Whitney Elements on Sparse Grids

Dissertation

zur Erlangung des Grades
eines Doktors der Naturwissenschaften
der Mathematischen Fakultät
der Eberhard-Karls-Universität Tübingen

vorgelegt von
Vasile Catrinel Grădinaru

Tübingen, April 2002

Tag der mündlichen Prüfung:	12. Juli 2002

Dekan:	Prof. Dr. Wolfgang Knapp
1. Berichterstatter:	Prof. Dr. Harry Yserentant
2. Berichterstatter:	Prof. Dr. Ralf Hiptmair

To Katharina

Acknowledgment

First of all, I thank my supervisor Prof. H. Yserentant for his valuable direction. My best thanks go to Prof. R. Hiptmair for his support, encouragement and fruitful discussions.

I thank the German Research Foundation (DFG), Sonderforschungsbereich 382 “Verfahren und Algorithmen zur Simulation physikalischer Prozesse auf Höchstleistungsrechnern” and its head Prof. H. Ruder for the financial support. I am also grateful to all my colleagues at the Numerical Analysis Group at the Universität Tübingen for a climate conductive to creative work.

A lot of people influenced my way of doing mathematics, starting with my father and my high school teacher of mathematics. Prof. Gh. Morosanu helped me in the first steps in the scientific research. Prof. V. Arnăutu revealed to me the beauty of the numerical analysis. His art to explain made me choose numerical analysis as field of research. Last, but not least, I thank Prof. R.H.W. Hoppe for giving me the chance to join his research group at Universität Augsburg. It is there that I met R. Hiptmair and that I had the idea for the subject of this thesis.

Finally, I thank my wife, Adi, for her patience and support.

Abstract

The aim of this work is to generalize the idea of the discretizations on sparse grids to discrete differential forms. The extension to general l-forms in d dimensions includes the well known Whitney elements, as well as $\boldsymbol{H}(\mathrm{div};\Omega)$- and $\boldsymbol{H}(\mathbf{curl};\Omega)$-conforming mixed finite elements.

The formulation of Maxwell's equations in terms of differential forms gives a crucial hint how they should be discretized. The focus is on discrete differential forms of lowest order, i.e. Whitney elements. Taking the cue from Lagrangian finite elements on sparse grids, we present the hierarchical decomposition of the Whitney spaces. The tensor product structure and the hierarchical multilevel principle give rise to hierarchical basis for Whitney l-forms in d dimensions.

Relying on the hierarchical basis, we define the sparse grid interpolation operator and we prove the commuting diagram property as well as the existence of discrete potentials in sparse grid spaces. The interpolation estimates generalize the known results for Lagrangian finite elements. Approximate interpolation is needed for the Galerkin method for boundary value problems on sparse grids. The combination technique and a two point quadrature rule ensure that similar error estimate as for the exact interpolation hold.

Discrete inf-sup conditions are shown theoretically and experimentally for mixed second order problems. The focus is on the stability of the discretization of the primal and of the dual mixed problem by sparse grid Whitney forms. The existence of stable potentials is a sufficient condition. We prove it in particular cases, completely covering the three dimensional case. Numerical results give evidence for $d = 4$, too. The results show that discrete differential forms on sparse grids give rise to viable numerical schemes for the discretization of both $H(\mathbf{d},\Omega)$-elliptic variational problems and second order mixed problems.

The algorithms involved are presented both in a general form and with particular design solutions. They mirror the two pervasive ideas in the theory of the sparse grids, namely the hierarchical decomposition and the reduction to the one dimensional case via tensor product. We give examples for smooth and no-smooth forms, where we compute numerically the interpolation error on the full grid. Algorithmic details are given for the the multiplication with the mass and the stiffness matrix. The last chapter is confined to the the multigrid scheme and the needed automatic construction of stencils on anisotropic full grids.

Zusammenfassung

Das Ziel der vorliegende Arbeit ist, Dünngitter-Verfahren für diskrete Differentialformen zu verallgemeinern. Die Erweiterung auf l-Formen in d Dimensionen schließt die bekannten Whitney Elemente und die $\boldsymbol{H}(\mathrm{div};\Omega)$- und $\boldsymbol{H}(\mathbf{curl};\Omega)$-konformen gemischten Finite Elemente ein.

Die Darstellung der Maxwellgleichungen im Differentialformenkalkül fuhrt auf geeignete Diskretisierung. Es werden nur diskrete Differentialformen niedrigster Ordnung, dass heißt Whitney-Formen, behandelt. Die hierarchischen Basisfunktionen für Whitney-Formen lassen sich mit der Multiskalentechnik unter Verwendung eines Tensorproduktansatzes herleiten.

Der Dünngitter-Interpolationsoperator wird auf Grundlage der hierarchischen Basis definiert. Die kommutierende-Diagram-Eigenschaft und die Existenz diskreter Potentiale werden im Rahmen der Dünngitter-Diskretisierung bewiesen. Die Abschätzung des Interpolationsfehlers entspricht im Spezialfall $l = 0$ den bekannten Ergebnissen für Lagrange Finite Elemente.

Im Rahmen eines Galerkin-Ansatzes für die Lösung elliptischer Differentialgleichungen muss man eine Approximation des Interpolationsoperators benutzen. Die Kombinationsmethode und die Trapezregel mit zwei Punkten liefern die algorithmische Realisierung des approximierten Interpolationsoperators. Der Interpolationsfehler behält dieselbe Ordnung wie im Fall des "exakten" Operator.

Diskrete inf-sup–Bedingungen für gemischte Formulierungen elliptischer Probleme zweiter Ordnung werden sowohl theoretisch als auch numerisch untersucht. Der Schwerpunkt liegt auf der Stabilität der Dünngitter-Diskretisierung des primalen und des dualen Problems mit Whitney-Formen. Die Existenz stabiler Potentiale ist eine hinreichende Bedingung, die in verschiedenen Einzelfällen bewiesen wird. Der praktisch interessante drei dimensionale Fall wird vollständig untersucht. Numerische Ergebnisse liefern Hinweise auf Stabilität im Falle $d = 4$ und lassen uns eine allgemeinere Aussage vermuten. Die Ergebnisse zeigen, dass diskrete Differentialformen streng begründbare Berechnungsverfahren sowohl für $H(\mathbf{d},\Omega)$-elliptische als auch für gemischte Probleme zweiter Ordnung zur Verfügung stellen.

Die algorithmischen Aspekte werden ausführlich dargestellt, sowohl in einer allgemeinen Form, als auch mit konkreten Implementierungslösungen. Die Algorithmen spiegeln die zwei grundliegenden Ideen in der Dünngitter-Theorie wieder: die hierarchische Zerlegung und die Zurückführung auf den eindimensionalen Fall mit Hilfe eines Tensorproduktansatzes. Die hierarchischen Transformationen verdeutlichen dieses unidirektionale Prinzip. Beispiele für die Interpolation glatter und nicht-glatter Differentialformen werden betrachtet. Desweiteren werden die Algorithmen für die Steifigkeits- und Massenmatrix Multiplikationen, die automatische Konstruktion der Operatorsterne auf anisotropen vollen Gitter und das Mehrgitterverfahren detailiert beschrieben.

