the a priori strategy the overhead is larger but less than 25\%. However, the ratios are for all computations less than 2. Thus, the general asymptotic behavior of the classical and wavelet-free approach is similar. This is again confirming the observation from above. The heuristic strategy is the better choice for this test case. Moreover, we conclude that the wavelet-free approach seems to be a reasonable alternative to the classical approach.

### 4.9.2 Compressible Euler equations

As a next step, we validate the adaptive strategy for a non-linear hyperbolic system of conservation laws, namely the compressible Euler equations, i.e.,

\[
\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p \mathbf{I} \\ \rho (E + p) \end{pmatrix} = 0, \tag{4.39}
\]

where \( \mathbf{u} = (\rho, \rho \mathbf{v}, \rho E)^t \). These equations express conservation of mass, momentum and total energy density. Here \( \rho, \mathbf{v}, E = \varepsilon + \frac{1}{2} \mathbf{v}^2 \) and \( \varepsilon \) denote the density, velocity, specific total energy and specific internal energy, respectively. The system is closed by the equation of state for the pressure

\[ p = \rho e (\gamma - 1) \]

for a thermally and calorically perfect gas with the ratio of specific heats \( \gamma = 1.4 \) for air at standard conditions [6].
4.9. NUMERICAL RESULTS

![Graphs showing wavelet-free vs. classical and heuristic vs. a priori strategies](image)

(a) heuristic vs. a priori strategy  
(b) wavelet-free vs. classical approach

Figure 4.6: Burgers’ equation: quantification of the efficiency.

**Sod’s shock tube**

At first, we focus on a one-dimensional Riemann problem for the compressible Euler equations in one spatial dimension. For that purpose, the computational domain is chosen as $\Omega = (-0.5, 0.5)$. Initially we consider two constant initial states separated by a membrane at $x = 0$, i.e.,

$$u(0, x) = \begin{cases} u_l, & x \leq 0, \\ u_r, & \text{else}. \end{cases}$$

Then, at time $t = 0$ the membrane breaks and emanating from the initial states a wave structure develops. Here, we consider a well-known example of such a Riemann problem: Sod’s shock tube problem [166] where the initial states are chosen as

$$u_l := (1, 0, 2.5)^t, \quad u_r := (0.125, 0, 0.25).$$

This configuration is frequently considered for the validation of numerical schemes, cf. [46, 172]. The wave structure consists of a rarefaction wave, a contact discontinuity and a shock wave. Thus, all relevant wave types of the compressible Euler equations are present in this test case. The exact solution is known and can be determined using an exact Riemann solver, cf. [124]. For that reason, this configuration is well-suited to investigate the quality of our adaptive strategy for the compressible Euler equations.

We consider a CFL number of 0.1 in the computations. On level 0 the grid consists of 10 cells. We perform computations up to $T = 0.25$ with different maximum refinement levels $1 \leq L \leq 10$ using the adaptive scheme. Again, we perform computations with the classical approach employing multiwavelets as well as the wavelet-free approach. Moreover, we compare the heuristic strategy with the a priori strategy for the choice of the
threshold value for both approaches. Thus, we perform four computations for each maximum refinement level \( L \). Since the test case contains discontinuities, we consider \( \beta = 1 \) and \( C_{\text{thr}} = 1 \) in the heuristic strategy and the a priori strategy for the choice of the threshold value, i.e., \( \varepsilon_L = h_L \) and \( \varepsilon_L = h^2_L \), respectively.

In Figure 4.7 the adaptive solution and the corresponding grid are shown exemplarily for the \( L = 9 \) computation using the wavelet-free approach and the heuristic strategy. Here, we observe that the grid is only refined up to the finest level at the shock and the contact discontinuity, whereas the grid at the rarefaction wave is only refined up to an intermediate refinement level. Thus, only very few cells in the adaptive grid are on the maximum refinement level \( L \).

![Figure 4.7: Sod’s shock tube: adaptive solution and grid for \( L = 9 \) at the final time \( T = 0.25 \). In the upper plot the solution is shown. In the bottom plot the refinement level of the cells in the adaptive grid is shown.](image)

In order to evaluate the temporal evolution of the grid adaptation, we consider in Figure 4.8 a space-time plot of the adaptive solution and the actual refinement level of the cells for the \( L = 9 \) computation using the wavelet-free approach together with the heuristic strategy for choosing \( \varepsilon_L \). The space-time plots confirm the findings of Figure 4.7: the dynamical grid adaptation clearly tracks the moving shock, the contact discontinuity as well as the kinks of the rarefaction wave with its refinement during the temporal evolution.

In order to verify whether the adaptive scheme using the different approaches and strategies is capable of maintaining the accuracy of the reference scheme, we again compare
the discretization errors of the adaptive schemes with the uniform results of the reference scheme. To this end, we compare the $L_1$-error in the density of the adaptive scheme with the reference scheme in Figure 4.9(a) for the heuristic strategy and 4.10(a) for the a priori strategy, respectively. It can be seen from these figures that using both approaches, the classical and the wavelet-free approach together with the heuristic as well as the a priori strategy, the accuracy of the reference scheme is maintained. Again, the accuracy is almost identical with the accuracy of the reference scheme.

Figure 4.8: Sod’s shock tube: space-time plot of the adaptive solution and grid: solution on the left; refinement level on the right.

Figure 4.9: Sod’s shock tube: accuracy and efficiency of the adaptive scheme using the *heuristic* strategy for choosing the threshold value. Each symbol denotes the results from a computation with a different maximum refinement level.

For the purpose of estimating the efficiency of the different approaches together with both strategies for choosing $\varepsilon_L$, we again compare $N_{\text{max}}$ in Figure 4.9(b) and 4.10(b), respectively. We note that the heuristic strategy is more efficient than the a priori strategy.
using the classical as well as the wavelet-free approach. In fact, the a priori strategy is too pessimistic in this case. In comparison to the inviscid Burgers’ equation this difference is significantly larger. However, we note that the difference between the classical and wavelet-free approach is comparably small using either the a priori or the heuristic strategy. Thus, the results of this test case again indicate that the wavelet-free approach is a suitable alternative to the classical approach using multiwavelets. Finally, we conclude that the adaptive strategy reduces the computational cost significantly while maintaining the accuracy of the reference scheme. In particular, this holds true independently of the approach for the basis of the complements.

Two-dimensional problems

In order to demonstrate that our adaptation is also capable of dealing with multi-dimensional compressible Euler equations, we consider two-dimensional test cases. Since we conclude from the previous test case that the a priori strategy leads to overrefined grids, we abstain from computing solutions using this strategy and consider only the heuristic strategy for the remaining test cases. Hence, the primary focus is on the comparison of the classical and the wavelet-free approach.

When computing numerical solutions of the compressible Euler equations one has to be aware of the fact that the classical concept of entropy solutions does not necessarily guarantee the well-posedness of the problem, cf. [42,65]. However, the compressible Euler equations are commonly used for the simulation of compressible fluid flows. Here, we consider well-known benchmark test cases in order to investigate whether our adaptation strategy is capable of resolving all relevant features while reducing the computational cost.

A well-known phenomenon when solving the compressible Euler equations in higher
4.9. NUMERICAL RESULTS

spatial dimensions is the occurrence of instabilities, such as Kevin-Helmholtz instabilities at shear layers or Rayleigh-Taylor instabilities, cf. [49, 81, 138, 150], respectively. These instabilities can also be observed in nature, cf. [75, 149, 159]. Thus, these instabilities are not only numerical artifacts. However, their appearance in the numerical approximation depends on the amount of numerical viscosity in the numerical scheme. If a numerical scheme introduces a lot of viscosity for stabilization, e.g. in low-order finite volume schemes, then the numerical solution does not exhibit any instabilities, cf. [131, 132, 178], whereas numerical solutions of high-order schemes such as DG schemes typically exhibit these structures, cf. [49, 81]. Hence, it can be understood as an indicator for the amount of numerical viscosity introduced by the numerical scheme.

Wave interaction in a Riemann problem

First, we consider a comparatively simple test case: a classical Riemann problem on the computational domain $\Omega = [0, 1] \times [0, 1]$. The initial data consist of four constant states. Their alignment and values are depicted in Figure 4.11. This test case has been proposed for the validation of a finite volume scheme in [131]. Later, it has been used for the validation of multi-scale based grid adaptation in the context of finite volume schemes in [133].

\[ u_1 = \begin{pmatrix} 0.125 \\ 0.125 \\ 0.375 \end{pmatrix}, \quad u_2 = \begin{pmatrix} 1 \\ 0 \\ 2.5 \end{pmatrix} \]

Figure 4.11: Two-dimensional wave interaction: initial configuration

From the four constant initial states complex wave structures emerge in the center of the domain. Away from these structures the solution exhibits a one-dimensional wave structure consisting of a rarefaction wave, a contact discontinuity and a shock wave similar to Sod’s shock tube.

We consider a grid consisting of $10 \times 10$ cells on level 0. We perform computations up to $T = 0.15$ with different maximum refinement levels $5 \leq L \leq 7$ using the adaptive scheme. Since the test case contains discontinuities, we consider the heuristic strategy with $\beta = 1$ and $C_{\text{thr}} = 1$ for the choice of the threshold value, i.e., $\varepsilon_L = h_L$.

In Figure 4.12 the density profile of the adaptive solution and the corresponding adaptive grid are shown for $L = 7$ using the wavelet-free approach. We observe that the wave structures emerging from the wave interaction show instabilities as discussed above. Furthermore, we note that the (one-dimensional) shock waves and contact discontinuities
away from the center are resolved up to the finest level, whereas the (one-dimensional) rarefaction wave is only refined up to an intermediate refinement level. This coincides with the findings for Sod’s shock tube problem. In the central area the wave structures are appropriately resolved. Moreover, we notice that the grid is only refined up to the finest level when the solution is locally discontinuous or contains complex wave structures. In regions with small fluctuations the grid is relatively coarse.

Figure 4.12: Two-dimensional wave interaction: adaptive solution (density) and grid for \( L = 7 \) using the wavelet-free approach.

In order to investigate the gain of increasing the resolution in the numerical approximation, i.e., increase the maximum refinement level, we focus on the central area of the computational domain and compare here the solutions and corresponding adaptive grids for different maximum refinement levels. For that purpose, the adaptive solution and the corresponding adaptive grids in this area are shown for \( 5 \leq L \leq 7 \) in Figure 4.13. Here, we note that with increasing resolution the instabilities in the solution appear more clearly. Moreover, with increasing maximum refinement level the structures such as discontinuities are resolved sharper. The grid adaptation captures these structures and refines these areas. However, in all of these computations the grid adaptation is capable of resolving the relevant structures of the solution. In previous works considering this test case [131, 133] no instabilities can be seen in the numerical results. This is related to the facts that (i)
a more diffusive second-order finite-volume scheme is used and (ii) the grid resolution is much coarser. That was to be expected, since using a coarse resolution with our method results in instability-free solutions as well.

![Wave interaction images](image)

Figure 4.13: Two-dimensional wave interaction: details of the adaptive solutions (density) and the corresponding grids for $5 \leq L \leq 6$ using the wavelet-free approach.

Since the exact solution is not known for this test case, we cannot estimate the efficiency in terms of the error with regard to the maximum numbers of cells $N_{\text{max}}$. However, we compare the maximum numbers of cells in the adaptive schemes using either the wavelet-free or the classical approach with the computational complexity of the reference scheme in Table 4.1.

From Table 4.1, we note that using the wavelet-free approach results in a slightly more refined grid than using the classical approach. Nevertheless, the reduction of the number of cells in the adaptive grid compared to the (uniform) reference scheme is substantial for both approaches. For instance, in the $L = 7$ computations the adaptive grid contains at most 7.60% of the cells of the reference grid using the wavelet-free and 6.54% using the classical approach, respectively.

Exemplarily, the same details of the adaptive grids using both the wavelet-free approach as well as the classical approach are shown in Figure 4.14. Here, additional refinement when using the wavelet-free approach can be noticed by slightly wider re-
finement regions around the wave structures and slightly more refinement in the smooth regions. Nevertheless, the difference between both grids is comparably small. Both adaptive schemes can clearly resolve the wave structures and avoid refinement in regions where the solution is smooth.