Contents

List of notations

Symbol		Page
$\mathbf{d}$	:= Exterior derivative operator	7
$\mathbf{H}(\mathbf{d},\Omega)$	$:= \{l-\text{forms } \omega,\ \Vert\omega\Vert_{L^2(\Omega)} + \Vert\mathbf{d}\omega\Vert_{L^2(\Omega)} < \infty\}$	8
$\Vert\omega\Vert^2_{\mathbf{H}(\mathbf{d},\Omega)}$	$:= \Vert\omega\Vert^2_{L^2(\Omega)} + \Vert\mathbf{d}\omega\Vert^2_{L^2(\Omega)}$	8
$\mathbf{H}_0(\mathbf{d},\Omega)$	$:= \{\omega \in \mathbf{H}(\mathbf{d},\Omega),\ \omega_{\vert\partial\Omega} = 0\}$	8
$\boldsymbol{H}_\Gamma(\mathbf{curl};\Omega)$	$:= \{\mathbf{u} \in \boldsymbol{H}(\mathbf{curl};\Omega),\ \mathbf{u}\times\mathbf{n} = 0 \text{ on } \Gamma\}$	9
$\boldsymbol{H}_\Gamma(\mathrm{div};\Omega)$	$:= \{\mathbf{u} \in \boldsymbol{H}(\mathrm{div};\Omega),\ \mathbf{u}\cdot\mathbf{n} = 0 \text{ on } \Gamma\}$	9
φ	:= Hat function	17
$\varphi_{\nu i}$	:= Scaled and translated hat functions	17
$\bar{\varphi}_{\nu i}$	:= Scaled, translated and normed hat functions	17
B	:= Box function	19
$B_{\nu k}$	:= Scaled, translated and normed box functions	19
ψ	:= Normed Haar wavelet	19
$\psi_{\nu\,k}$	:= Scaled, translated and normed Haar wavelets	19
Ω	$:= [0,1]^d$	22
$\boldsymbol{\nu}$	$:= (\nu_1,\ldots,\nu_d) \in \mathbb{N}^d$	22
$\mathbf{h}_{\boldsymbol{\nu}}$	$:= (h_{\nu_1},\ldots,h_{\nu_d}) = (2^{-\nu_1},\ldots,2^{-\nu_d})$	22
$x_{\boldsymbol{\nu}\mathbf{q}}$	$:= (q_1 h_{\nu_1},\ldots,q_d h_{\nu_d})$	22
$\boldsymbol{\mathcal{I}}_{\boldsymbol{\nu}}$	:= Nodal interpolation operator	23
$\mathcal{I}^I_{\boldsymbol{\nu}}$	:= Nodal interpolation operator for the I-component	23
$V^l_{\boldsymbol{\nu}}$	:= Space of the Whitney l-forms on the anisotropic full grid $\Omega_{\boldsymbol{\nu}}$ of resolution $\boldsymbol{\nu}$	23
Φ^I_F	$:= \prod_{i\in I} B_{\nu_i q_i} \prod_{j\notin I} \varphi_{\nu_j q_j}$ nodal basis function	23
$\boldsymbol{\Theta}_{\boldsymbol{\nu}}$	$:= \{\mathbf{q} = (q_1,\ldots,q_d),\ q_i = 1,3,\ldots,\ 2^{\nu_i}-1 \text{ if } \nu_i > 0,\ \text{and } q_i = 0,1 \text{ if } \nu_i = 0\}$	25
D^I	$:= \prod_{i\in I}\frac{\partial}{\partial x_i}\prod_{j\notin I}\frac{\partial^2}{\partial x_i^2} = \frac{\partial^{l+2(d-l)}}{\partial x_{i_1}\ldots\partial x_{i_l}\partial x^2_{j_1}\ldots\partial x^2_{j_{d-l}}}$	29
$\Psi^I_{\boldsymbol{\nu}\mathbf{q}}$	$:= \prod_{i\in I}\psi_{\nu_i q_i}\prod_{j\notin I}\varphi_{\nu_j q_j}$ hierarchical basis functions	29
$\mathcal{H}^I_{\boldsymbol{\nu}} u_I$	$:= \sum_{\mathbf{q}\in\boldsymbol{\Theta}_{\boldsymbol{\nu}}} \beta_{\boldsymbol{\nu}\mathbf{q}}\Psi^I_{\boldsymbol{\nu}\mathbf{q}}$ hierarchical surplus corresponding to u_I	29
$W^I_{\boldsymbol{\nu}}$	:= Range($\mathcal{H}^I_{\boldsymbol{\nu}}$) detail space	29
$\mathcal{H}_{\boldsymbol{\nu}}$	$:= \sum_I \left(\mathcal{H}^I_{\boldsymbol{\nu}} u_I\right) dx_I$ hierarchical interpolation operator	29

Symbol		Page
$\lvert\boldsymbol{\nu}\rvert_1$	$:= \sum_i \nu_i$	29
V_n^I	$:= \bigoplus_{\lvert\boldsymbol{\nu}\rvert_1 \le n+d-1} W_{\boldsymbol{\nu}}^I$	34
$\mathcal{I}_n^I$	$:= \sum_{\lvert\boldsymbol{\nu}\rvert_1 \le n+d-1} \mathcal{H}_{\boldsymbol{\nu}}^I$ sparse grid interpolation operator for the I-component	34
$\mathcal{I}_n$	$:= \sum_{\lvert\boldsymbol{\nu}\rvert_1 \le n+d-1} \mathcal{H}_{\boldsymbol{\nu}}$ sparse grid interpolation operator	34
V_n^l	$:=$ Range($\mathcal{I}_n$) space of Whitney-l-forms on a sparse grid of resolution n in d dimensions	34
$\tilde{\mathcal{H}}_{\boldsymbol{\nu}}^I$	$:= \underset{i\in I}{\circ} \tilde{\mathcal{H}}_{\nu_i} \underset{j\notin I}{\circ} \mathcal{H}_{\nu_j} := \underset{i\in I}{\circ} (\tilde{\mathcal{I}}^{\text{const}}_{\nu_i} - \tilde{\mathcal{I}}^{\text{const}}_{\nu_i-1}) \underset{j\notin I}{\circ} (\mathcal{I}^{\text{lin}}_{\nu_j} - \mathcal{I}^{\text{lin}}_{\nu_j-1})$ approximate hierarchical surplus operator	45
$\vec{\omega}$	$:= (\beta^I_{\boldsymbol{\nu}\mathbf{q}})_{\boldsymbol{\nu},\mathbf{q},I}$ vector containing the hierarchical coeffcients	74
$\mathcal{S}_n$	$:=$ Space of coefficient vectors corresponding to the sparse grid of resolution n	76
$\mathcal{A}_{\boldsymbol{\nu}}$	$:=$ Space of coefficient vectors corresponding to the anisotropic full grid of resolution $\boldsymbol{\nu}$	76

Chapter 1

Introduction

For a successful numerical simulation, reality must be modelled correctly, for Nature cannot be fooled. Hence, reality is an imperative in the discretization of the physical laws. But how much of the continuous model can we preserve in the discrete one? Consider Maxwell's equations, which offer an universal theory. Yet, concrete problems need practical solutions, so we must reduce the general model to an adequate and solvable one. Attention has to be paid to avoiding undue simplifications. It took a lot of time to refute the idea that one can cope with electromagnetics once one knows to solve Laplace equation numerically [48]. Now we know that (discrete) differential forms afford a discrete model better than conventional Lagrangian finite elements. Another source of error in the solution is the discretization level. We can rarely compute exactly the continuous solution by means of analytical methods. What we usually do, is to increase the resolution of the discretization, until the numerical solution satisfies some a-priori criteria. In this point, we encounter the curse of the dimensionality: for problems posed in spaces of dimension $d \geq 3$, we cannot manage the amount of data, if we resort to standard discretization strategy. Only through highly efficient methods it becomes possible to obtain in a reasonable time a satisfactory solution by means of a storable data set.

Sparse grids, sometimes called "hyperbolic cross approximations", have been introduced to provide efficient approximations of smooth functions [23, 92]. More precisely, they provide a device to describe a function up to a prescribed accuracy with very few degrees of freedom. If conventional Sobolev-norms are used to measure accuracy, sparse grids can be shown to be optimal or near optimal. In any case a tremendous reduction of the amount of data is achieved compared to standard schemes of multivariate approximations that rely on low-order polynomials. For an introduction into the fundamentals of approximation on sparse grids the reader is referred to [23, 64, 92]. Here, we only mention that the crucial idea is to drop certain insignificant contributions of hierarchical representations of functions. In this respect there is a close link between sparse grids and wavelet schemes (cf. [26, 75]).

The classical sparse grid approximation is based on piecewise linear continuous functions with respect to special meshes of a tensor product structure. This immediately gives rise to $H^1(\Omega)$-conforming finite element spaces and renders sparse grids useful for devising generalized finite element discretizations of boundary value problems. The history of the sparse grids is long, and I mention here only that Ch. Zenger's ideas [92] catalyzed the research in this domain. Provided that the solutions are smooth, a tremendously reduced number of unknowns yields almost the same asymptotic rates of approximation as for standard grids.

Meanwhile, there are plenty of applications of sparse grids in the discretization of partial differential equations: Poisson's equation [20, 21, 70], the Helmholtz equation [8], parabolic problems [7], general linear elliptic operator of second order in two dimensions [30] were treated. Adaptive sparse grid techniques were investigated, too [20, 38, 43, 72]. Research has also focused on the parallelization [36, 62] of sparse grid finite element methods and on the efficient multilevel solution of the resulting linear systems [38, 39, 65].

Thus far, the focus in all these works has been on $H^1(\Omega)$-conforming approximations. However, the scope of finite elements reaches far beyond $H^1(\Omega)$. Prominent examples are so-called mixed schemes that target variational problems set in the spaces $\boldsymbol{H}(\mathrm{div};\Omega)$ and $\boldsymbol{H}(\mathbf{curl};\Omega)$. They are used for flux-conserving discretizations of second order elliptic boundary value problems [18], provide stream functions in fluid dynamics [31] and are indispensable for viable discretizations of the Maxwell equations [9, 58].

Well known mixed finite element schemes are those introduced by Raviart and Thomas in 2D [68] for $\boldsymbol{H}(\mathrm{div};\Omega)$ and Nédélec in 3D for $\boldsymbol{H}(\mathbf{curl};\Omega)$ [59]. At first glance they have little in common, but a closer scrutiny reveals that they are all specimens of discrete differential forms [11, 46]. As such they are members of a family of finite element schemes that also includes the conventional Lagrangian $H^1(\Omega)$-conforming elements (as discrete 0-forms). Up to now, sparse grid schemes have been confined to generalizing exactly these Lagrangian finite elements. If one is aware of the close relationship of the above-mentioned finite element schemes it is natural to believe that they can all be fit into a sparse grid framework. This motivated the research leading to the current work.

The integral formulation of the Maxwell equations is the best starting point for their discretization [48]. The reason is that differential forms properly describe the geometric character of the electromagnetic laws [13]. Their discrete counterparts, the Whitney forms in the lowest order case, preserve as much as possible of the physical properties of the electromagnetic fields [55, 86, 87]. As Whitney elements are for differential forms what finite elements represent for functions [12], the avenue to suitable discretizations is open. A short justification of the usage of differential forms (pointing to the applications in view) precede the exposition of the Whitney forms on the hypercube. Taking the cue from the sparse grids for Lagrangian finite elements, the treatment of the hierarchical decomposition of the Whitney spaces follows.

Two main ideas are pervasive in the world of sparse grids: the hierarchical multilevel representation of functions and the reduction to one-dimensional considerations thanks to a tensor product structure. Selection of particular hierarchical basis functions yield sparse grids spaces and operators. The commuting diagram property and the existence of discrete sparse potentials are proved. These are issues specific to discrete differential forms, and have no parallels in the theory of sparse grids for Lagrangian finite elements. Then, estimates for the hierarchical contributions of detail spaces are given. The error estimate for the interpolation justifies the choice of the sparse grid. Approximate interpolation is needed for the Galerkin scheme in partial differential equations. The combination technique and a two point quadrature rule ensure that a similar error estimate holds as for the exact interpolation.

With sparse grid interpolation operators for Whitney l-forms in d dimensions and their approximation properties at our disposal, we focus on two problems from the theory of mixed discretizations of second order elliptic PDEs. We are concerned with the stability conditions of the discretization (by sparse Whitney forms) of the primal and of the dual mixed problem. The existence of stable potentials is a sufficient condition, which we can establish for particular cases. The three dimensional case is complete covered by the proofs. Numerical results reveal stability for $d = 4$, too. The complexity of the exterior derivative operator has so far stymied efforts to prove the existence of discrete potentials in the most general formulation.

The last two chapters deal with the general form of some algorithms involved in sparse grid problems and with particular design options and implementation details. The unidirectional paradigm underlies the multilevel transformations. The combination technique is the key to the approximate interpolation. As the hierarchical representation lead to dense the system matrices, the clever ways to realize the multiplication of a vector with the mass matrix are essential and need particular attention. The stiffness matrix multiplication combines the exterior derivative operator and the algorithm for mass matrix. The multigrid method for Whitney forms on sparse grids appears as a multiplicative Schwarz method based on semicoarsening. For the multigrid solver, one needs stencils for mass and stiffness matrices on general anisotropic full grids. The generality is given by three parameters: the dimension of the space, the order of the differential form, and the multidimensional resolution of the grid.

When I started research on Whitney forms on sparse grids I hoped for quick progress and fast implementation. These were illusions, as the sophisticated algorithms were poorly documented in the literature. [6, 19, 23]. I confess, I felt very uncomfortable knowing that efficient algorithms existed, but actually to not be able to write them down. I had to re-discover the tricks of implementation. This yielded, as expected, new ideas, new solutions and new algorithms. Meanwhile, other papers appeared [72, 93] with different approaches. The detailed description of the algorithms and the extensively documented `C++`-code effect my experience that implementation is a scientific challenge peer to theoretical developments.

While writing this thesis, I imagined a reader with my knowledge at the beginning of the work. This is why I have decided to dwell on elementary aspects, as well. However, familiarity with finite element techniques is required. I elaborated on this subject starting from my related previous publications [32–35], but most of the results in Chapter 5 are entirely new, as well as the most of the explanations of the algorithms. Let me end this introduction by quoting R.P. Feynman on scientific writing:

> We have a habit in writing articles published in scientific journals to make the work as finished as possible, to cover up all the tracks, to not worry about the blind alleys or describe how you had the wrong idea first, and so on. So there isn't any place to publish, in a dignified manner, what you actually did in order to get to do the work.

Chapter 2

Discrete Differential Forms

The integral formulation of the Maxwell equations is the best starting point for their discretization [48]. The reason is that differential forms properly describe the geometric character of the electromagnetic laws [13]. Their discrete counterparts, the Whitney forms in the lowest order case, preserve as much as possible of the physical properties of the electromagnetic fields [55, 86, 87]. In the context of the finite element method [16, 17] it has been realized that discrete differential forms supply excellent choices for finite element approximation spaces [11]. They immediately supply conforming finite elements, for instance, in $\boldsymbol{H}(\mathrm{div};\Omega)$ and $\boldsymbol{H}(\mathbf{curl};\Omega)$. This insight accounts for the popularity of so-called edge elements [1, 10, 47, 54, 58, 80, 85]. They are representatives of discrete 1-forms and the natural discrete space for electric and magnetic fields. In this chapter, we shortly justify the use of differential forms in the context of electromagnetics and we present the Whitney forms on hypercube.

2.1 Differential Forms in Electromagnetics

The usual mathematical formulation of Maxwell's equations in terms of vector fields is not suitable for their approximation [11–13]. Denoting by **E** the electric field, **B** the magnetic induction, **H** the magnetic field and **D** the displacement current, it reads in the frequency domain at frecquency $k > 0$

$$\begin{aligned} \mathbf{curl}\,\mathbf{E} &= -ik\mathbf{B} && \text{(Faraday's law)} \\ \mathbf{curl}\,\mathbf{H} &= ik\mathbf{D} + \mathbf{j} && \text{(Ampère's law).} \end{aligned} \tag{2.1}$$

The fields **E**, **H** and **B**, **D** have entirely different nature, as Maxwell remarked in his "Treatise on Electricity and Magnetism" [56]:

> Physical vector quantities may be divided into two classes, in one which the quantity is defined with reference with respect to a line, while in the other the quantity is defined with reference to an area.

Physics suggests an integral formulation

$$\begin{aligned} \int\limits_{\partial\Sigma} \mathbf{E}\cdot d\overrightarrow{s} &= -ik\int\limits_{\Sigma} \mathbf{B}\cdot\mathbf{n}dS && \text{(Faraday's law)} \\ \int\limits_{\partial\Sigma} \mathbf{H}\cdot d\overrightarrow{s} &= ik\int\limits_{\Sigma} \mathbf{D}\cdot\mathbf{n}dS + \int\limits_{\Sigma} \mathbf{j}\cdot\mathbf{n}dS && \text{(Ampère's law)}, \end{aligned} \tag{2.2}$$

for any bounded, oriented, two-dimensional, piecewise smooth sub-manifold Σ of $A(\mathbb{R}^3)$, equipped with oriented unit normal vectorfield **n**.
The local point of view, "field at a point in space" is more comfortable to think about, and the measurement procedures rely on it: The electric field is measured by determining the virtual work for the infinitesimal displacement $\delta\mathbf{x}$ of a test charge q at $\mathbf{x}$

$$\delta w = q\mathbf{E}\cdot\delta\mathbf{x}\,.$$

The magnetic induction is measured through the Lorenz force, i.e. the work required for the infinitesimal shift of a test charge q at $\mathbf{x}$ moving with velocity $\mathbf{v}$

$$\delta w = q\,(\mathbf{B}\times\mathbf{v})\cdot\delta\mathbf{x}\,.$$

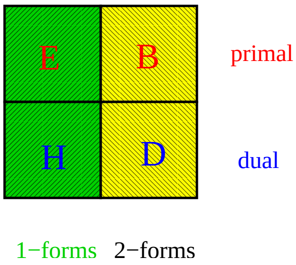


Figure 2.1: Electromagnetic Fields

From this point of view, **E** and **B** are classical (continuous) differential forms of degree 1 and 2, as showed in Fig 2.1. The duality between **E**, **B** and **H**, **D** appear as a consequence of the material laws [13].

An l-form in $\mathbb{R}^d$ $(0 \le l \le d)$ is a mapping of $\mathbb{R}^d$ into the $\binom{d}{l}$-dimensional vector space of alternating l-multilinear forms on $\mathbb{R}^d$. After a basis of $\mathbb{R}^d$ has been chosen, there is a canonical way to identify differential forms with vector fields, their "vector proxies" [14]:

$$\omega = \sum_{(i_1,\ldots,i_l)\subset\{1,\ldots,d\}} u_{i_1,\ldots,i_l}\, dx_{i_1} \wedge \ldots \wedge dx_{i_l} \ , \tag{2.3}$$

where $u_{i_1,\ldots,i_l}$ (with an ordered tuple $i_1 < i_2 < \cdots < i_l$) are the components of the "vector proxies" and $\{dx_1, \ldots, dx_d\}$ form some dual basis of $(\mathbb{R}^d)'$. Actually, the choice of this dual basis is arbitrary and will affect the representation through vector proxies. However, it is perfectly legal to fix one basis and work with the resulting vector proxies of the forms. In particular, we will rely on the canonical basis of $\mathbb{R}^d$. Then for instance, in the tree-dimensional case, a 0-form is a function, a 1-form $\omega = u_1 dx_1 + u_2 dx_2 + u_3 dx_3$ has the vector proxy $(u_1, u_2, u_3)^T$, a 2-form $\omega = u_{12} dx_1 \wedge dx_2 + u_{23} dx_2 \wedge dx_3 + u_{13} dx_1 \wedge dx_3$ has the vector proxy $(u_{12}, u_{23}, u_{13})^T$, and a 3-form $\omega = u_{123} dx_1 \wedge dx_2 \wedge dx_3$ again corresponds to a function u_{123}. The usual identification in $\mathbb{R}^3$ is depicted in Table 2.1. Using this identification, the exterior derivative $\mathbf{d}$ of differential forms spawns the familiar differential operators of vector analysis: **grad**, **curl** and div for 0–forms, 1–forms and 2–forms, respectively. For further details on the calculus of differential forms see [24, 52]. For details on the interpretation of the electromagnetic fields as differential forms, we refer to [48] and the citations quoted there. We synthesize the main equations of the electromagnetics in the language of the differential forms in Fig. 2.2. The material laws link the primal and the dual diagrams building the Maxwell house, in the parlance of Bossavit [13]. Apart from the unification of the classical differential calculus, the geometrical point of view has one more advantage. Passing a material interface, **E** and **H** feature tangential continuity, whereas **D** and **B** have normal continuity. This is automatically ensured by the continuity of the l-form at a surface traversal.