<table>
<thead>
<tr>
<th>Computation</th>
<th>$N_{\text{max}}$ (wavelet-free)</th>
<th>$N_{\text{max}}$ (classical)</th>
<th>$N_{\text{max}}$ (reference grid)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L = 4$</td>
<td>7,717 (30.14%)</td>
<td>6,484 (25.3%)</td>
<td>25,600</td>
</tr>
<tr>
<td>$L = 5$</td>
<td>20,128 (19.66%)</td>
<td>16,981 (16.58%)</td>
<td>102,400</td>
</tr>
<tr>
<td>$L = 6$</td>
<td>50,557 (12.34%)</td>
<td>43,888 (10.71%)</td>
<td>409,600</td>
</tr>
<tr>
<td>$L = 7$</td>
<td>124,447 (7.60%)</td>
<td>107,191 (6.54%)</td>
<td>1,638,400</td>
</tr>
</tbody>
</table>

Table 4.1: Two-dimensional wave interaction: comparison of the maximum number of cells in the adaptive scheme during the computation.

**Double Mach reflection**

Next, we consider a more challenging configuration: the reflection of a shock making a 60° angle with a reflecting wall. This test case has been introduced by Woodward and Colella in [178]. Although the exact solution is not known, it is a well-known benchmark test case for numerical schemes for compressible Euler equations, cf. [49,94,114,138,146,168,172]. Induced by the reflection of the shock wave complex wave structures develop.
4.9. NUMERICAL RESULTS

The computational domain is \( \Omega = [0, 4] \times [0, 1] \). The initial data consists of two constant states separated by a shock at \( x_1 = 1/6 + \tan(30^\circ) x_2 \), i.e.,

\[
\mathbf{u}(0, x) := \begin{cases} 
\mathbf{u}_l & \text{if } x_1 < 1/6 + \tan(30^\circ) x_2 , \\
\mathbf{u}_r & \text{else}
\end{cases}
\]

where the values for \( \mathbf{u}_l \) and \( \mathbf{u}_r \) are depicted in Figure 4.15. This flow is more challenging than the previous test case because the flow on the left-hand side of the separating shock is supersonic, whereas the flow on the right-hand side is subsonic. This change of the flow regime can cause problems in the numerical simulation.

At the lower boundary, i.e., \( x_2 = 0 \) we impose a reflecting wall between \( x_1 = 1/6 \) and \( x_1 = 4 \). Since the shock position in the free flow away from the wall is explicitly known, it is used to specify suitable boundary conditions for the flow field at all other boundaries, i.e.,

\[
\mathbf{u}_{\text{ext}}(t, x) := \begin{cases} 
(\rho^-, (\rho v_1)^-, -(\rho v_2)^-,(\rho E)^-)^t & \text{if } x_2 = 0 \text{ and } x_1 > 1/6 , \\
\mathbf{u}(0, (x_1 - 10/\cos(30^\circ) t, x_2)^t) & \text{otherwise}
\end{cases}
\]

where \( \mathbf{u}_{\text{ext}} \) depends on the solution \( \mathbf{u}^- = (\rho^-, (\rho v_1)^-, (\rho v_2)^-,(\rho E)^-)^t \) itself. From \( \mathbf{u}_{\text{ext}} \) we compute a suitable boundary condition \( \mathbf{u}_{\text{bc}} \) using (2.22). Since the regime of the flow is changing from subsonic to supersonic in the upper boundary condition, we consider the stabilized boundary condition (2.25) in the DG scheme. The overall configuration is shown in Figure 4.15.

![Figure 4.15: Double Mach reflection: initial configuration.](image)

In order to demonstrate that our adaptive scheme is capable of dealing with this configuration, we perform adaptive computations starting with a grid on level 0 containing \( 20 \times 5 \) cells. We perform computations up to \( T = 0.2 \) with different maximum refinement levels \( 5 \leq L \leq 7 \). Since this test case contains discontinuities, we consider the heuristic strategy with \( \beta = 1 \) and \( C_{\text{thr}} = 1 \) for the choice of the threshold value, i.e., \( \varepsilon_L = h_L \). In
Figure 4.16 the adaptive solution (density $\rho$) and the corresponding adaptive grid are exemplarily shown for $L = 6$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4.16.png}
\caption{Double Mach reflection: adaptive solution (density) and corresponding grid for $L = 6$ using the wavelet-free approach.}
\end{figure}

First, we find that all relevant structures of the solution are present in our results by comparing our adaptive results using $L = 6$ with the (uniform) results from other works, for instance [49, 114]. Furthermore, we note that the grid is only refined in areas where discontinuities appear or where the solution consists of small structures needing a high resolution. In regions with less fluctuations the adaptive grid is relatively coarse.

In Figure 4.17 we compare the adaptive solutions and the corresponding grids for different maximum refinement levels $L$ in the blow-up region on the right-hand side of the domain. Thereby we note that the structures are more sharply resolved with increasing resolution. Moreover, instabilities occur at the bottom of the 'red triangle' with increasing resolution. It is well-known and observed by many others that these instabilities occur, cf. [49, 138]. In fact, the absence of these instabilities is usually related to a too coarse resolution or too diffusive stabilization in the scheme, cf. [114, 172]. Thus, it seems to be reasonable that we observe the instabilities only if the maximum refinement level is sufficiently large. Moreover, this shows that the adaptive scheme is capable of resolving these features even if they develop during the computation and are not present from the very beginning.
4.9. NUMERICAL RESULTS

Figure 4.17: Double Mach reflection: comparison of fine structures and adaptive grids on successive refinement levels.

<table>
<thead>
<tr>
<th>Computation</th>
<th>adaptive $N_{\text{max}}$ (wavelet-free)</th>
<th>adaptive $N_{\text{max}}$ (classical)</th>
<th>uniform $N_{\text{max}}$ (reference grid)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L = 4$</td>
<td>6,022 (23.52%)</td>
<td>5,341 (20.86%)</td>
<td>25,600</td>
</tr>
<tr>
<td>$L = 5$</td>
<td>15,520 (15.16%)</td>
<td>13,573 (13.25%)</td>
<td>102,400</td>
</tr>
<tr>
<td>$L = 6$</td>
<td>40,768 (9.95%)</td>
<td>33,469 (8.17%)</td>
<td>409,600</td>
</tr>
<tr>
<td>$L = 7$</td>
<td>124,447 (7.60%)</td>
<td>91,888 (5.61%)</td>
<td>1,638,400</td>
</tr>
</tbody>
</table>

Table 4.2: Double Mach reflection: comparison of the maximum number of cells in the adaptive scheme during the computation.

In order to assess the efficiency of the adaptive scheme and to bring out the difference between the classical and the wavelet-free approach we list the maximum number of cells in the grid $N_{\text{max}}$ for the different computations in Table 4.2. For instance, the adaptive grid for $L = 6$ consists of at most 40,768 cells using the wavelet-free approach and of at most 33,469 using the classical approach, respectively. Again, we observe that the grids using the wavelet-free approach are slightly more refined than the grids using the classical approach. However, the uniform grid of the reference scheme on level $L = 6$ consists of 409,600 cells. Thus, the adaptive grids contain only approx 9.95% and 8.17% of the cells...
of the uniform grid of the reference scheme. For all considered maximum refinement levels the difference between both approaches is around 2–3 percentage points. Compared with the computational complexity of the reference scheme this difference is of minor relevance, since the computational costs are reduced significantly with both approaches.

4.9.3 Shock vortex interaction with a boundary layer in a viscous shock tube

In this section we consider the two-dimensional compressible Navier-Stokes equations in dimensionless form (2.45). When considering flows over a solid surface, boundary layers emerge due to the presence of viscosity. In order to resolve the boundary layers appropriately, normal to the solid surface a fine resolution is required close to the wall, whereas away from the walls a coarse resolution might be sufficient. Since the position of the walls is known a priori, this kind of problems is predesignated for the use of stretched grids. Thus, such a test case is ideal to apply the adaptive scheme using a non-uniform hierarchy and the wavelet-free approach to realize the MRA.

To this end, we focus on the interaction of a shock wave with the boundary layer of an adiabatic wall which is a well-known test case for compressible Navier-Stokes equations, cf. [63, 123, 164, 175]. We consider a quadratic shock tube $[0, 1]^2$ bounded by insulated adiabatic walls. The tube is filled with ideal gas at rest. Initially, a membrane is located at $x_1 = 0.5$ in order to separate two reservoirs containing gas with different densities and pressures. After removing the membrane at $t = 0$, a shock wave followed by a weak contact discontinuity and a weak right-moving rarefaction wave is moving to the right. Due to the presence of viscosity, boundary layers develop at the solid walls at $x_2 = 0$ and $x_2 = 1$. When the shock reaches the wall at $x_1 = 1$ it is reflected. Then, the reflected wave interacts with the boundary layer resulting in complex wave structures.

![Figure 4.18: Shock vortex interaction: initial configuration.](image)

Due to the symmetry of the problem, we compute the numerical solution only on the
4.9. NUMERICAL RESULTS

lower half of the tube, i.e., \( \Omega = [0, 1] \times [0, 0.5] \) and consider symmetry boundary conditions at \( x_2 = 0.5 \). The initial configuration is shown in Figure 4.18. The symmetry boundary conditions are realized by imposing

\[
\mathbf{u}_{bc} = (\rho^-, (\rho v_1)^-, (\rho v_2)^-, (\rho E)^-)^T.
\]

as boundary state for the evaluation of the fluxes in the boundary integrals in (2.29). Due to the presence of viscosity we abstain from using the characteristic boundary condition (2.22) and consider directly (4.40). The viscous (boundary) flux is chosen as

\[
\mathbf{G}_{bc} = \mathbf{G}(\mathbf{u}_{bc}, \nabla_h \mathbf{u}^- + \mathbf{R}_e(\mathbf{u}_{bc} - \mathbf{u}_h^-)).
\]

The adiabatic wall boundary condition is prescribed by using the boundary state

\[
\mathbf{u}_{bc} = (\rho^-, 0, 0, (\rho E)^- - \rho^- (\mathbf{v}^-)^t(\mathbf{v}^-))^t
\]

and using the viscous (boundary) boundary flux

\[
\mathbf{G}_{bc} = \frac{1}{Re} \begin{pmatrix}
0 \\
0
\end{pmatrix}
\begin{pmatrix}
\mathbf{\sigma}(\mathbf{u}_{bc}, \nabla_h \mathbf{u}^- + \mathbf{R}_e(\mathbf{u}_{bc} - \mathbf{u}_h^-))
\end{pmatrix},
\]

where the stress tensor \( \mathbf{\sigma} \) is computed by (2.46) using \( \mathbf{u}_{bc} \) and the stabilized inner gradient \( \nabla_h \mathbf{u}^- + \mathbf{R}_e(\mathbf{u}_{bc} - \mathbf{u}_h^-) \). We consider two situations with different Reynolds numbers, namely \( Re = 200 \) and \( Re = 1000 \).

Numerical results for \( Re = 200 \)

First, we consider a Reynolds number of \( Re = 200 \). In this configuration the influence of the viscous terms in the compressible Navier-Stokes equations is comparably large such that it is questionable if this problem can be considered as a purely convection-dominated problem. Nevertheless, this configuration has been investigated more detailed and more frequently than the \( Re = 1000 \) configuration, cf. [63, 123, 164, 175]. Thus, we consider this configuration for the validation of our adaptive method. More precisely, we investigate whether the adaptive scheme is capable of resolving all relevant features in the solution. Since the exact solution is not known, we compare our results with available results in the literature.