Differential form	Related function u/vector field $\mathbf{u}$
$\mathbf{x} \mapsto \omega(\mathbf{x})$	$\omega(\mathbf{x}) = u(\mathbf{x})$
$\mathbf{x} \mapsto \{\mathbf{v} \mapsto \omega(\mathbf{x})(\mathbf{v})\}$	$\omega(\mathbf{x})(\mathbf{v}) = \langle \mathbf{u}(\mathbf{x}), \mathbf{v} \rangle$
$\mathbf{x} \mapsto \{(\mathbf{v}_1, \mathbf{v}_2) \mapsto \omega(\mathbf{x})(\mathbf{v}_1, \mathbf{v}_2)\}$	$\omega(\mathbf{x})(\mathbf{v}_1, \mathbf{v}_2) = \langle \mathbf{u}(\mathbf{x}), \mathbf{v}_1 \times \mathbf{v}_2 \rangle$
$\mathbf{x} \mapsto \{(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) \mapsto \omega(\mathbf{x})(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)\}$	$\omega(\mathbf{x})(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) = u(\mathbf{x}) \det(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$

Table 2.1: Relationship between differential forms and vector fields in 3D

$$\begin{tikzcd}
\text{0–forms:} & \Phi \arrow[dr, "\mathbf{d}"] & & & & & \\
\text{1–forms:} & \mathbf{A} \arrow[r, "-ik"] \arrow[dr, "\mathbf{d}"'] & \mathbf{E} \arrow[dr, "\mathbf{d}"] & & \mathbf{H} \arrow[dr, "\mathbf{d}"] & & \\
\text{2–forms:} & & \mathbf{B} \arrow[r, "-ik"] \arrow[dr, "\mathbf{d}"] & 0 & \mathbf{D} \arrow[r, "-ik"] \arrow[dr, "\mathbf{d}"'] & \mathbf{j} \arrow[dr, "\mathbf{d}"] & \\
\text{3–forms:} & & & 0 & & \rho \arrow[r, "-ik"] & 0
\end{tikzcd}$$

Figure 2.2: Electomagnetics equations in the language of differential forms

Essentially, the calculus of differential forms is coordinate-free. However, owing to the underlying tensor product structure, the choice of Cartesian coordinate directions is inherent to sparse grid schemes. They will never achieve invariance with respect to transformations of coordinates anyway. Thus, there is no point in sticking to a coordinate free setting and we will always rely on the Euclidean structure of $\mathbb{R}^d$. Hence, we do not distinguish between forms and their vector representatives. Based on the latter we introduce L^2-norms for l-forms. Further, Sobolev spaces of differential forms $\mathbf{H}(\mathbf{d},\Omega)$ can be defined (see [52]) for a bounded domain $\Omega \subset \mathbb{R}^d$ as the Hilbert space

$$\mathbf{H}(\mathbf{d},\Omega) := \{l - \text{forms } \omega,\ \|\omega\|_{L^2(\Omega)} + \|\mathbf{d}\omega\|_{L^2(\Omega)} < \infty\} \ .$$

The corresponding norm involves the exterior derivative $\mathbf{d}$:

$$\|\omega\|^2_{\mathbf{H}(\mathbf{d},\Omega)} := \|\omega\|^2_{L^2(\Omega)} + \|\mathbf{d}\omega\|^2_{L^2(\Omega)} \ .$$

Recalling the incarnations $\mathbf{grad}, \mathbf{curl}, \operatorname{div}$ of the exterior derivative in the case of vector proxies in three dimensions, the spaces $\mathbf{H}(\mathbf{d},\Omega)$ are easily identified as generalization of $H^1(\Omega)$, $\boldsymbol{H}(\mathbf{curl};\Omega)$, and $\boldsymbol{H}(\operatorname{div};\Omega)$. The corresponding spaces of functions with zero traces on the boundary $\Gamma := \partial\Omega$ can also be generalized and yield the spaces

$$\mathbf{H}_0(\mathbf{d},\Omega) := \{\omega \in \mathbf{H}(\mathbf{d},\Omega),\ \omega_{|\partial\Omega} = 0\} \ .$$

They are required for imposing Dirichlet boundary conditions.

For a given square integrable l-form ξ, consider the following $\mathbf{H}(\mathbf{d},\Omega)$-elliptic variational problem with constant real coefficients α,β: seek the l-form $\omega \in \mathbf{H}_0(\mathbf{d},\Omega)$ such that

$$(\alpha\mathbf{d}\omega, \mathbf{d}\eta)_{L^2(\Omega)} + (\beta\omega,\eta)_{L^2(\Omega)} = (\xi,\eta)_{L^2(\Omega)} \quad \forall \eta \in \mathbf{H}_0(\mathbf{d},\Omega) \ , \tag{2.4}$$

This is the generalization of the second order elliptic boundary value problem with constant real coefficients and right hand side $f \in L^2(\Omega)$: seek the function $u \in H_0^1(\Omega)$ such that

$$(\alpha\, \mathbf{grad}\, u, \mathbf{grad}\, v)_{L^2(\Omega)} + (\beta u, v)_{L^2(\Omega)} = (f, v)_{L^2(\Omega)} \quad \forall v \in H_0^1(\Omega)\,, \tag{2.5}$$

which arises from (2.4) for $l = 0$. Moreover, in 3D we also get the following two variational problems from (2.4). For $l = 1$: seek

$$u \in \boldsymbol{H}_\Gamma(\mathbf{curl};\Omega) := \{\mathbf{u} \in \boldsymbol{H}(\mathbf{curl};\Omega),\ \mathbf{u} \times \mathbf{n} = 0 \text{ on } \Gamma\}$$

such that

$$(\alpha\, \mathbf{curl}\, \mathbf{u}, \mathbf{curl}\, \mathbf{v})_{L^2(\Omega)} + (\beta \mathbf{u}, \mathbf{v})_{L^2(\Omega)} = (\mathbf{f}, \mathbf{v})_{L^2(\Omega)} \quad \forall \mathbf{v} \in \boldsymbol{H}_\Gamma(\mathbf{curl};\Omega)\,. \tag{2.6}$$

For $l = 2$ we end up with: seek

$$\mathbf{u} \in \boldsymbol{H}_\Gamma(\mathrm{div};\Omega) := \{\mathbf{u} \in \boldsymbol{H}(\mathrm{div};\Omega),\ \mathbf{u} \cdot \mathbf{n} = 0 \text{ on } \Gamma\}$$

such that

$$(\alpha\, \mathrm{div}\, \mathbf{u}, \mathrm{div}\, \mathbf{v})_{L^2(\Omega)} + (\beta \mathbf{u}, \mathbf{v})_{L^2(\Omega)} = (\mathbf{f}, \mathbf{v})_{L^2(\Omega)} \quad \forall \mathbf{v} \in \boldsymbol{H}_\Gamma(\mathrm{div};\Omega)\,, \tag{2.7}$$

where the right hand side $\mathbf{f} \in \boldsymbol{L}^2(\Omega)$ is given and $\mathbf{n}$ is the normal to the boundary Γ oriented towards the exterior of the domain Ω.

Before we present the discretizaton of differential forms, we give the variational formulations of Maxwell's equations. The material laws come into play by the sequi-linear forms $a_\mu, a_{1/\mu}, a_\epsilon, a_{1/\epsilon}$. We do not delve into technical details, rather we refer to [48]. The first is the "**E**-based" formulation, or, by analogy to the weak formulations of second order elliptic problems [18], the *primal variational formulation*

$$a_{1/\mu}(\mathbf{dE}, \mathbf{d}\eta) - k^2 a_\epsilon(\mathbf{E}, \eta) = -ik \int\limits_\Omega \mathbf{j} \wedge \eta + ik \int_{\partial\Omega} \mathbf{H} \wedge \eta \text{ for all 1-forms } \eta\,.$$

The second is the "**H**-based" formulation, or the *dual variational formulation*

$$a_{1/\epsilon}(\mathbf{dH}, \mathbf{d}\eta) - k^2 a_\mu(\mathbf{H}, \eta) = a_{1/\epsilon}(\mathbf{j}, \eta) - ik \int_{\partial\Omega} \mathbf{E} \wedge \eta \text{ for all 1-forms } \eta\,.$$

In the case of an eddy current model, the $\mathbf{E}$-based formulation gives the problem

$$a_{1/\mu}(\mathbf{dE}, \mathbf{d}\eta) + ika_\sigma(\mathbf{E}, \eta) = -ik \int\limits_\Omega \mathbf{j} \wedge \eta \ ,$$

or, in the time domain we get the degenerate parabolic problem

$$a_{1/\mu}(\mathbf{dE}, \mathbf{d}\eta) + a_\sigma(\partial_t\mathbf{E}, \eta) = - \int\limits_\Omega \partial_t\mathbf{j} \wedge \eta \ .$$

Implicit time discretizations of the above problem lead to elliptic formulations as in equation (2.4).

2.2 Whitney Forms on the Hypercube

Discrete differential forms should preserve as much as possible of the properties of their continuous counterparts. In light of the previous section, it is mandatory to consider vector valued finite elements with partial continuity. The discretization relies on integral degrees of freedom corresponding to currents and flows [13] and in the lowest order case yields the Whitney forms. The discrete differential forms are built upon triangulations and provide parametric equivalent families of finitele elements [46]. Their construction by interpolation is presented in [15] and was used in [33] to construct Whitney forms on pyramids.