To this end, we first perform computations using a uniform subdivision of the cells. The quality of the solution is strongly influenced by the grid resolution near the wall. For that reason, in [123] (uniform) computations were performed on a stretched grid using a \( tanh \) stretching function. Thereby, the cells are clustered near the walls. We pick up this idea and perform adaptive computations using a non-uniform hierarchy where the subdivision of the cells is stretched towards the wall boundary at \( y = 0 \). For the purpose
of obtaining the non-uniform hierarchy by mapping of a uniform dyadic hierarchy as described in Section 3.1.1, we consider the stretching function

\[
\theta(x) = \left( \frac{1}{2} + \tanh (\zeta (2 x_2 - 1)) / (2 \tanh (\zeta)) \right),
\]

where the strength of the stretching is specified by \( \zeta \). For \( \zeta \) tending to zero the stretching coincides with the identity resulting in a uniform dyadic hierarchy, whereas for increasing \( \zeta \), the cells are more and more condensed toward \( x_2 = 0 \). Here, we consider \( \zeta = 1.2 \). In this hierarchy, there holds \( \min_{\mu \in \nu} \text{diam}(V_\mu) \leq \text{diam}(V_\nu) / 2 \) for all \( \lambda \). Thus, in the choice for the local thresholds as described in Section 4.5.2, we use the diameter instead of the \( d \)th root of the volume to quantify the local resolution.

We apply the heuristic strategy using \( C_{\text{thr}} = 1 \) and \( \beta = 1 \). In order to realize the MRA and grid adaptation, we consider the wavelet-free approach to avoid the costly cell-wise construction of multiwavelets on the non-uniform hierarchy. We perform adaptive computations with different maximum refinement levels \( 3 \leq L \leq 5 \) based on a grid consisting of \( 30 \times 15 \) cells on level 0. These computations are performed on a hierarchy with a uniform subdivision as well as with a non-uniform subdivision using the stretching as described above. We perform the computations up to \( T = 1 \).

The adaptive solutions with the corresponding grids at \( T = 1 \) are shown in Figure 4.19, Figure 4.20 and Figure 4.21 for the uniform hierarchy and the non-uniform hierarchy, respectively. First of all we note that with sufficient resolution, i.e., sufficiently large \( L \), our solutions reveal the same structures as computed by other groups using solvers with uniform grids, cf. [63, 123, 164, 175]. Thus, the adaptation seems to be capable of resolving all relevant features of this configuration. Moreover, we note that the lambda structure at approx. \( x_1 = 0.4 \) (the ‘red’ triangle) is resolved more sharply with increasing resolution. Furthermore, we observe that the vortices are deformed if the resolution is not sufficient, cf. Figures 4.19 and 4.20. Especially the right vortex near \( x_1 = 0.8 \) differs for the computations with different maximum refinement levels. Here, we observe that with a better resolution of the boundary layer of the wall at \( x_2 = 0 \) the vortex straightens up. This can be either achieved by increasing the maximum refinement level or condensing the cells towards the wall using the stretching in the non-uniform dyadic hierarchy: this vortex is better resolved in the stretched \( L = 4 \) computation than in the corresponding computation with a uniform subdivision.

To analyze the quality of the solutions and to justify grid convergence, the profiles of the density at the wall \( x_2 = 0 \) as well as the momentum in \( x_2 \)-direction at \( x_2 = 0.05 \) are compared for increasing grid resolution, cf. [123, 164]. In Figures 4.22 and 4.23 we plot these profiles. Comparing our results with the results in [123, 164] we conclude that (i) with increasing resolution the solutions seem to converge and (ii) our results are in agreement with results from other groups. Moreover, we observe that considering a stretched hierarchy has a positive effect on the convergence behavior of the scheme: higher resolution near the wall increases the quality of the adaptive solution.
Figure 4.19: Shock vortex interaction (Re = 200): adaptive solutions and grids for L = 3.
Figure 4.20: Shock vortex interaction (Re = 200): adaptive solutions and grids for L = 4.
Figure 4.21: Shock vortex interaction ($Re = 200$): adaptive solutions and grids for $L = 5$. 
Figure 4.22: Shock vortex interaction ($Re = 200$): distribution of the density at the wall $x_2 = 0$.

Figure 4.23: Shock vortex interaction ($Re = 200$): distribution of the momentum in $x_2$-direction at $x_2 = 0.05$. 
4.9. **Numerical Results**

So far, we have only addressed the quality of the solution. Next, we investigate the acceleration of the adaptive scheme through the grid adaptation. To this end, we consider the maximum number of cells in the adaptive grid to estimate the gain of adaptivity. In Table 4.3 we list the maximum number of cells for the different computations and compare them with the number of cells in the fully refined grid of the reference scheme. Here, we note that maximum number of cells during the computation does not differ significantly for the uniform and the stretched computation on the same refinement level. However, the quality of the solution using a stretched hierarchy is better as discussed above. Thus, we benefit from the stretching in terms of efficiency of the adaptive scheme as well.

<table>
<thead>
<tr>
<th>Comput.</th>
<th>$N_{\text{max}}$ (uniform-hierarchy)</th>
<th>$N_{\text{max}}$ (stretched-hierarchy)</th>
<th>$N_{\text{max}}$ (reference grid)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L = 3$</td>
<td>10,206 (35.44%)</td>
<td>10,509 (36.49%)</td>
<td>28,800</td>
</tr>
<tr>
<td>$L = 4$</td>
<td>33,348 (28.95%)</td>
<td>34,764 (30.18%)</td>
<td>115,200</td>
</tr>
<tr>
<td>$L = 5$</td>
<td>92,355 (20.04%)</td>
<td>96,240 (20.89%)</td>
<td>460,800</td>
</tr>
</tbody>
</table>

Table 4.3: Shock vortex interaction ($Re = 200$): comparison of the maximum number of cells in the adaptive scheme during the computation.

In summary we note that the adaptation is capable of resolving all relevant features of the flow using a uniform grid hierarchy as well as a stretched non-uniform hierarchy. Moreover, we benefit from the additional resolution near the wall resulting from the stretching in terms of the quality in which the vortices are resolved and in efficiency of the scheme.

**Numerical results for $Re = 1000$**

Next, we consider the flow for Reynolds number $Re = 1000$. Here, due to the higher Reynolds number less viscosity is present. For that reason, the boundary layer which develops as a consequence of the initial Riemann problem is thinner. In order to appropriately resolve this layer, a fine resolution near the wall is required. To this end, we condense the cells more strongly to the wall at $x_2 = 0$ using the $\text{tanh}$ stretching (4.44) but with a larger parameter $\zeta = 1.8$. Due to the higher Reynolds number the flow reveals a more complex wave structure consisting of more vortices. For that reason, a higher resolution is required and consequently this configuration is computationally more expensive than the previous configuration. Therefore, we abstain from performing a computation on a uniform hierarchy.

We perform adaptive computations with maximum refinement level $3 \leq L \leq 5$ up to $T = 1$. The simulation parameters are the same as for $Re = 200$ computations. The adaptive solutions and the corresponding grids at $T = 1$ are shown in Figure 4.24. First of all, we note that with sufficient resolution, i.e., sufficiently large $L$ our solutions reveal
Figure 4.24: Shock vortex interaction ($Re = 1000$): adaptive solutions and grids at $t = 1$ for $3 \leq L \leq 5$ using a \textit{stretched-dyadic} subdivision.
the same structures as computed by other groups using solvers with uniform grids, cf. [63,164,175]. Hence, we observe that the grid adaptation is capable of refining the relevant structures in the solution. The solution using $L = 5$ reveals some indications that this flow might be unstable. This is backed up by observations in [63,164].

![Graph showing shock vortex interaction](image)

**Figure 4.24:** Shock vortex interaction ($Re = 1000$): adaptive solutions and grids at $t = 1$ for $3 \leq L \leq 5$ using a **stretched-dyadic** subdivision.

In order to access the efficiency of the grid adaptation, we again focus on the maximum number of cells during the computation in the adaptive grid $N_{\text{max}}$ provided in Table 4.4. Here, we note that the adaptation is capable of reducing the computational cost significantly.
Table 4.4: Shock vortex interaction ($Re = 1000$): comparison of the maximum number of cells in the adaptive scheme during the computation.

<table>
<thead>
<tr>
<th>Computation</th>
<th>$N_{\text{max}}$ (stretched-hierarchy)</th>
<th>$N_{\text{max}}$ (reference grid)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L = 3$</td>
<td>12,438 (43.19%)</td>
<td>28,800</td>
</tr>
<tr>
<td>$L = 4$</td>
<td>43,989 (38.18%)</td>
<td>115,200</td>
</tr>
<tr>
<td>$L = 5$</td>
<td>146,937 (31.88%)</td>
<td>460,800</td>
</tr>
</tbody>
</table>

4.9.4 The bottom line of the (numerical) comparison of the wavelet-free and the classical approach

The numerical results presented in this section lead to the conclusion that the wavelet-free approach is a suitable alternative to the classical wavelet-based approach to overcome the issues related to the construction of wavelets. However, the grids resulting from the use of the classical approach are slightly coarser than the ones using the more general wavelet-free approach. Thereby, both approaches maintain the accuracy of the reference scheme. For that reason, the classical approach might be a good choice whenever the construction of the multiwavelets can be realized efficiently, e.g., when it is sufficient to construct them only on a single reference element. However, the additional refinement of the wavelet-free approach is comparably small in comparison to the overall gain from the grid adaptation. Thus, in most situations this overhead is negligible. Moreover, the realization of the MRA and the adaptive scheme using the wavelet-free approach is computationally less costly than using the classical approach. Therefore, the wavelet-free approach is the preferable approach in the general case.
Chapter 5

Grid adaptation for shallow water equations

In this section we apply the dynamical grid adaptation from Chapter 4 to the shallow water equations (SWE) which are a hyperbolic system of conservation laws including an additional source term. These equations are a simplified model for shallow free surface water flows over the non-constant bottom topography. The SWE are derived from the Navier-Stokes equations via depth-integration. Thereby, the spatial dimension is reduced by one, i.e., considering two-dimensional SWE is sufficient to model a three-dimensional problem. The use of SWE is very common since using the (higher dimensional) Navier-Stokes equations becomes extremely costly or even unfeasible with increasing size of the underlying computational domain. The SWE are given by

$$\frac{\partial}{\partial t} \begin{pmatrix} h \\ h\mathbf{v} \end{pmatrix} + \nabla \cdot \begin{pmatrix} h\mathbf{v} \\ h\mathbf{v} \otimes \mathbf{v} + \frac{1}{2} gh^2 \mathbf{I} \end{pmatrix} = \begin{pmatrix} 0 \\ -g h \nabla b \end{pmatrix},$$

where the unknown quantities are the water height $h$ and the velocity vector $\mathbf{v}$. The given bottom topography is represented by $b$, the gravitational acceleration is denoted by $g$ and $\mathbf{I}$ is the $d \times d$ identity matrix.

When discretizing the SWE, two important issues related to the bottom topography and the source term have to be addressed: a numerical scheme must be well-balanced [23,88,92], i.e., maintain steady states over non-constant topography, as well as positivity-preserving [77], i.e., the water height must not become negative. Thus, the reference scheme has to be adjusted in order to ensure these properties. For that purpose, we consider an available approach from the literature, cf. [180,182].

In order to reduce the computational complexity of the reference scheme, we consider local grid adaptation. In an adaptive scheme the aforementioned constraints are not automatically fulfilled. However, the grid adaptation must not spoil well-balancing and positivity-preserving of the reference scheme to obtain a stable scheme. Therefore,
the main focus of this chapter is to investigate how to ensure that these properties are maintained during the adaptation process. To this end, we modify the adaptive strategy presented in Chapter 4 and verify that well-balancing and positivity-preserving are maintained during the grid adaptation. Parts of this work have already been published in [80]. Furthermore, the presented strategy has been applied in further works [38, 120].

5.1 The reference scheme

To ensure well-balancing and positivity-preserving of the reference scheme, we apply the strategy of Xing et al. [180, 182]. For convenience of the reader, we briefly summarize the modifications to the reference scheme given in Chapter 2 in the following.

5.1.1 Treatment of bottom topography

In order to handle the bottom topography in the source term, Xing et al. [180] suggest to determine a representation $b_h \in S_L$ of the bed topography $b$. Thus, the source term in the DG scheme is treated as

$$
\left\langle \begin{pmatrix} 0 \\ -gh \nabla_h b_h \end{pmatrix}, v \right\rangle_{\Omega}.
$$

An advantage of using the representation $b_h$ instead of $b$ in the DG scheme is that one can easily switch between $h$ and $h + b$ by simply adding and subtracting the coefficients in the basis expansion of $h$ and $b$, respectively. A switch between $h$ and $h + b$ is helpful in order to ensure well-balancing of the reference scheme.

For determining $b_h \in S_L$ from $b$ we proceed as follows: if the bottom topography is provided by a function $b \in L_\infty(\Omega)$ this is done by an $L_2$-projection, i.e., $b_h$ is defined by the weak formulation

$$
\langle b_h, v \rangle_{\Omega} = \langle b, v \rangle_{\Omega} \quad \forall v \in S_L.
$$

If one is furthermore interested in a continuous representation for the bed, i.e., $b_h \in C(\Omega) \cap S_L$, an interpolation can be used instead of a projection. In several applications the bottom topography is not provided as an explicitly represented function but as a set of discrete samples $(x_i, b_i)$. Then, $b_h$ must be locally interpolated or locally computed by solving least square problems, i.e.,

$$
\sum_{x_i : x_i \in \Gamma_{\lambda}} (b_h(x_i) - b_i)^2 \to \min, \quad \lambda \in I_L.
$$

For details we refer to [82]. If desired, continuity of $b_h$ can be thereby enforced as well.
5.1. THE REFERENCE SCHEME

5.1.2 Well-balancing

When discretizing SWE with a non-constant bottom topography $b$ one difficulty is to preserve the lake at rest, where

$$h + b = \text{const} \quad \text{and} \quad \mathbf{v} = \mathbf{0}.$$  

Since $h$ is not constant in this configuration, the fluxes in (5.1) are non-zero. However, here these fluxes are balanced with the source term in (5.1). If the numerical discretization does not guarantee this balancing as well, spurious oscillations might appear during the computation. An extension of the lake at rest to a steady state flow is of interest where the fluxes and source terms have to be balanced in order to maintain the steady state. A numerical scheme satisfying this is called well-balanced [88,92].

In order to ensure well-balancing, the DG scheme (2.23) has to be modified. For that purpose, several approaches are available in the literature. In the following we shortly discuss three different strategies. The simplest way to guarantee well-balancing is to use a continuous representation $b_h \in S_L$ of the bottom topography in the scheme, cf. [180]. Alternatively, the computation of the source term can be modified, cf. [181]. The drawback of this additional modification of the source term is that it is computationally expensive and the extension to the general multidimensional case is not trivial. For that reason, a simpler and more flexible strategy has been proposed by Xing et al. [180]. This strategy is motivated by the well-balancing strategy of Audusse et al. [13] for finite volume schemes. The basic idea is to substitute the numerical fluxes $\mathbf{F}$ in the surface integrals

$$\langle \mathbf{F}(u_h, n_F), [[\mathbf{v}]] \rangle_{\Gamma_f}$$

in (2.23) by modified fluxes

$$\langle \tilde{\mathbf{F}}(u_m, \tilde{n}), [[\mathbf{v}]] \rangle_{\Gamma_f} + \frac{g}{2} \left( (h^+)^2 - (h^-)^2 \right) \left( \frac{1}{\rho_h} \right)_{\tilde{n}},=[[\mathbf{v}]]_{\Gamma_f}. \quad (5.2)$$

The modified values are defined by

$$u_m^\pm := (h_m^\pm, (h_m^\pm \mathbf{v}_h^\pm))^t, \quad \text{where} \quad h_m^\pm := \max(0, h^+ + b^+ - \max(b^+, b^-)).$$

In case of a static grid, Xing et al. [180] proved that the resulting scheme preserves the lake at rest.

In order to guarantee well-balancing of the reference scheme, we pursue the following strategy: we incorporate the flux modification (5.2) from Xing et al. [180] into our scheme in order to be able to deal with discontinuous bottom topographies. However, the modification is triggered by the magnitude of the discontinuity of $b_h$ at the interfaces. Thus, in case of a continuous representation of the bottom topography this strategy coincides with the firstly mentioned strategy. Hence, our scheme can deal with continuous and discontinuous representations $b_h$ of the bottom topography $b$. 
5.1.3 Positivity-preserving

Another difficulty in discretizing the SWE is the treatment of wetting and drying. Thereby, the key problem is that in the actual computation the water height might locally become negative. That has to be avoided under any circumstances since it results in instabilities. For that purpose, one has to ensure that the water height \( h \) remains non-negative at the quadrature nodes (2.42) during the computation. To this end, a drying tolerance \( \delta_{\text{dry}} \) is introduced: if \( h \) is smaller than this tolerance at a point this is considered as dry, cf. [148,182]. If \( h \) tends to zero \( v = \frac{h}{h} \) cannot be computed properly from the momentum \( hv \) in the evolution of the DG scheme. Thus, if \( h \leq \delta_{\text{dry}} \) we set \( v = 0 \). To find the right choice for this tolerance can be challenging. If it is chosen too large, the quality of the solution is damaged. However, if it is chosen too small the scheme becomes unstable, i.e., oscillations occur at the wet/dry interface resulting in a dramatical reduction of the time step size.

In order to ensure the positivity of \( h \), we apply the strategy of Xing et al. [182] and Gandham et al. [78]. Xing et al. proved that the local mean values of the water height \( h \) remain positive during the time evolution of the DG scheme. By \( u_L^L \) we denote the fully discrete DG solution at time step \( t_n \). Due to (3.28) we can rewrite \( u_L^L \) as the sum of local contributions, i.e.,

\[
u_L^L(x) = \sum_{\lambda \in L} u_{\lambda}^{n,L}(x),
\] (5.3)

where the local contributions are defined by \( u_{\lambda}^{n,L} := u_L^L 1_{V_{\lambda}} \). The local mean values are denoted by

\[
\bar{u}_{\lambda}^{n,L} \equiv (\bar{h}_{\lambda}^{n,L}, \bar{u}_{\lambda}^{n,L}) := \frac{1}{|V_{\lambda}|} \int_{V_{\lambda}} u_{\lambda}^{n,L}(x) dx.
\] (5.4)

The basic idea of the strategy of Xing et al. is to modify the high-order part of the DG solution in order to ensure positivity at all quadrature nodes \( x_i \in X_{\lambda} \). To this end, \( u^n \) is replaced by

\[
u^n(x) = \sum_{\lambda \in L} \bar{u}_{\lambda}^{n,L}(x).
\] (5.5)

The modified DG solution \( \bar{u}^n(x) \) is defined by a local down-scaling of the high-order parts using a bound-preserving limiter [140,188], i.e.,

\[
\bar{u}_{\lambda}^{n,L} := \theta_{\lambda} \left( u_{\lambda}^{n,L} - \bar{u}_{\lambda}^{n,L} \right) + \bar{u}_{\lambda}^{n,L},
\] (5.6)

where the damping factor \( \theta_{\lambda} \) is defined by

\[
\theta_{\lambda} := \min \left\{ 1, \frac{h_{\lambda}^n}{h_{\lambda}^n - h_{\text{min},\lambda}} \right\} \quad \text{and} \quad h_{\text{min},\lambda} := \min_{x_i \in X_{\lambda}} h^n(x_i).
\] (5.7)
Due to the non-negativity of the water height in the mean, non-negativity in all quadrature nodes used in the DG scheme is thereby guaranteed. The local mean value is left unmodified in order to ensure conservation. Furthermore, it has been verified in [188] that this modification does not spoil the order of consistency of the DG-solution.

5.1.4 Local projection limiting for shallow water equations

In order to obtain a stable scheme, a limiter has to be applied after each stage of the Runge-Kutta scheme. The limiting must not conflict with well-balancing and positivity-preserving. A direct application of the limiter described in Algorithm 2.1 would spoil both properties. For this reason, Xing et al. [182] modified the TVB limiter of Cockburn et al. [48]. In order to ensure well-balancing, the limiter described in Algorithm 2.1 is applied to $h + b, v$ instead of $h, v$. In some quadrature points the limiting might result in a negative water height $h$. Since the limiter only acts on the higher-order coefficients it is guaranteed that the local average of the water height $h$ remains positive. Thus, after applying the TVB limiter, we additionally apply the positivity limiter (5.6). The resulting limiter is summarized in Algorithm 5.1.

**Algorithm 5.1** Limiting for SWE addressing positivity-preserving and well-balancing

1: for all $\lambda \in \mathcal{I}$ do
2: \hspace{1cm} if $V_\lambda$ is completely wet, i.e., $\min_{x_i \in X_\lambda} h(x_i) > \delta_{\text{dry}}$ then
3: \hspace{2cm} apply TVB Limiter from Algorithm 2.1 on $(h + b, v)^T$.
4: \hspace{1cm} else
5: \hspace{2cm} project $u_h$ to $\Pi_1$ using (2.41) and apply the TVB limiter from Algorithm 2.1 on $(h, hv)^T$ with Shu constant $M = 0$ in conserved variables, i.e., omit the
6: \hspace{2cm} transition to local characteristic variables in Algorithm 2.1.
7: \hspace{1cm} end if
8: \hspace{1cm} compute $\theta_\lambda$ according to (5.7) and apply the positivity limiter (5.6) in order to
9: \hspace{2cm} ensure positivity of $h$ at all quadrature nodes $x_i \in X_\lambda$.
10: end for

5.2 Grid adaptation for shallow water equations

To ensure that the adaptation procedure does not spoil well-balancing and positivity-preserving we modify the adaptive strategy presented in Chapter 4. First of all, we note that we can apply the DG scheme including the modifications by Xing et al. [182] on a fixed non-uniform grid which is constant in time and thereby ensure positivity-preserving and well-balancing. Thus, we focus on the dynamical changes in the grid, i.e., we have to ensure that these properties are preserved during coarsening and refinement.
For that reason, we shortly recap the structure of coarsening and refinement in the dynamical grid adaptation from Chapter 4. First, we consider grid coarsening: after thresholding \( \tilde{u}_{n+1}^\varepsilon \in (S^{n+1})^m \) is projected to a subspace \( S_{\varepsilon}^{n+1} \subset (S^{n+1})^m \) corresponding to the coarser grid. Next, we focus on refinement: the solution \( u_{n}^\varepsilon \) from the previous time step \( t_n \) is represented in the basis of the richer space \( (S^{n+1})^m \) corresponding to the predicted refined grid \( \mathcal{I}^{n+1} \) of the next time step. We call this the prolongation to the refined grid. Note that since \( S_{\varepsilon}^{n} \subset S^{n+1} \) the prolongation is only a change of basis and thus, the function is not modified.

Hence, we modify the coarsening and refinement steps in the adaptive scheme from Chapter 4 such that well-balancing and positivity-preserving are maintained during these steps. For that purpose, we proceed as follows. First, we consider the treatment of the bottom topography. Then, we focus on modifications related to the well-balancing constraint. Afterwards, we concentrate on positivity-preserving in the adaptation process. Finally, we summarize the modified adaptive scheme for the SWE and present numerical results.

### 5.2.1 Bottom topography and well-balancing

The well-balancing strategy of the reference scheme is based on the assumption that a finite-dimensional representation \( b_h \in S_L \) of the bed topography \( b \) is used in the scheme; this representation is claimed to be in the same space as the solution \( u_h \). When performing dynamical grid adaptation, the grid is altered from time step to time step and thus, the corresponding discretization spaces \( S^n \) and \( S_{\varepsilon}^n \) change as well. In principle, we could use the same representation \( b_h \in S_L \) as for the reference scheme. However, this would result in high computational cost and memory requirements. For reason of efficiency we use an adaptive representation \( b_{h, \varepsilon} \) corresponding to a static adaptive grid. This can be obtained by applying thresholding (3.40) to \( b_h \). Thus, only significant local contributions of \( b_h \) are considered. By \( D_{\varepsilon}^b \) we denote the index set of significant contributions of \( b_{h, \varepsilon} \) by means of (3.41).