Now, we focus on the essential properties of discrete differential forms.

As we aim at conforming finite element spaces, the traces of discrete differential forms onto any interelement boundary (a $(l-1)$-face) have to be unique and they have to be fixed by the degrees of freedom associated with that face. This makes the vector proxies fulfill the patching condition: C^0-continuity for discrete 0-forms, tangential and normal continuity for 1-forms and, respectively, 2-forms. This guarantees that they provide finite elements conforming in $H^1(\Omega)$, $\boldsymbol{H}(\mathbf{curl};\Omega)$, and $\boldsymbol{H}(\mathrm{div};\Omega)$, respectively.

Following the lines of [46], the construction of conforming finite element spaces for differential forms must provide some more properties than the unisolvence of degrees of freedom.

First, an "*exact sequence property*" must hold for the spaces of discrete differential forms, if Ω is contractible: The exterior derivative of a discrete l-form is to yield a valid discrete $(l+1)$-form. In addition, any discrete

$(l+1)$-form with vanishing exterior derivative should have a representation as the exterior derivative of some discrete l-form.

Finally, discrete differential forms have to possess approximation properties, in order to be useful for Galerkin discretizations. It is a standard insight in finite elements that satisfactory approximation properties are directly linked to the fact that all polynomials of a certain degree are contained in the spaces on the reference elements [17]. The components of discrete differential forms are multilinear polynomials. In the case of Whitney-forms that provide only first order schemes, we have to make sure that all constant forms can be represented.

We summarize the essential algebraic properties demanding that the following diagram commutes and that the horizontal sequences are exact:

$$\begin{tikzcd}
\mathcal{DF}^0 \arrow[r, "\mathbf{d}"] \arrow[d, "\mathcal{I}_0"] & \mathcal{DF}^1 \arrow[r, "\mathbf{d}"] \arrow[d, "\mathcal{I}_1"] & \mathcal{DF}^2 \arrow[r, "\mathbf{d}"] \arrow[d, "\mathcal{I}_2"] & \cdots \arrow[r, "\mathbf{d}"] \arrow[d, phantom, "\cdots"] & \mathcal{DF}^d \arrow[d, "\mathcal{I}_d"] \\
\mathcal{V}^0(\mathcal{T}_h) \arrow[r, "\mathbf{d}"] & \mathcal{V}^1(\mathcal{T}_h) \arrow[r, "\mathbf{d}"] & \mathcal{V}^2(\mathcal{T}_h) \arrow[r, "\mathbf{d}"] & \cdots \arrow[r, "\mathbf{d}"] & \mathcal{V}^d(\mathcal{T}_h)
\end{tikzcd}$$

Here, $\mathcal{DF}^l$ represents the space of continuous differential l-forms on the domain Ω, and $\mathcal{V}^l(\mathcal{T}_h)$ is the space of discrete l-forms on a triangulation $\mathcal{T}_h$ of the domain, whereas $\mathcal{I}_0, \ldots, \mathcal{I}_d$ are the corresponding approximation operators.

In the one dimensional case, the Whitney 0-forms and 1-forms have plain functions as vector representatives. To 0-forms correspond the continuous, piecewise linear finite elements. The degrees of freedom are values of functions in the nodes of the triangulation, and the interpolation is linear. For 1-forms, the degrees of freedom are integrals of the function along intervals. In this case, we interpolate by piecewise constants.

In general d dimensions, we define the space of Whitney l-forms on the hypercube $\hat{T} :=]0;1]^d$ in $\mathbb{R}^d$ (cf. [59] as main reference). Denote by $\mathcal{P}_k$ the space of polynomials of degree k and by $dx_I = dx_{i_1} \wedge \ldots \wedge dx_{i_l}$, the ordered tuple $I = \{i_1, \ldots, i_l\}$. For a multi-index I, the I-component of the vector representative of a Whitney form is tensor product of linear (for $j \notin I$) and constant functions (for $j \in I$).

Definition 2.2.1. *The space of Whitney l-forms on the reference element is*

$$\begin{aligned}
\mathcal{V}^l(\hat{T}) := \mathit{span}\{\omega(\mathbf{x}) \;&=\; p_I(\mathbf{x}) dx_I,\ I \subset \{1, \ldots, d\},\ |I| = l, \\
&\qquad \mathit{where}\ p_I(\mathbf{x}) = p_1(x_1) \ldots p_d(x_d), \\
&\quad p_j \in \mathcal{P}_1(]0,1]),\ \mathit{if}\ j \notin I\ \mathit{and}\ p_j \in \mathcal{P}_0(]0,1]),\ \mathit{if}\ j \in I\}\,.
\end{aligned}$$

After the local spaces have been fixed we have to specify local degrees of freedom. For Whitney forms, these are integrals over the l-dimensional "faces" of $\hat{T}$. We point out that integration over l-dimensional surfaces is the most natural operation for differential l-forms. Unisolvence of these local degrees of freedom is established by a straightforward counting argument. It goes without saying that the corresponding global degrees of freedom involve integrals over all l-dimensional faces of the tensor product grid. Then, the global space $\mathcal{V}^l$ of discrete l-forms on a tensor product grid Ω_h can be defined by demanding that on each d-dimensional brick Def. 2.2.1 applies and that the global degrees of freedom have to be uniquely defined.

Let us consider the example of the unit cube $\hat{T} =]0;1]^3$ in three dimensions ($d = 3$). The Whitney 0-forms are just polynomials $\omega(x) = (a_1x_1+b_1)(a_2x_2+b_2)(a_3x_3+b_3)$, so

$$\mathcal{V}^0(\hat{T}) = \operatorname{span}\{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}\ ,$$

where

$$\begin{aligned}
u_1 &= (1-x_1)(1-x_2)(1-x_3)\ , & u_2 &= x_1(1-x_2)(1-x_3)\ ,\\
u_3 &= (1-x_1)x_2(1-x_3)\ , & u_4 &= (1-x_1)(1-x_2)x_3\ ,\\
u_5 &= x_1x_2(1-x_3)\ , & u_6 &= (1-x_1)x_2x_3\ ,\\
u_7 &= x_1(1-x_2)x_3\ , & u_8 &= x_1x_2x_3\ .
\end{aligned}$$

The corresponding degrees of freedom are the values of the function in the vertices of the cube (nodal values). We have recovered the classical Lagrangian finite elements.

Denoting

$$\begin{aligned}
u^1_1 &= (1-x_2)(1-x_3)\ , & u^1_2 &= (1-x_2)x_3\ ,\\
u^1_3 &= x_2(1-x_3)\ , & u^1_4 &= x_2x_3\ ,\\
u^2_1 &= (1-x_3)(1-x_1)\ , & u^2_2 &= (1-x_3)x_1\ ,\\
u^2_3 &= x_3(1-x_1)\ , & u^2_4 &= x_3x_1\ ,\\
u^3_1 &= (1-x_1)(1-x_2)\ , & u^3_2 &= (1-x_1)x_2\ ,\\
u^3_3 &= x_1(1-x_2)\ , & u^3_4 &= x_1x_2\ ,
\end{aligned}$$

the 1-forms are generated by

$$\begin{aligned}
\mathcal{V}^1(\hat{T}) = \operatorname{span}\{ \quad & u^1_1dx_1\ ,\ u^1_2dx_1\ ,\ u^1_3dx_1\ ,\ u^1_4dx_1\ ,\\
& u^2_1dx_2\ ,\ u^2_2dx_2\ ,\ u^2_3dx_2\ ,\ u^2_4dx_2\ ,\\
& u^3_1dx_3\ ,\ u^3_2dx_3\ ,\ u^3_3dx_3\ ,\ u^3_4dx_3\}\ ,
\end{aligned}$$

This is Nédélec's $\boldsymbol{H}(\mathbf{curl};\Omega)$-conforming space [59], and the degrees of freedom are given by path integrals along the edges. The local space of Whitney 2-forms is

$$\begin{aligned}\mathcal{V}^2(\hat{T}) = \mathrm{span}\{ \quad & u_1^{12} dx_1 \wedge dx_2, u_2^{12} dx_1 \wedge dx_2,\ u_1^{23} dx_2 \wedge dx_3, \\ & u_2^{23} dx_2 \wedge dx_3,\ u_1^{31} dx_3 \wedge dx_1,\ u_2^{31} dx_3 \wedge dx_1\}\ ,\end{aligned}$$

where

$$\begin{array}{rclcrcl} u_1^{12} & = & 1 - x_3\ , & \quad & u_2^{12} & = & x_3\ , \\ u_1^{23} & = & 1 - x_1\ , & & u_2^{23} & = & x_1\ , \\ u_1^{31} & = & 1 - x_2\ , & & u_2^{31} & = & x_2\ . \end{array}$$

This is the Raviart-Thomas space [68] in three dimensions. In this case, the degrees of freedom are integrals of normal components over surfaces (fluxes). For completeness we give the local discrete 3-forms, which are just constants

$$\mathcal{V}_0^3(\hat{T}) = \mathrm{span}\{dx_1 \wedge dx_2 \wedge dx_3\}$$

with the integrals over elements as degrees of freedom. The location of the degrees of freedom for Whitney 0, 1, and 2-forms are depicted in Figure 2.3. It turns out that the traces onto l-faces of the local discrete l-forms according to Def. 2.2.1 are already fixed by prescribing the value of the integrals of the form over these faces. As explained in [46], this ensures conformity of the spaces of discrete differential forms in the corresponding Sobolev spaces of continuous differential forms.