In order to adopt the well-balancing and positivity-preserving strategies of the reference scheme to the adaptive scheme, we claim that

\[
b_{h, \varepsilon} \subset S_{\varepsilon}^n
\]

holds for each time step \( t_n \). Since the adaptive representation is fixed in advance, \( (5.8) \) is an additional constraint to the choice of the spaces \( S_{\varepsilon}^n \) and \( S^n \), respectively. Thus, the dynamical adaptive grid has to be at least as fine as the adaptive grid of the bottom topography. This can be realized by adding the additional constraints \( D_{\varepsilon}^b \subset D^n \) and \( D_{\varepsilon}^b \subset D_{\varepsilon}^n \) to the prediction and thresholding procedure, respectively. For further details we refer to [80]. Moreover, we ensure by (5.8) that the shape of \( b \) is involved in the adaptation process, i.e., whenever \( b \) is non-smooth the corresponding grid is refined in these areas.
5.2. GRID ADAPTATION FOR SHALLOW WATER EQUATIONS

The thresholding procedure in the coarsening step might spoil the well-balancing. For that reason, the adaptivity criteria are based on \( h + b, \mathbf{v} \) instead of \( h, \mathbf{v} \). Thus, the MRA, the prediction and the thresholding are performed on \( h + b, h \mathbf{v} \).

5.2.2 Positivity-preserving

Since the grid dynamically changes from time step to time step, we have to ensure that the positivity is not spoiled by the adaptation process. In the time evolution with the DG scheme it is ensured in the mean and at all quadrature nodes. Our objective is to ensure positivity in the mean on all cells in the adaptive grid after coarsening and refinement. Then, it can be guaranteed at all quadrature nodes by applying the positivity limiter (5.6). For that purpose we first focus on the coarsening step and then consider the refinement step.

Grid coarsening

First, we analyze the positivity in the coarsening step realized by the thresholding procedure. The foundation for grid coarsening is the multi-scale decomposition of \( \tilde{u}_{L,\varepsilon}^n \in (S^n)^m \), i.e.,

\[
\tilde{u}_{L,\varepsilon}^n = \sum_{\ell=0}^L \sum_{\lambda \in \mathcal{I}_\ell \cap \mathcal{I}_\varepsilon} \tilde{u}_\lambda^{\ell,n} = \sum_{\lambda \in \mathcal{I}_0} \bar{u}_\lambda^0 + \sum_{\ell=0}^{L-1} \sum_{\lambda \in \mathcal{I}_\ell \cap \mathcal{I}_\varepsilon} \tilde{d}_\lambda^{\ell}.
\]  

(5.9)

The coarsening is realized by neglecting non-significant local contributions \( \tilde{d}_\lambda^{\ell} \) in (5.9) according to (4.5) and (3.40). By this, a subspace \( S^n_\varepsilon \subset S^b \) is determined and \( \tilde{u}_{L,\varepsilon}^n \) is projected component-wise to this smaller space, i.e.,

\[
u_{L,\varepsilon}^n := P_{S^n_\varepsilon} \left( \tilde{u}_{L,\varepsilon}^n \right) = \sum_{\ell=0}^L \sum_{\lambda \in \mathcal{I}_\ell \cap \mathcal{I}_\varepsilon} \tilde{u}_\lambda^{\ell,n}.
\]

This projection is realized by recursively applying Algorithm 3.2 to determine the projections to the next coarser level. This is performed cell-wise with two-scale transformations using Algorithm 3.4 and 3.5, respectively. Thus, we first consider the projection of \( \Sigma_{\mu \in \mathcal{M}_\lambda} u_{\mu}^{\ell+1} \) to the local contribution \( u_\lambda^{\ell} \) corresponding to the father cell \( V_\lambda \) and analyze the positivity in this local two-scale decomposition.

**Proposition 5.1 (Positivity in the local two-scale decomposition)**

Let \( \lambda \in \mathcal{I}_\ell \) and \( u_{\mu}^{\ell+1} = (h_{\mu}^{\ell+1}, h_{\mu}^{\ell+1})^t \), \( \mu \in \mathcal{M}_\lambda \), be local contributions of \( u_{\ell+1} \in (S_{\ell+1})^m \) corresponding to the subcells of \( V_\lambda \). Let \( u_\lambda^{\ell} = (h_\lambda^{\ell}, h_\lambda^{\ell})^t \) denote the component-wise projection onto \( S_\ell \), i.e.,

\[
u_\lambda^{\ell} = P_{S_\ell} \left( \sum_{\mu \in \mathcal{M}_\lambda} u_{\mu}^{\ell+1} \right).
\]
Furthermore, we assume that the local contributions on level $\ell + 1$ are locally positive in the mean, i.e.,

$$\int_{V_\mu} h^\ell_{\mu}(x) dx \geq 0, \quad \mu \in M_\lambda. \tag{5.10}$$

Then, the mean value of $h^\ell_\lambda$ is positive, i.e.,

$$\int_{V_\lambda} h^\ell_\lambda(x) dx \geq 0. \tag{5.11}$$

Proof: From Conclusion 3.2 we conclude using (3.28) that

$$\int_{V_\lambda} u^\ell_\lambda(x) dx = \sum_{\mu \in M_\lambda} \int_{V_\mu} u^{\ell + 1}_\mu(x) dx. \tag{5.12}$$

Finally, we deduce (5.11) from the first component of (5.12) using (5.10). □

Next, we extend the results from Proposition 5.1 to the complete adaptive solution $u^n_{L, \varepsilon}$.

**Conclusion 5.1 (Positivity in the mean after coarsening)**

If $\tilde{u}^n_{L, \varepsilon} = (h^n_{L, \varepsilon}, (h^\varepsilon u^n_{L, \varepsilon})^t)$ is locally positive in the mean, i.e.,

$$\int_{V_\lambda} \tilde{h}^n_{L, \varepsilon}(x) dx \geq 0, \quad \forall \lambda \in I^n, \tag{5.13}$$

then, $u^n_{L, \varepsilon} = (h^n_{L, \varepsilon}, (h^\varepsilon u^n_{L, \varepsilon})^t)$ is locally positive in the mean as well, i.e.,

$$\int_{V_\lambda} h^n_{L, \varepsilon}(x) dx \geq 0, \quad \forall \lambda \in I^n. \tag{5.14}$$

Proof: Since $u^n_{L, \varepsilon} := P_{S^n}(\tilde{u}^n_{L, \varepsilon})$ is the component-wise projection of $\tilde{u}^n_{L, \varepsilon} \in (S^n)^m$ to the subspace $(S^n)^m \subset (S^n)^m$, $u^n_{L, \varepsilon}$ can be written as a composition of local two-scale projections of $\tilde{u}^n_{L, \varepsilon}$, see (3.30) and Theorem 3.1(ii). Thus, we conclude (5.14) from (5.13) by repeatedly applying Proposition 5.1. □

Thus, the positivity of the water height $h$ after the coarsening via thresholding can be guaranteed in the mean. In order to ensure positivity at all quadrature nodes, we apply the positivity limiter (5.6) cell-wise after the thresholding procedure and the corresponding grid coarsening.

**Grid refinement**

Next, we consider the refinement step. Here, the crucial step is the prolongation, i.e., the computation of the coefficients in

$$u^n_{L, \varepsilon} = \sum_{\ell=0}^L \sum_{\lambda \in I^{n+1} \cap I_\ell} u^{n\ell}_\lambda = \sum_{\ell=0}^L \sum_{\lambda \in I^{n+1} \cap I_\ell} (\Phi_\lambda(x))^T U^n_{\lambda}. \tag{5.15}$$
5.2. GRID ADAPTATION FOR SHALLOW WATER EQUATIONS

from the coefficients corresponding to the coarser grid in

\[ u_{L,\varepsilon}^n = \sum_{\ell=0}^L \sum_{\lambda \epsilon P L \cap \mathcal{T}_\varepsilon} u_{\ell,\lambda}^n = \sum_{\ell=0}^L \sum_{\lambda \epsilon P L \cap \mathcal{T}_\varepsilon} (\Phi(\lambda)(x))^T U^{n}. \]

This is realized by recursively applying Algorithm 3.3 to determine the coefficients of the basis expansion corresponding to the next finer level. This is performed cell-wise using Algorithm 3.6 and 3.7, respectively. Here, the contributions from the orthogonal complement are zero since we only deal with additional refinement. Thus, Algorithm 3.6 as well as Algorithm 3.7 are reduced to

\[ \overline{U}^{n}_{\mathcal{M}_\lambda} = (M_{\lambda,0})^T \overline{U}^{n}_{\mathcal{M}}. \]

Figure 5.1: Counterexample for positivity in refinement: the water height is highlighted in blue and the corresponding mean values are highlighted in red. The water height and the corresponding mean value on the coarse grid are shown on the left. In the middle, the same is shown for the refined grid. The modified water height and the resulting mean values on the refined grid are shown on the right.

Again, we first focus on a single local contribution \( u_{\lambda}^\ell = (h_{\lambda}^\ell, h_{\lambda}^\ell v_{\lambda}^\ell)^t \). For ease of notation we neglect the superscripted \( n \) in the following. The key question is whether the water height in the mean on the subcells \( V_{\mu}, \mu \in \mathcal{M}_\lambda \), is non-negative if the water height in the mean on \( V_{\lambda} \) of \( u_{\lambda}^\ell \) is non-negative, i.e.,

\[ \int_{V_{\lambda}} h_{\lambda}^\ell dx \geq 0 \ \Rightarrow \ \int_{V_{\mu}} h_{\lambda}^\ell dx \geq 0, \ \mu \in \mathcal{M}_\lambda. \]  

(5.16)

If this holds true, the positivity at all quadrature nodes of the refined grid can be guaranteed by applying the positivity limiter (5.6) after the grid refinement. In fact, (5.16) does not generally hold true: even if \( \int_{V_{\lambda}} h_{\lambda}^\ell dx > 0 \), the mean values of the water height on the subcells \( V_{\mu}, \mu \in \mathcal{M}_\lambda \), can be negative, i.e.,

\[ \int_{V_{\mu}} h_{\lambda}^\ell dx < 0. \]
In Figure 5.1 an example for this situation is depicted for $p = 2$.

In the following we denote the mean value of $u^x_\lambda$ by $\overline{u}_\lambda^x$.

Motivated by the bound preserving limiter (5.6) we introduce an additional bound preserving limiter with a different objective: modify $u^x_\lambda(x)$ such that equation (5.16) holds true. For that purpose, we define a modified local contribution by

$$
\hat{u}_\lambda^x(x) = \alpha_\lambda \left( u^x_\lambda(x) - \overline{u}_\lambda^x \right) + \overline{u}_\lambda^x,
$$

where the scaling factor $\alpha_\lambda$ is defined analogously to (5.7) as

$$
\alpha_\lambda = \min \left( 1, \frac{\overline{h}_\lambda^x}{\overline{h}_\lambda - \min_{\mu \in \mathcal{M}_\lambda} \frac{1}{|V_\mu|} \int_{V_\mu} h_\lambda^x(x) dx} \right).
$$

In the right graph of Figure 5.1 the unmodified water height (dashed) and the corresponding modified water height for the counterexample discussed above are shown in blue. Furthermore, the mean values of the modified solution on the subcells are shown additionally. In the following theorem we verify that (5.16) holds true for the modified solution. Moreover, we prove that the modification is conservative and only influences cells which are partially dry.

**Theorem 5.1 (Properties of the local modification)**

Let $u^x_\lambda = (h^x_\lambda, (h^y_\lambda)_y)$ be a local contribution and let $\hat{u}_\lambda^x = (\hat{h}_\lambda^x, (\hat{h}^y_\lambda)_y)$ be the corresponding modified local contribution using the positivity correction (5.17). Moreover, we assume that $h^x_\lambda$ is non-negative in the mean, i.e.,

$$
\int_{V_\lambda} h^x_\lambda dx \geq 0.
$$

Then, the following statements hold true:

(i) The modification (5.17) is conservative, i.e.,

$$
\int_{V_\lambda} \hat{u}_\lambda^x(x) dx = \int_{V_\lambda} u^x_\lambda(x) dx.
$$

(ii) If $\min_{\mu \in \mathcal{M}_\lambda} \frac{1}{|V_\mu|} \int_{V_\mu} h^x_\lambda(x) dx \geq 0$ then $\hat{u}_\lambda^x = u^x_\lambda$.