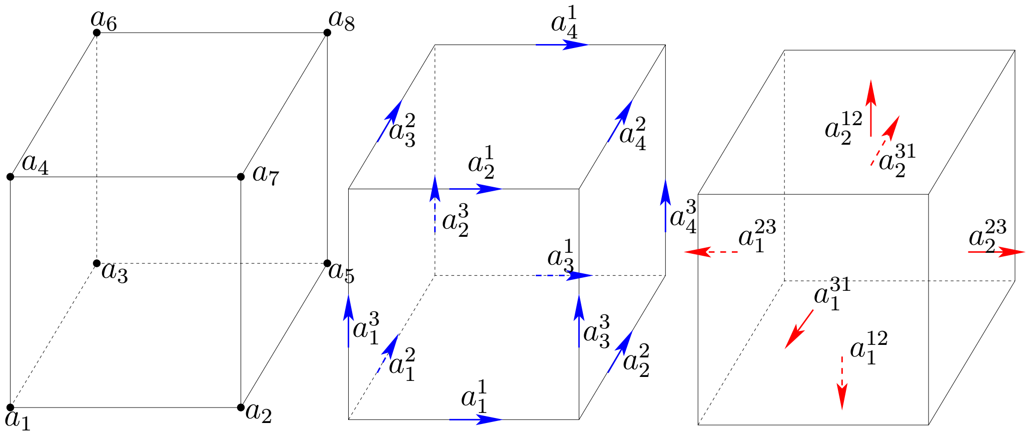


Figure 2.3: Degrees of freedom a_i, a_j^i, a_k^{ij} correspond to u_i, u_j^i, u_k^{ij}, respectively

For the concrete case of $d = 3$, continuity of the traces across interelement faces boils down to the conventional conformity conditions for finite elements. These are C^0-continuity for discrete 0-forms, tangential and normal continuity for 1-forms and, respectively, 2-forms. This guarantees that discrete differential forms provide finite elements conforming in $H^1(\Omega)$, $\boldsymbol{H}(\mathbf{curl};\Omega)$, and $\boldsymbol{H}(\mathrm{div};\Omega)$, respectively.

Let us mention two more essential properties of discrete differential forms. The first is the so-called commuting diagram property, which means that the nodal interpolation operators commute with the exterior derivative. What we call nodal interpolation is the projection onto the space of discrete differential forms defined via the global degrees of freedom. If we denote the local interpolation for discrete l-forms by $\mathcal{I}^l$, we find

$$\mathbf{d} \circ \mathcal{I}^l = \mathcal{I}^{l+1} \circ \mathbf{d} \ , \tag{2.8}$$

for sufficiently smooth differential forms. For details consult [46].

The second important feature of discrete differential forms is the existence of discrete potentials: if the domain triangulated by Ω_h is homeomorphic to a ball, and $\mathbf{d}\omega_h = 0$ for some $\omega_h \in \mathcal{V}^l(\mathcal{T}_h)$, then there exists $\eta_h \in \mathcal{V}^{l-1}(\mathcal{T}_h)$ such that $\omega_h = \mathbf{d}\eta_h$. A thorough discussion is provided in [46, Sect. 6].

Remark 2.2.1. *Discrete differential forms exist for higher polynomial orders. In general, the space of discrete differential l-forms of polynomial degree k on the unit hypercube is defined as*

$$\begin{aligned}\mathcal{V}_k^l := \mathit{span}\{\omega(\mathbf{x}) \;&=\; p_I(\mathbf{x})dx_I,\, I \subset \{1,\dots,d\},\, |I| = l,\\ &\quad\ \textit{where } p_I(\mathbf{x}) = p_1(x_1)\dots p_d(x_d),\\ &\quad p_j \in \mathcal{P}_{k+1}(]0,1]),\ \textit{if } j \notin I \ \textit{and } p_j \in \mathcal{P}_k(]0,1]),\ \textit{if } j \in I\}.\end{aligned}$$

The formulation of the Maxwell equations in terms of differential forms makes the discretization task easier. As Whitney elements are for differential forms in $\mathbf{H}(\mathbf{d},\Omega)$ what finite elements represent for functions in $H^1(\Omega)$, the avenue to suitable discretizations is open. Taking the cue from the sparse grid variant of Lagrangian finite elements, we present the hierarchical decomposition of the Whitney spaces in the next chapter.

Chapter 3

Hierarchical Decompositions

Two main ideas are pervasive in the world of sparse grids: the hierarchical multilevel representation of functions and the reduction to one-dimensional considerations thanks to a tensor product structure. Selection of particular hierarchical basis functions yields sparse grids spaces. In this chapter, we explain the hierarchical multilevel principle and give examples in the one dimensional case. Along the way, we supply essential prerequisites for the general construction. We handle the case of the l-forms in d-dimensions by the tensor product technique.

3.1 Concept and Examples

The interaction between different scales of discretization provides some of the most successful numerical schemes, for example the multigrid method. One of the principal ingredients in the derivation of the sparse grids is the hierarchical decomposition of spaces. First, we present the general framework which leads (through multiresolution) to wavelets [76, 78], too. Then, we give essential examples in the one dimensional case.

Consider an increasing sequence of nested finite-dimensional subspaces (equipped with their nodal base)

$$V_0 \subset V_1 \subset \cdots \subset V_\nu \subset V_{\nu+1} \subset \cdots \subset X$$

meant to approximate the space X. A function f in the whole space X has a component f_ν in each subspace V_ν. These components f_ν contain more and more of the full information on f. One requirement on the sequence of subspaces is completeness:

$$f_\nu \longrightarrow f \quad \text{as } \nu \longrightarrow \infty\,.$$

The crucial fact is that more information is hidden in the whole scale of spaces that in one space alone: from a level ν to the next, *new details* are revealed. Pieces of f from each subspace give finer and finer details of f. The space W_ν contains the new information $\Delta f_\nu = f_{\nu+1} - f_\nu$, the “detail” at level ν:

$$f_\nu + \Delta f_\nu = f_{\nu+1} \qquad V_\nu \oplus W_\nu = V_{\nu+1}$$

Recognizing how W’s add to V, we can see V_ν as a partial sum. Start from

$$V_0 \oplus W_0 = V_1 \qquad \text{and} \qquad V_1 \oplus W_1 = V_2.$$

Successive substitution gives

$$V_0 \oplus W_0 \oplus W_1 = V_2 \qquad V_0 \oplus W_0 \oplus W_1 \cdots \oplus W_\nu = V_{\nu+1}$$

For functions in those subspaces

$$f_0 + \Delta f_0 + \Delta f_1 + \cdots + \Delta f_\nu = f_{\nu+1}$$

which is nothing but the “telescopic” sum

$$f_0 + (f_1 - f_0) + (f_2 - f_1) + \cdots + (f_{\nu+1} - f_\nu) = f_{\nu+1}.$$

It is convenient (but not essential) if W_ν is orthogonal to V_ν in a Hilbert space X. Each W_ν is then automatically orthogonal to all $W_{\nu'}$. The completeness condition can be restated, for instance, as

$$V_0 \oplus \sum_{\nu=0}^{\infty} W_\nu = X.$$

Any nonorthogonal basis $b_0, b_1, \ldots$ in X gives a nonorthogonal example:

$$\begin{array}{lll} \text{Sum up to } \nu\text{:} \quad f_\nu = \sum_0^\nu c_k b_k & \text{is in} & V_\nu \\ \text{Next detail:} \quad \Delta f_\nu = c_{\nu+1} b_{\nu+1} & \text{is in} & W_\nu. \end{array}$$

Multilevel representations have a naturally built-in adaptivity though their ability to directly express and separate components living on different scales. At a fixed level ν,

$$V_0 \oplus W_0 \oplus W_1 \cdots \oplus W_\nu = V_{\nu+1},$$

we have *two kinds of bases* in $V_{\nu+1}$: the **nodal** and the **hierarchical**, where the latter are obtained from basis functions of $V_0, W_0, \ldots, W_\nu$. The basis transformation is called the hierarchical transform. It maps the *nodal* coefficients (degrees of freedom for the nodal basis) onto the *hierarchical* coefficients (for the hierarchical basis). Concrete hierarchical bases are specified by the requirement that the basis functions lie in the kernel of degrees of freedom on all coarser levels.

Now, let us consider two examples of the hierarchical decomposition in the one dimensional case. They address the case of 0-forms and 1-forms on $[0, 1]$.

Discrete 0-forms in 1D

We begin with the Lagrangian finite elements. The interpolation functions f_ν are determined by nodal values of f. Then, $f_{\nu+1} - f_\nu$ vanishes at all nodes corresponding to level ν and therefore can be represented by values of f only in the new nodes at level $\nu + 1$. The hat function

$$\varphi(t) = \begin{cases} 1+t, & \text{if } t \text{ is in }]-1,0] \\ 1-t, & \text{if } t \text{ is in }]0,1] \\ \quad 0, & \text{otherwise} \end{cases}$$

is in the following the “mother” of the scale functions:

$$\varphi_{\nu i} = \varphi(2^\nu t - i), \quad \bar{\varphi}_{\nu k}(t) = -2^{-(\nu+1)}\varphi_{\nu k}(t). \tag{3.1}$$

Note their orthogonality with respect to the bilinear form $a(f, g) = \int_0^1 f'g'dt$. There is no orthogonality with respect to the L^2-scalar product. Prewavelets [27, 40, 41, 61, 66] can be an alternative, since they are L^2-orthogonal at different levels, but we do not adopt this approach in the following.

We reserve the symbol ν for levels, Θ_ν is the corresponding set of odd indices from 1 to $2^\nu - 1$, and $\Theta_0 := \{0, 1\}$. The nodal spaces are defined as

$$V_\nu = \operatorname{span}\{\varphi_{\nu k}, k = 0, 1, \ldots, 2^\nu\}\,,$$

and the detail (hierarchical surplus) spaces for $\nu \geq 1$ are

$$\begin{aligned} W_\nu &= \operatorname{span}\{\varphi_{\nu k}, k \in \Theta_\nu\}\,, \quad \text{for } \nu \geq 1 \\ W_0 &= V_0\,. \end{aligned}$$

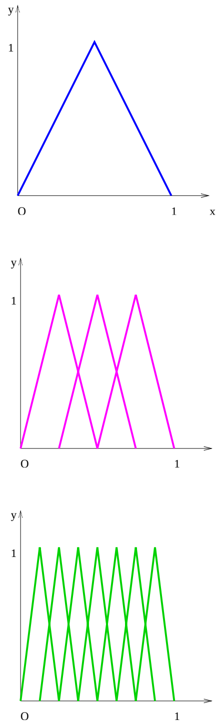


Figure 3.1: Nodal Basis

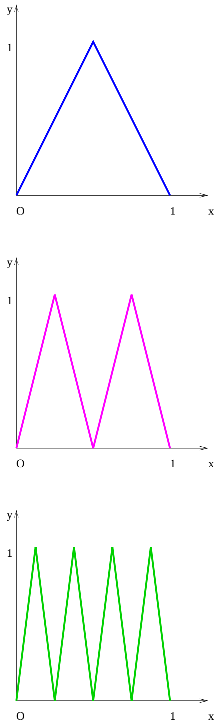


Figure 3.2: Hierarchical Basis

The **nodal representation** of a function u is

$$u_\nu(t) = \sum_{k=0}^{2^\nu} u_{\nu k}\varphi_{\nu k}(t)\,, \quad u_{\nu k} = u(k2^{-\nu})\,, \tag{3.2}$$

and the **hierarchical representation** is

$$u_\nu(t) = \sum_{m=0}^{\nu} u_m^W(t) = \sum_{m=0}^{\nu}\sum_{k\in\Theta_m} v_{mk}\varphi_{mk}(t). \tag{3.3}$$

We display the nodal basis functions in Figure 3.1, and the hierarchical bases for surplus spaces in Figure 3.2.
We use the values of the norm of the hat functions:

$$\|\varphi_{\nu\,q}\|_{L^2(\Omega)} = 3^{-1/2}2^{-(\nu-1)/2}\ . \tag{3.4}$$

Discrete 1-forms in 1D

Now, let us focus on Whitney 1-forms on an equidistant grid on $\Omega = [0,1]$. As pointed out in the preceding section, we interpolate by piecewise constants. The degrees of freedom are integrals over the grid cells. This leads to the box functions for the nodal representation

$$B_{\nu k}(t) = 2^\nu B(2^\nu t - k), \quad k = 0, 1, \ldots, 2^\nu\,,$$

where $B(t)$ is the characteristic function of the interval $]0,1]$. Note that

$$\int_{\mathbb{R}} B_{\nu k}(t)dt = \int_{k2^{-\nu}}^{(k+1)2^{-\nu}} B_{\nu k}(t)dt = 1\,.$$

The approximation of a function u at level ν is

$$u_\nu(t) = \sum_{k=0}^{2^\nu-1} a_{\nu k} B_{\nu k}(t), \tag{3.5}$$

where the (scaling) coefficients are

$$a_{\nu k} = \int\limits_{k2^{-\nu}}^{(k+1)2^{-\nu}} u(t)dt = \int\limits_{[0,1]} u(t)B(2^\nu t - k)dt = 2^{-\nu} \int\limits_{[0,1]} u(t)B_{\nu k}(t)dt. \tag{3.6}$$

It is easy to see that the hierarchical basis is comprised of the Haar-wavelets [78]: $w(t) = B(2t) - B(2t-1)$. In order to comply with the classical notations in the sparse grid theory, let the "mother" function be

$$\psi(t) = 2^{-1/2} \begin{cases} 1, & \text{if } t \text{ is in }]-1,0] \\ -1, & \text{if } t \text{ is in }]0,1] \\ 0, & \text{otherwise} \end{cases} = 2^{-1/2} w(\frac{1}{2}t + \frac{1}{2}),$$

which generates the hierarchical basis functions

$$\psi_{\nu\, k}(t) = 2^{\nu/2}\psi(2^\nu t - k)\,, \qquad k \in \Theta_\nu\,.$$

We reuse the symbols for the spaces in order to stress that the construction concept applies in both cases: 0-forms and 1-forms. The scale spaces are now

$$V_\nu = \text{span}\{B_{\nu\, k},\ k = 0, 1, 2, \ldots, 2^\nu - 1\}$$

and the detail (Haar-wavelet) spaces are

$$\begin{aligned} W_\nu &= \operatorname{span}\{\psi_{\nu\, k},\ k \in \Theta_\nu\} \quad \text{for } \nu \geq 1 \\ W_0 &= V_0 = \operatorname{span}\{B_{00}\}\,. \end{aligned}$$

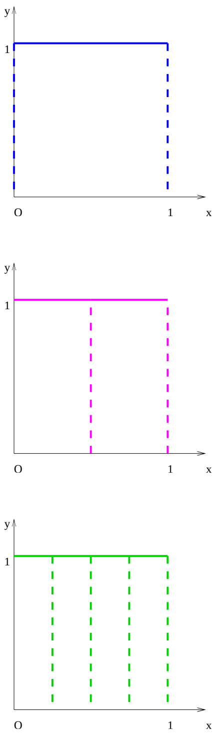


Figure 3.3: Scale functions

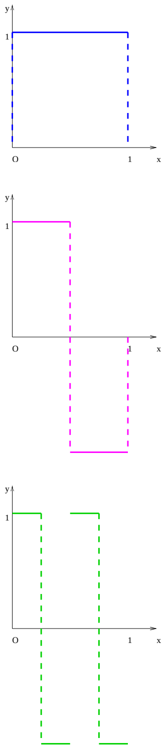


Figure 3.4: Basis for wavelet spaces

From the **nodal** representation (3.5) we arrive at the **hierarchical**

$$u_\nu(t) = a_{00}B_{00}(t) + \sum_{m=1}^{\nu} \sum_{k\in\Theta_m} b_{mk}\psi_{mk}(t) = a_{00}B_{00}(t) + \sum_{m=1}^{\nu} u_\nu^W(t)\ , \tag{3.7}$$

with

$$b_{mk} = \int\limits_{[0,1]} u(t)\psi_{mk}(t)\,dt\ , \quad m \in \{1,\ldots,\nu\},\ k \in \Theta_m\ . \tag{3.8}$$

We adopt the notation $\psi_{00} = B_{00}$. The hierarchical transform is just the Haar-wavelet transform, recursively done by the pyramid algorithm [76, 78].

Finally, we summarize some elementary properties:
The Haar-wavelet basis is L^2-orthogonal and it holds:

$$\|\psi_{\nu q}\|_{L^2(\Omega)} = 1 \, . \tag{3.9}$$

The derivative of a hat function is a Haar-wavelet:

$$\varphi'_{\nu k} = 2^{(\nu+1)/2} \psi_{\nu k} \, , \tag{3.10}$$

and it can be written in terms of box functions:

$$\varphi'_{\nu k} = B_{\nu k} - B_{\nu k-1} \, . \tag{3.11}$$

The Haar-wavelet coefficients $b_{\nu k}$ can be represented as

$$b_{\nu k} = 2^{-(\nu+1)/2} \int_\Omega \varphi_{\nu k}(t) u(t)' dt = 2^{(\nu+1)/2} \int_\Omega \bar{\varphi}_{\nu k}(t) u(t)' dt \, . \tag{3.12}$$

The Haar-wavelet coefficients decay according to

$$|b_{\nu k}| \leq \frac{2}{3} 2^{-\nu} \|u'\|_{L^2(\Omega)} \, . \tag{3.13}$$

3.2 Hierarchical Interpolation

The tensor product is the natural way to pass to the general d-dimensional case. The functions of one variable $f_j(t)$ for $j = 1, \ldots, d$ build the function of d variables $f_1 \otimes \cdots \otimes f_d = \otimes_{j=1}^d f_j$ defined as

$$\otimes_{j=1}^d f_j(x_1, \ldots, x_d) = \prod_{j=1}^d f_j(x_j).$$

Similarly, the closed subspaces $X_j \subset L^2(\mathbb{R})$ for $j = 1, \ldots, d$ form a closed subspace of $L^2(\mathbb{R}^d)$ denoted by $\otimes_{j=1}^d X_j = X_1 \otimes \cdots \otimes X_d$ and defined as the closed linear span in $L^2(\mathbb{R}^d)$ of all functions of the form $f_1(x_1) \cdot \ldots \cdot f_d(x_d)$ where $f_j \in X_j$ for all $j = 1, \ldots, d$. It is easy to check that the tensor product of orthonormal bases gives an orthonormal basis.