(iii) The modified local contribution using the positivity correction (5.17) fulfills (5.16), i.e.,

$$
\min_{\mu \in \mathcal{M}_\lambda} \frac{1}{|V_\mu|} \int_{V_\mu} \hat{h}_\lambda^x(x) dx \geq 0.
$$
5.2. GRID ADAPTATION FOR SHALLOW WATER EQUATIONS

Proof: First, we note that for the mean values defined in (5.4) there holds

\[ \int_{V_\lambda} u^\ell_\lambda(x)dx = \bar{u}^\ell_\lambda |V_\lambda| = \int_{V_\lambda} \bar{u}^\ell_\lambda dx. \]

For that reason, we conclude (i) by integrating (5.17).

In order to verify (ii), we first note that

\[ \bar{h}^\ell_\lambda = \sum_{\mu \in \mathcal{M}_\lambda} \frac{|V_\mu|}{|V_\lambda|} \left( \frac{1}{|V_\mu|} \int_{V_\mu} h^\ell_\lambda(x)dx \right). \]

Thereby, we conclude

\[ \bar{h}^\ell_\lambda \geq \min_{\mu \in \mathcal{M}_\lambda} \frac{1}{|V_\mu|} \int_{V_\mu} h^\ell_\lambda(x)dx. \] (5.20)

For that reason, we infer from \( \min_{\mu \in \mathcal{M}_\lambda} \frac{1}{|V_\mu|} \int_{V_\mu} h^\ell_\lambda(x)dx \geq 0 \) that

\[ \frac{\bar{h}^\ell_\lambda}{\bar{h}^\ell_\lambda - \min_{\mu \in \mathcal{M}_\lambda} \frac{1}{|V_\mu|} \int_{V_\mu} h^\ell_\lambda(x)dx} \geq 1. \]

Thus, \( \alpha_\lambda = 1 \). Finally, we deduce \( \bar{u}^\ell_\lambda = u^\ell_\lambda \) by (5.17).

In order to verify (iii), we distinguish two cases. First, we consider the situation where

\( \min_{\mu \in \mathcal{M}_\lambda} \frac{1}{|V_\mu|} \int_{V_\mu} h^\ell_\lambda(x)dx < 0 \). Here, we conclude

\[ \alpha_\lambda = \frac{\bar{h}^\ell_\lambda}{\bar{h}^\ell_\lambda - \min_{\mu \in \mathcal{M}_\lambda} \frac{1}{|V_\mu|} \int_{V_\mu} h^\ell_\lambda(x)dx} < 1. \]

By integrating (5.17) we obtain

\[ \frac{1}{|V_\mu|} \int_{V_\mu} \hat{h}^\ell_\lambda(x)dx = \bar{h}^\ell_\lambda - \bar{h}^\ell_\lambda - \frac{1}{|V_\mu|} \int_{V_\mu} h^\ell_\lambda(x)dx \]

Due to (5.20), we conclude that

\[ \frac{\bar{h}^\ell_\lambda - \frac{1}{|V_\mu|} \int_{V_\mu} h^\ell_\lambda(x)dx}{\bar{h}^\ell_\lambda - \min_{\mu \in \mathcal{M}_\lambda} \frac{1}{|V_\mu|} \int_{V_\mu} h^\ell_\lambda(x)dx} \leq 1. \]

Thus, we obtain (5.19).
For the case \( \min_{\mu \in \mathcal{M}_\lambda} \frac{1}{W_\mu} \int_{V_\mu} h_\lambda^\mu(x)dx \geq 0 \), we conclude by (ii) that (5.19) holds true. □

When performing the prolongation, we apply the limiter (5.17) before each local inverse two-scale transformation with Algorithm 3.6 and 3.7, respectively. This ensures that the local mean values on the refined grid are non-negative. Note that it is nevertheless possible that some of the refined cells become dry. Thereby, non-physical negative water heights in the mean on these cells are avoided by using correction (5.17). The resulting algorithm is presented in Algorithm 5.2, in which the modified solution corresponding to the refined grid is denoted by \( \tilde{u}_{L,\varepsilon}^n = (\tilde{h}_{L,\varepsilon}^n, (\tilde{h} v)^{n}_{L,\varepsilon})^t \).

Finally, we prove that applying the modification (5.17) during the prolongation ensures local non-negativity in the mean.

**Conclusion 5.2 (Positivity in the mean during refinement)**

Let \( u_{L,\varepsilon}^n \) be the DG solution before refinement and let \( \tilde{u}_{L,\varepsilon}^n \) be the modified solution of \( u_{L,\varepsilon}^n \) on the refined grid using Algorithm 5.2. Then, the following statements are valid.

(i) The correction (5.17) does not modify the local contribution in wet areas, i.e., for \( \lambda \in \mathcal{I}^n \) there holds

\[
 h_{L,\varepsilon}^n(x) > 0 \quad \forall x \in V_\lambda \quad \Rightarrow \quad u_{L,\varepsilon}^n(x) = \tilde{u}_{L,\varepsilon}^n(x) \quad \forall x \in V_\lambda.
\]

(ii) The mean remains unmodified by the correction, i.e.,

\[
 \int_\Omega u_{L,\varepsilon}^n(x)dx = \int_\Omega \tilde{u}_{L,\varepsilon}^n(x)dx.
\]

(iii) \( \tilde{h}_{L,\varepsilon}^n \) is locally non-negative in the mean, i.e.,

\[
 \int_{V_\lambda} \tilde{h}_{L,\varepsilon}^n(x)dx \geq 0, \quad \forall \lambda \in \mathcal{I}^{n+1}.
\]

**Proof:** In order to prove (i) we note that \( h_{L,\varepsilon}^n(x) \geq 0 \) for all \( x \in V_\lambda, \lambda \in \mathcal{I}^n \), implies that \( \int_V h_{L,\varepsilon}^n(x)dx \geq 0 \) for all subsets \( V \subset V_\lambda \). Thus, we conclude by Theorem 5.1(ii) that the limiter (5.17) in line 5 of the second step in Algorithm 5.2 does not modify \( \tilde{u}_{L,\varepsilon}^n \). Hence, statement (i) holds true. In order to verify (ii), we note that due to Theorem 5.1(i) each modification by (5.17) in Algorithm 5.2 is locally conservative. Furthermore, due to Conclusion 3.2, the subsequent prolongation via (5.15) is conservative as well. For that reason, statement (ii) holds true. Statement (iii) is a consequence of repeatedly applying Theorem 5.1(iii). □
Algorithm 5.2 Modified prolongation to the refined grid

Input: DG solution on the (coarser) grid $\mathcal{I}_k^n$, i.e.,

$$ u_{L,\varepsilon}^n = \sum_{\ell=0}^{L} \sum_{\lambda \in \mathcal{I}_\ell \cap \mathcal{I}_L} u_{\lambda}^n. $$

Step 1: initialize
1: for all $\ell = 0, \ldots, L - 1$ do
2: for all $\lambda \in \mathcal{I}_k^n \cap \mathcal{I}_\ell$ do
3: $\hat{u}_{\lambda}^n = u_{\lambda}^n,$
4: if $\lambda \notin \mathcal{D}^{n+1}$ then
5: insert $\lambda$ into $\mathcal{I}^{n+1},$ 
6: end if
7: end for
8: end for

Step 2: modified prolongation to the refined grid
1: for all $\ell = 0, \ldots, L - 1$ do
2: for all $\lambda \in \mathcal{D}^{n+1} \setminus \mathcal{D}^n \cap \mathcal{I}_\ell$ do
3: apply (5.17) in order to compute $\hat{u}_{\lambda}^{\ell,n}$ from $u_{\lambda}^{\ell,n}$
4: for all $\mu \in \mathcal{M}_\lambda$ do
5: compute $\hat{u}_{\mu}^{\ell+1,n}$ from $\hat{u}_{\lambda}^{\ell,n}$ using (5.15),
6: if $\mu \notin \mathcal{D}^{n+1}$ then
7: insert $\mu$ into $\mathcal{I}^{n+1},$
8: end if
9: end for
10: end for
11: end for

Output: modified DG solution on the (finer) grid $\mathcal{I}^{n+1}$, i.e.,

$$ \tilde{u}_{L,\varepsilon}^n = \sum_{\ell=0}^{L} \sum_{\lambda \in \mathcal{I}_\ell \cap \mathcal{I}_L} \hat{u}_{\lambda}^{\ell,n}. $$

For sake of generality we avoided the use of basis expansions in this section. In [80] the presented strategy and proofs are presented in terms of the local basis expansions in detail.
5.2.3 Limiting in the adaptive scheme

In order to improve the quality of the troubled cell indicator in the limiting process, the limiter described in Algorithm 5.1 in the adaptive scheme is only applied on cells $V_\lambda$, $\lambda \in \mathcal{I}_L$ as discussed in Chapter 4. For stability purposes it might be of interest to ensure that at the wet/dry interface the limiter in Algorithm (5.1) is applied in any case. For that reason, we propose an optional strategy that ensures that wet/dry fronts will be refined in the following section.

5.2.4 Tracking of the wet/dry front

In some cases there is a special interest in resolving the position of the wet/dry fronts as precise as possible. When not refining the wet/dry interface up to the finest refinement level, accuracy in the position of the front might be lost. Furthermore, a refinement of the front up to the finest level might be of interest for stability purposes when the limiter is only applied to cells in the finest refinement level.

For the aforementioned reasons, we present an optional strategy guaranteeing that existing wet/dry fronts will be refined up to the finest refinement level, cf. [80]. During the computation new wet/dry fronts might appear. For that reason, the strategy ensures that these will be refined step by step up to the finest refinement level.

At each time step $t_n$ wet/dry interfaces are characterized by the index set

$$\mathcal{I}_{\text{interface}}^n := \{ \lambda \in \mathcal{I}_L^n : h_{\lambda,\max} > \delta_{\text{dry}} \text{ and } h_{\lambda,\min} < \delta_{\text{dry}} \},$$

(5.21)

where the local maximum water height is defined as

$$h_{\lambda,\max} := \max \left( \bar{h}_{\lambda}^n, \max_{x_i \in \Lambda_{\lambda}} h_{L_{\mu},L_{\mu}}(x_i), \max_{\lambda' \in \mathcal{V}_{\lambda}} \bar{h}_{\lambda'}^n \right).$$

The minimum value is defined analogously. Following (5.21), we define the index set

$$\mathcal{D}_{\text{interface},1}^n := \left\{ \lambda \in \bigcup_{\ell=0}^{L-1} \mathcal{I}_\ell^n : \exists \mu \in \mathcal{I}_{\text{interface}}^n \text{ such that } V_\mu \not\subseteq V_\lambda \right\},$$

characterizing the wet/dry interface (5.21) by means of 'significant' contributions from the orthogonal complement spaces. Based on the ideas of the prediction strategy in Chapter 4.3, we account for the fact that the interface might move during one time step. For that purpose, we add indices of the neighboring cells analogously to (4.23), i.e.,

$$\mathcal{D}_{\text{interface},2}^n := \mathcal{D}_{\text{interface},1}^n \cup \left\{ \tilde{\lambda} : V_{\tilde{\lambda}} \text{ is a neighbour of } V_\lambda, \lambda \in \mathcal{D}_{\text{interface},1}^n \right\}.$$

In addition, new wet/dry interfaces might appear during the computation. For that reason, we refine the corresponding cell if not yet resolved on the finest level. To ensure this, we inflate this index set by

$$\mathcal{D}_{\text{interface}}^n := \mathcal{D}_{\text{interface},2}^n \cup (\mathcal{I}_{\text{interface}}^n \setminus \mathcal{I}_L^n).$$
5.3 Summary of the adaptive scheme

Finally, we summarize our adaptive strategy. For that purpose, in Algorithm 5.3 the required steps for one time step of the adaptive scheme are presented. Therein, prediction and thresholding are carried out on the local contributions from the orthogonal complements in the multi-scale decomposition, while all other operations such as the time evolution are carried out on the local contributions from the (adaptive) DG spaces. If an enforced refinement of the wet/dry front is desired, then (5.22) has to be added to the refinement and the coarsening.

Note that if the time evolution of the DG scheme is well-balanced and preserves the positivity, the grid adaptation will maintain these properties during coarsening and refinement according to Proposition 5.1 and 5.2.

5.4 Numerical results

The adaptive scheme for the SWE presented in this section has been validated for numerous test cases in [38, 80, 120]. Here, we focus on two-dimensional applications and present numerical results for two different test cases. First, we consider an oscillating lake in a parabolic bowl. This is a challenging benchmark test case in which the issue of wetting and drying can lead to instabilities if positivity-preserving is not carefully addressed in the numerical scheme. Next, we apply our adaptive scheme for the simulation of a tsunami run-up onto a beach which is based on a scale experiment. Thus, experimental data for the validation of the results from the computations are available. In this section we consider Cartesian meshes with uniform dyadic subdivision of the cells. Here, we apply the classical approach (3.68) using multiwavelets with the corresponding choice of the local norm (3.77) for the characterization of significance.

5.4.1 Oscillating lake in a parabolic bowl

We consider a parabolic bowl on the squared domain $\Omega = [-2,2] \times [-2,2]$ with bottom topography $b(x) = \frac{x_1 + x_2}{10}$. The initial states are given by $u_0 = (h_0, h_0 v_0)^T$, where

$$h_0(x) = \max\left(0, \frac{1}{20} (2x_1 - 0.5) + 0.1 - b(x)\right), \quad v_0(x) = \left(\begin{array}{c} 0 \\ \frac{1}{2} \end{array}\right),$$
Algorithm 5.3 One time step of the adaptive scheme for SWE

Input: $u_{L,\varepsilon}^n$.

(i) **Grid refinement** (on $(\ell + b, \ell v)^t$):
    
    (a) locally apply the multi-scale transformation (3.79) to determine the multi-scale decomposition using Algorithm 3.2,
    
    (b) compute $D^{n+1}_z$ from $D^n_{\ell}$ by performing the prediction strategy according to (4.22), (4.23), (4.24), (4.25) and the additional constraint $D^0_{\ell} \subset D^n_{\ell}$,
    
    (c) determine the adaptive grid $T^{n+1}$ from $D^{n+1}$ and compute the corresponding single-scale coefficients by using Algorithm 3.9,
    
    (d) locally apply the positivity correction of the high-order part according to (5.6).

(ii) **Time evolution** (on $(\ell + b, \ell v)^t$):
    
    perform Runge-Kutta time evolution on the single-scale coefficients using (2.37), thereby after each stage the limiter described in Algorithm 5.1 is applied to all cells of the adaptive grid on the finest level, i.e., $\lambda \in T^{n+1} \cap T_L$.

(iii) **Grid coarsening** (on $(\ell + b, \ell v)^t$):
    
    (a) locally apply the multi-scale transformation (3.79) to determine the multi-scale decomposition using Algorithm 3.2,
    
    (b) compute $D^{n+1}_{\ell}$ by performing thresholding according to (4.5) and (3.40) with the additional constraint $D^0_{\ell} \subset D^{n+1}_{\ell}$ using Algorithm 3.8,
    
    (c) determine the adaptive grid $T_{\ell}$ from $D^{n+1}_{\ell}$ and compute the corresponding single-scale coefficients by using Algorithm 3.9,
    
    (d) locally apply the positivity correction of the high-order part according to (5.6).

Output: $u_{L,\varepsilon}^{n+1}$.
with frequency $\omega = \sqrt{\frac{7}{5}}$. The water is circling around in the bowl and after one period the initial position is reached again. This test case was proposed by Gallardo et al. [77] with the objective to test positivity-preserving and limiting of their numerical scheme. Since the boundary remains dry during the computation we can choose any boundary conditions; for simplicity we use reflecting boundary conditions. For this test case the analytical solution $\mathbf{u}(t, \mathbf{x}) = (h, hv)^T(t, \mathbf{x})$ is given by

$$h(t, \mathbf{x}) = \max \left( 0, \frac{1}{20} \left( 2x_1 \cos(\omega t) + 2x_2 \sin(\omega t) - 0.5 \right) + 0.1 - b(x) \right),$$

$$v(t, \mathbf{x}) = \left( -\frac{\omega}{2} \sin(\omega t), \frac{\omega}{2} \cos(\omega t) \right).$$

We apply a third-order scheme for the computations, i.e., $p = 3$ and an explicit third-order SSP-RK method with three stages for time discretization with a CFL number of 0.05. On level 0 the grid consists of $4 \times 4$ cells. The Shu constant in the troubled cell indicator for the limiter is chosen as $M = 0.1$. We perform computations with different maximum refinement levels $2 \leq L \leq 8$. In order to ensure convergence of the scheme, the dry tolerance $\delta_{\text{dry}}$ in the wetting and drying is chosen dependent on the number of refinement levels $L$ as $\delta_{\text{dry}}(L) = \overline{C} h_L^2$, where $\overline{C}$ is determined such that $\delta_{\text{dry}}(2) = 10^{-4}$ for the coarsest level $L = 2$.

Since the limiter is applied at partially wet cells in any case, we cannot expect the scheme to be locally high-order accurate in these cells. For that reason, it is sufficient that $\delta_{\text{dry}}(L)$ decays only with second order in $h_L$. In fact, we can only guarantee that the scheme converges with first order accuracy in these cells. Hence, we use the heuristic strategy with $\beta = 1$ and $C = 1$ for the choice of the threshold parameter, i.e., $\varepsilon_L = h_L$. However, we gain from the third-order scheme by a more accurate approximation of the bed. Here, the precision in which the wet/dry front is resolved strongly influences the accuracy of the numerical scheme. For that reason, we enforce its refinement up to the finest level using the strategy proposed in Section 5.2.4.

For the different levels of refinement we have computed adaptive solutions up to $T = 4.5$, such that one period is completed. In order to investigate the efficiency of the adaptive scheme, we have additionally computed uniform solutions with the reference scheme. Exemplarily, we present the adaptive solution and the corresponding grid for $L = 8$ at different instants of time in Figure 5.2. These pictures demonstrate that the refinement of the adaptive scheme tracks the wet-dry front, whereas in all other regions in which the solution is smooth the grid is relatively coarse.

The objective of the adaptive strategy is to maintain the accuracy of the reference scheme. For that reason, we compare the $L_1$-error in the water height of the adaptive scheme with the $L_1$-error in the water height of the reference scheme for different refinement levels in Figure 5.3(a). Thereby, we observe that accuracy of the reference scheme is not only maintained in the adaptive scheme, but the error of the adaptive scheme is
Figure 5.2: Oscillating lake \((L = 8)\): adaptive solution and grids at different times \(t\). Shape: water surface \(h + b\). Contour: wet regions in blue and \(b\) in brown. Underneath the adaptive grid is shown.
5.4. NUMERICAL RESULTS

smaller than the error of the reference scheme. This surprising result might be explained by the fact that grid adaptation can be understood as a filter: small perturbations in the solution are discarded when the grid is coarse. Nevertheless, we note that the adaptive and the reference scheme show the same asymptotic behavior.

In order to estimate the gain of the adaptive scheme, we compare the maximum number of cells according to (4.38) of the adaptive computation with the reference scheme in Figure 5.3(b). Thereby, we observe that in the adaptive scheme significantly less cells are needed to obtain the accuracy of the reference scheme.

![Graphs showing comparison between reference scheme and adaptive scheme](image)

Figure 5.3: Oscillating lake in a parabolic bowl: accuracy and efficiency by means of (4.38) of the adaptive scheme ($L_1$-error in $h$).

5.4.2 Tsunami run-up on a complex three-dimensional beach

In 1993 an off-shore earthquake in the Sea of Japan caused a tsunami wave. Amongst others the tsunami wave reached Okushiri Island in the prefecture Hokkaido and caused serious damages. Near the village of Monai a small offshore island named Muen Island caused an unexpected high run-up of 31.7$m$ in a valley at the coast behind the island, whereas the run-up heights on the beaches directly exposed to the tsunami wave were only around 25$m$.

In order to investigate this phenomenon the Central Research Institute for Electric Power Industry (CRIEPI) in Abiko, Japan performed 1 : 400 scale experiments of the run-up in this area using a tank of 205$m$ length and 3.4$m$ width [127]. The coastal region of the experimental setup is shown in Figure 5.4(a). From the experiments measurement data are available [169]. In the experiments the position of the highest run-up has been measured. Furthermore the time evolution of the water surface $h + b$ at three control points between Muen Island and the shore has been recorded. The locations of these control points are depicted in Figure 5.5. Due to the availability of these measurement data, this scale experiment is well-suited for a validation of a numerical scheme.
Figure 5.4: Tsunami run-up: bottom topography and water surface $h + b$.

Figure 5.5: Tsunami run-up: location of the control points (marked in blue) on the bottom topography.

Figure 5.6: Tsunami run-up: time evolution of the water surface for the inflow wave at $x = 0$.

For the computation we focus on the scale experiment instead of the real tsunami event. Thereby, we focus only on the coastal area since the inflow wave has been recorded near the coast. Thus, the simulation of the very long channel can be avoided and the computational domain covers only an area of the size $5.488 \text{ m} \times 3.402 \text{ m}$, i.e., $\Omega = [0, 5.488] \times [0, 3.402]$. At $x = 0$ the tsunami wave is entering the computational domain. The recorded water surface $h + b$ of the inflow wave at $x = 0$ is provided by [169]; for details see Figure 5.6. From these data we compute the momentum $h \mathbf{v}$ at $x = 0$ by using Riemann invariants such that $(h, h \mathbf{v})$ corresponds to a simple incoming wave [87]. These values are prescribed at the inflow boundary $x = 0$ using the characteristic boundary conditions (2.22). Coinciding with the experiment, we consider reflecting wall boundary conditions at all other boundaries. In the domain we initially consider a lake at rest, i.e., $h + b = 0$ and $\mathbf{v} = 0$. In Figure 5.4(b) the profiles of $b$ and the initial surface $h + b$ in the direction normal to the beach at $y = 2$ are shown.

Triggered by the boundary conditions, the incoming wave is moving towards the shoreline. Thereby, the water is first retracting from the shoreline before the following tsunami
wave is eventually flooding the dried up regions and the coastal area. Afterwards the reflecting wave at the shoreline causes wave structures moving away from the coast.

We compute numerical solutions with a third-order scheme, i.e., piecewise quadratic polynomials and the third-order SSP-RK scheme with three stages. We consider $8 \times 5$ cells in the coarsest grid on level 0 and $5 \leq L \leq 6$ levels of refinement. On $\Omega$ the bottom topography is given by a discrete set of 97,857 points. From these discrete samples we determine via interpolation and thresholding an adaptive representation of the bottom topography consisting of piecewise quadratic polynomials. The dry tolerance in the wetting and drying strategy is chosen as $10^{-4}$. Since the position of the wet/dry front at the coast is of special interest in this test case, we enforce refinement of the wet/dry front using the strategy proposed in Section 5.2.4. The threshold parameter is chosen using the heuristic strategy as $\varepsilon_L = C_{\text{thr}} h_L$. Here, the amplitude of the inflow wave is in the order of $10^{-2}$, whereas the quantity $h + b$ is in the order of 1, see Figure 5.6. Thus, the wave is 100 times smaller than the order of magnitude of $h + b$. In order to resolve the wave structures caused by the tsunami, we have chosen the factor $C_{\text{thr}}$ as 1/100.

In Figures 5.8 and 5.9 the numerical solutions and the corresponding adaptive grids using $L = 6$ are presented at different time instances. In these pictures the dynamical grid adaptation can be observed: the grid refinement is tracking the arising waves, whereas in regions with small variations the grid is coarse. For the $L = 6$ computation the adaptive grid consists of at most 18,139 cells, whereas the uniform grid of the reference scheme would consist of 163,840 cells. Thus, the computational cost can be significantly reduced using the grid adaptation.

In order to validate our adaptive scheme, we compare the numerical results with measurement data from the scale experiments. For that purpose, we first compare the water height at the three control points located between Muen Island and the shore. The specific locations of these points can be seen in Figure 5.5. In Figure 5.7 the water surface $h + b$ from the $L = 6$ computation and from the experiment are shown. Here, we observe a
Table 5.1: Tsunami run-up: comparison of maximum run-up height, its position and time of occurrence.

<table>
<thead>
<tr>
<th>Computation</th>
<th>maximum run-up</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>height [m]</td>
</tr>
<tr>
<td>$L = 5$</td>
<td>0.079435</td>
</tr>
<tr>
<td>$L = 6$</td>
<td>0.088415</td>
</tr>
<tr>
<td>GeoClaw [87]</td>
<td>0.09 -0.1</td>
</tr>
<tr>
<td>experimental data [127]</td>
<td>0.09</td>
</tr>
</tbody>
</table>

A slight difference between computational and experimental data. However, our results are similar to results computed with other numerical schemes, cf. [106, 121]. The differences might be caused by measurement inaccuracies in the experiments or modeling errors due to the use of SWE. However, the essential evolution of the surface $h + b$ is reproduced by the simulation.

Next, we focus on the maximum run-up of the tsunami wave. To this end, we have determined the maximum height, its position and time of occurrence in our adaptive computations. In Table 5.1 these are listed. Furthermore, the experimental data from [127] and the numerical results of the GeoClaw solver [87] are listed for comparison. When comparing our results with the experimental data and the results using GeoClaw, we conclude that our adaptive solver can identify the maximum run-up height and its position of occurrence very accurately. Moreover, we note that the $L = 6$ computation is more accurate than the one with $L = 5$. The time of occurrence of the maximum run-up in our computation is similar to the one from the GeoClaw solver provided in [87]. The instant of the highest run-up is shown in Figure 5.9(a).

Summarizing we infer from the validation, that our results are just as well accurate as the results available in the literature. However, due to the use of local grid adaptation, we are able to significantly reduce the computational complexity and cost in comparison to the typically considered uniform schemes.
Figure 5.8: Tsunami run-up: adaptive solutions ($L = 6$) and the corresponding computational grids at time $t = 0\, s$, $t = 13.5\, s$, $t = 14.3\, s$ and $t = 15\, s$. 
Figure 5.9: Tsunami run-up: adaptive solutions ($L = 6$) and the corresponding computational grids at time $t = 16.5\, s$, $t = 18\, s$, $t = 19.5\, s$ and $t = 22.5\, s$. 
Chapter 6

Conclusion

This thesis focuses on the development of adaptive multiresolution-based discontinuous Galerkin schemes for conservation laws. In [108,156] the basic concept has been introduced and investigated for one-dimensional scalar problems using uniform dyadic grid hierarchies. This setting significantly restricts the range of application. To overcome this limitation we have aimed at (i) developing a wavelet-free multiresolution analysis to deal with non-uniform grid hierarchies, (ii) deriving a reliable and efficient strategy to choose the threshold parameters, (iii) providing a strategy to deal with constraints of the underlying partial differential equations and (iv) validating the concept for non-linear systems of conservation laws in multiple space dimensions.

Wavelet-free approach The construction of a basis for the orthogonal complements on a non-uniform grid hierarchy is extremely costly because it has to be performed on each element instead of a few reference elements. Thus, the choice of the generators is a key aspect for an efficient realization of the multiresolution analysis. To overcome this issue we have developed a new approach avoiding the explicit construction of the bases for the orthogonal complements. Thus, this new approach has enabled a simpler implementation of the multiresolution analysis resulting in a significantly improved performance. Furthermore, the wavelet-free approach has allowed for a straightforward application to arbitrary non-uniform nested hierarchies. For instance, hierarchies composed of triangular or curved elements as well as mixed hierarchies consisting of elements with different shapes can be considered in the future.

Choice of the thresholds The adaptation procedure is solely specified by the local threshold parameters. The performance of the adaptive scheme is strongly influenced by their choice. For this purpose, we have developed a strategy that for many applications turned out to be robust, reliable and cost-efficient. In [108,156], the analysis of the scheme for one-dimensional scalar conservation laws provides an a priori choice of the thresholds, but numerous computations have revealed it being too pessimistic. Alternatively, we
have developed a heuristic strategy and have compared both the a priori and the heuristic strategy for the classical wavelet-based approach. It has turned out that with both alternatives the accuracy of the reference scheme is maintained, but using the heuristic approach is more efficient than using the a priori choice as long as the solution exhibits discontinuities. We have found that this holds true also for the wavelet-free approach. We emphasize that for both the heuristic and the a priori strategy no tuning and parameter fitting are required.

**Constraints** Many problems are subjected to constraints, e.g., some quantities have to be positive or divergence-free as well as discretizations have to be well-balanced. Typically uniform numerical schemes are modified to ensure that these constraints hold in a discrete setting. When considering an adaptive scheme these constraints are not automatically satisfied. To this end, we have developed a strategy to deal with constraints in the adaptive scheme. Exemplarily, we have considered the shallow water equations where the scheme has to be positivity-preserving and well-balanced. We have modified the adaptive scheme and proven that these constraints are maintained during grid adaptation. The strategy behind these modifications is not limited to the shallow water equations: whenever a quantity has to satisfy some bound, the herein presented strategy can be directly applied.

**Numerical validation** For the purpose of validating the concept, we have performed adaptive computations for several well-known benchmark test cases for the Burgers’ equation, the compressible Euler equations, the shallow water equations as well as the compressible Navier-Stokes equations. These have shown that the adaptive scheme is capable of accelerating the reference scheme significantly while maintaining its accuracy.

More precisely, these computations have confirmed that using the a priori strategy for the choice of the thresholds is too pessimistic, whereas the heuristic strategy results in coarser grids and, thus, is more efficient. Thereby, we have found that both strategies are asymptotically capable of maintaining the accuracy of the uniform reference scheme. Hence, it has turned out that the heuristic strategy is the preferable choice.

Furthermore, we have noted that the grids resulting from the classical wavelet-based approach are slightly coarser than using the wavelet-free approach. However, the additional refinement of the wavelet-free approach has turned out to be small in comparison to the overall gain from the grid adaptation. In general, we have found that the reduced computational cost for the wavelet-free approach compensates for the computational overhead resulting from the additional refinement and, thus, the wavelet-free approach is preferable.

Exemplarily, for non-uniform grid hierarchies we have applied the adaptive scheme to a shock vortex interaction with a boundary layer in a viscous shock tube. We have noted that the adaptive scheme is capable of resolving all relevant features in the solution while reducing the computational cost significantly.

To validate the modified adaptive scheme for the shallow water equations, we have
applied it to two well-known benchmark test cases. First we have considered an oscillating lake in a parabolic bowl, a challenging configuration with known analytical solution in which the issue of wetting and drying can lead to instabilities if positivity-preserving is not carefully addressed. Then, we have investigated the tsunami run-up onto a complex three-dimensional beach. For this configuration no analytical solution is known but measurement data from a scale experiment are available for validation. We have inferred from these validations that our scheme is capable of computing these configurations including wetting and drying fronts confirming that the adaptive scheme is well-balanced and positivity-preserving. In particular, our results have been as accurate as the results available in the literature but have been obtained at significantly reduced computational cost on locally refined grids in comparison to typically used uniform discretizations. The application of the adaptive multiresolution-based scheme for the shallow water equations to large-scale real-world scenarios has now been made possible.

Outlook

Besides employing the MRA for grid adaptation it can also be used for modeling purposes, e.g., in the context of turbulent flows as will be discussed in the following. Performing direct numerical simulations of turbulent flows is extremely costly and only feasible for small problems. Thus, one is interested in reducing the computational complexity of the simulation of turbulent flows. However, applying the adaptive scheme it cannot be guaranteed that the turbulent flow features are maintained for both presented choices for the thresholds. In fact, applying the adaptive scheme to the fully-developed turbulent channel flow from Section 2.6.3 results in a laminar flow. For that reason a different strategy to reduce the computational complexity is required.

To this end, future developments include the combination of the multi-scale setting with turbulence modelling which plays a central role in the investigation of turbulent phenomena. One recent approach is the variational multi-scale (VMS) method introduced by Hughes et al. [111–113]. In this method a separation of resolved and unresolved scales is obtained by projections in the variational formulation. To derive an adaptive VMS-DG method, the multi-scale decomposition of the turbulent flow can be employed to decide where to model the turbulence instead of resolving all turbulent features.

Exemplarily, a multi-scale decomposition of the instantaneous momentum in \( x_1 \)-direction of the fully-developed turbulent channel flow from Section 2.6.3 is shown in Figure 6.1, where the turbulent flow is decomposed into contributions from three refinement levels, i.e., \( \mathbf{u} = \mathbf{u}^0 + \mathbf{d}^0 + \mathbf{d}^1 \). We note that \( \mathbf{u}^0 \), i.e., the projection to level 0, is sufficient to represent the substantial features of the turbulent flow. However, the fluctuations \( \mathbf{d}^0 \) and \( \mathbf{d}^1 \) are essential in the DNS to maintain turbulence because the flow becomes laminar when performing a naive local thresholding of fluctuations.

These observations motivate to locally coarsen the grid and locally model the influence
of the fluctuations instead of resolving them all, cf. [62]. This can be realized using the local contributions from the orthogonal complements in the multi-scale decomposition to characterize the fluctuations. Then the key question is how the fluctuations influence large scale effects characterized by projections to coarser levels. To this end, it is important to investigate the mapping of the fluctuations from the orthogonal complements with the nonlinear fluxes from the compressible Navier-Stokes equations, cf. [53, 54]. Thereby a MRA-based turbulence model can be derived resulting in adaptive variational multi-scale method.

![Multi-scale decomposition](image)

**Figure 6.1**: Multi-scale decomposition of the instantaneous momentum in $x_1$-direction of the fully-developed turbulent channel flow.

Since turbulent flow simulations are very costly and the computational time is typically in the order of days or even weeks, it is *indispensable* that a multi-scale decomposition can be realized very efficiently. Turbulent flows typically require stretched grids. Therefore the construction of wavelets has to be performed on several reference elements and, thus, should be avoided. In addition, the need of a high-order scheme results in a huge number of degrees of freedom and, consequently, the mask matrices $M_{\lambda,0}$ and $M_{\lambda,1}$ become very large. For instance, in the fifth-order scheme considered for the turbulent channel flow $M_{\lambda,0}$ and $M_{\lambda,1}$ have 125,000 and 875,000 entries, respectively. Consequently, the realization of a local two-scale transformation on a *single cell* requires 3,998,000 operations (multiplications and additions) for the classical wavelet-based approach and only 749,875 operations for the wavelet-free approach. In particular, for rectangular grids the number of matrix entries to be stored can be reduced significantly using a tensor-product basis, i.e., for the wavelet-free approach only 150 matrix entries and for the classical wavelet-based approach 300 matrix entries have to be stored. Obviously, the use of the wavelet-free
The wavelet-free approach reduces the memory requirements and the computational cost. Although the cost is only reduced by a factor of about 5, the computational time is reduced from several weeks to a few days. Moreover, the memory requirements are reduced by a factor of 8 and 2 for a general and a tensor-product basis, respectively. Thus, the use of the wavelet-free approach is substantial for the efficient realization of a multiresolution analysis in the context of turbulent flows.
Bibliography


[164] B. Sjögreen and H. C. Yee. Grid convergence of high order methods for multiscale


[166] G.A. Sod. A survey of several finite difference methods for systems of nonlinear

Institute of Technology, 1996.


[169] The third international workshop on long-wave runup models.

[170] K.W. Thompson. Time-dependent boundary conditions for hyperbolic systems, II.

[171] T. von Kármán. The analogy between fluid friction and heat transfer. Trans. ASME
61, pages 705–710, 1939.


7661, 2009.

2000.

[175] Y.S. Weber, E.S. Oran, J.P. Boris, and J.D. Anderson. The numerical simulation of
shock bifurcation near the end wall of a shock tube. Phys. Fluids, 7(10):2475–2488,
1995.

[176] L. Wei. Direct Numerical Simulation of Compressible and Incompressible Wall
Bounded Turbulent Flows with Pressure Gradients. PhD thesis, Queen’s University