This construction leads to sparse grids for Lagrangian finite elements [23], which are just the discrete differential 0-forms. Clearly, the Haar-wavelet approximation is the Whitney d-form. For intermediate orders $1 \leq l \leq d-1$,

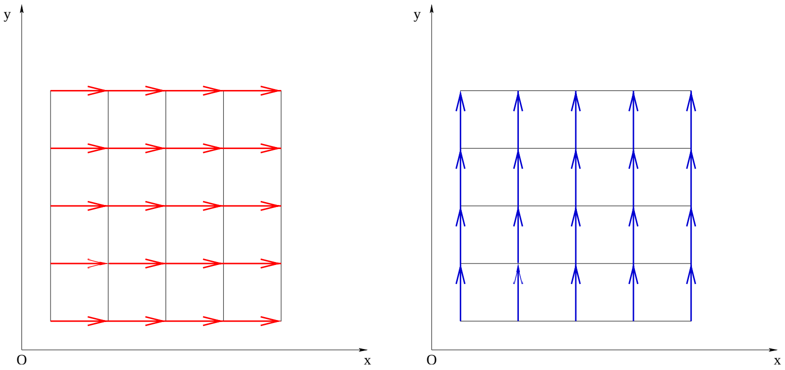


Figure 3.5: Degrees of freedom for components of an 1-form ($d = 2$, $n = 2$)

we have to resort to tensor products of hat and box functions, as pointed out in Section 2.2.
Denote the domain

$$\Omega := [0,1]^d$$

and use a multi-index for the level

$$\boldsymbol{\nu} = (\nu_1, \ldots, \nu_d) \in \mathbb{N}^d.$$

The (possibly anisotropic) grid $\Omega_{\boldsymbol{\nu}}$ with mesh size

$$\mathbf{h}_{\boldsymbol{\nu}} := (h_{\nu_1}, \ldots, h_{\nu_d}) = (2^{-\nu_1}, \ldots, 2^{-\nu_d})$$

contains the interior points

$$x_{\boldsymbol{\nu}\mathbf{q}} = (q_1 h_{\nu_1}, \ldots, q_d h_{\nu_d}), \text{ with } 1 \leq q_j \leq 2^{\nu_j} - 1 \text{ (for } j = 1, \ldots, d)$$

and the boundary points

$$x_{\boldsymbol{\nu}\mathbf{q}} = (q_1 h_{\nu_1}, \ldots, q_d h_{\nu_d}), \text{ with } q_i = 0, 1 \text{ for } \nu_i = 0.$$

Consider now an l-form in d-dimensions

$$\omega = \sum_I u_I dx_I$$

with the ordered multi-index $I = \{i_1, i_2, \ldots, i_l\} \subset \{1, 2, \ldots, d\}$ of length l, and $dx_I = dx_{i_1} \wedge dx_{i_2} \wedge \ldots \wedge dx_{i_l}$.

According to Sect. 2.2, the **(nodal) basis function** corresponding to the l-face $F := [x_{\boldsymbol{\nu}\mathbf{q}};\, \vec{e}_{i_1}, \vec{e}_{i_2}, \ldots, \vec{e}_{i_l}]$ is

$$\Phi_F^I := \prod_{i \in I} B_{\nu_i q_i} \prod_{j \notin I} \varphi_{\nu_j q_j} \tag{3.14}$$

with allowed indices

$$\mathbf{q} \in \{\mathbf{q} = (q_1, \ldots, q_d),\ 0 \leq q_j \leq\ 2^{\nu_j},\ j \notin I,\ 0 \leq q_i \leq\ 2^{\nu_i} - 1,\ i \in I\}.$$

The tensor product grid allows us to work on components in order to define the **nodal interpolation operator**:

$$\boldsymbol{\mathcal{I}}_{\boldsymbol{\nu}} \omega = \sum_I \left(\mathcal{I}_{\boldsymbol{\nu}}^I u_I\right) dx_I, \qquad \mathcal{I}_{\boldsymbol{\nu}}^I u_I = \sum_{F \,\|\, e_I} \alpha_F \Phi_F^I \tag{3.15}$$

where the last sum runs over all grid faces parallel to the space generated by directions of interpolation $e_I = \operatorname{span}\{\vec{e}_{i_1}, \vec{e}_{i_2}, \ldots, \vec{e}_{i_l}\}$. As mentioned in the previous chapter, we mainly deal with differential forms through the components of their vector proxies. In a sense, a Whitney l-form $\omega_n \in V_n^l$ arises by discretizing each of the $\binom{d}{l}$ components of the vector proxies separately.

Definition 3.2.1. *The space $V_{\boldsymbol{\nu}}^l$ of the Whitney l-forms on the anisotropic full grid $\Omega_{\boldsymbol{\nu}}$ of resolution $\boldsymbol{\nu}$ is the range of the nodal interpolation operator $\mathcal{I}_{\boldsymbol{\nu}}$. The function space for the I-component of the vector proxy $V_{\boldsymbol{\nu}}^{I,l}$ is the range of the corresponding nodal interpolation operator $\mathcal{I}_{\boldsymbol{\nu}}^I$.*

The degrees of freedom α_F from (3.15) are integrals on corresponding faces (see Figure 3.5)

$$\alpha_F := \int_F \omega. \tag{3.16}$$

For an arbitrary fixed face F_0 parallel to the space generated by directions from I_0, that is $F_0 \ \| \ e_{I_0} = \operatorname{span}\{\vec{e}_{i_1}, \vec{e}_{i_2}, \ldots, \vec{e}_{i_l}\}$, we have:

$$\int_{F_0} \mathcal{I}_{\boldsymbol{\nu}} \omega = \int_{F_0} \mathcal{I}_{\boldsymbol{\nu}}^{I_0} u_{I_0} dx_{I_0} = \int_{F_0} \sum_{F \,\|\, e_{I_0}} \alpha_F \Phi_F^{I_0} dx_{I_0}.$$

The hat functions selects the face F_0 from the sum:

$$\int_{F_0} \mathcal{I}_{\boldsymbol{\nu}} \omega = \int_{F_0} \alpha_{F_0} \prod_{i \in I_0} B_{\nu_i q_i} dx_{I_0}$$

and our definition of box functions gives

$$\int_{F_0} \mathcal{I}_{\boldsymbol{\nu}}\omega = \alpha_{F_0} := \int_{F_0} \omega \,.$$

Hence, the interpolation operator introduced in (3.15) is the correct nodal interpolation operator.

The commuting diagram property (2.8) reads here

$$\mathbf{d} \circ \mathcal{I}_{\boldsymbol{\nu}} = \mathcal{I}_{\boldsymbol{\nu}} \circ \mathbf{d} \,, \tag{3.17}$$

where the last operator interpolates $(l+1)$-forms on the full grid. Let us prove it. By the definition of the exterior derivative operator, we have

$$\mathbf{d}\,(\mathcal{I}_{\boldsymbol{\nu}}\omega) = \sum_I \sum_{k \notin I} \frac{\partial}{\partial x_k} \mathcal{I}^I_{\boldsymbol{\nu}} u_I \; dx_k \wedge dx_I \,,$$

whereas

$$\mathcal{I}_{\boldsymbol{\nu}}(\mathbf{d}\omega) = \sum_I \sum_{k \notin I} \mathcal{I}^{I\cup k}_{\boldsymbol{\nu}} \left(\frac{\partial}{\partial x_k} u_I \right) \; dx_k \wedge dx_I \,.$$

It remains to show that

$$\frac{\partial}{\partial x_k} \mathcal{I}^I_{\boldsymbol{\nu}} u_I = \mathcal{I}^{I\cup k}_{\boldsymbol{\nu}} \left(\frac{\partial}{\partial x_k} u_I \right) \,,$$

which follows from the definition (3.15) of $\mathcal{I}^I_{\boldsymbol{\nu}} u_I$, with the degrees of freedom given by (3.16) and using (3.11).

Note that $\mathcal{I}^I_{\boldsymbol{\nu}} u_I$ can be seen as compound of one-dimensional interpolation operators: $\mathcal{I}^{\mathrm{const}}_{\nu_i}$ — piecewise constant in directions from I and $\mathcal{I}^{\mathrm{lin}}_{\nu_j}$ — piecewise linear for the other directions:

$$\mathcal{I}^I_{\boldsymbol{\nu}} = \mathop{\circ}_{i \in I} \mathcal{I}^{\mathrm{const}}_{\nu_i} \mathop{\circ}_{j \notin I} \mathcal{I}^{\mathrm{lin}}_{\nu_j} . \tag{3.18}$$

For 0-forms (Lagrangian finite elements) we have the nodal representation

$$u_{\mathbf{n}}(\mathbf{x}) = \sum_{q_1=0}^{2^n} \dots \sum_{q_d=0}^{2^n} u_{\mathbf{nq}} \Phi^{\emptyset}_{\mathbf{nq}}(\mathbf{x}), \quad u_{\mathbf{nq}} = u(x_{\mathbf{nq}}) \tag{3.19}$$

and the hierarchical

$$u_{\mathbf{n}}(\mathbf{x}) = \sum_{|\boldsymbol{\nu}|_\infty \le n} u_{\boldsymbol{\nu}}^W = \sum_{|\boldsymbol{\nu}|_\infty \le n} \sum_{\mathbf{q} \in \boldsymbol{\Theta}_{\mathbf{n}}} v_{\nu \mathbf{q}} \Phi_{\boldsymbol{\nu}\mathbf{q}}^{\emptyset}(\mathbf{x}) \tag{3.20}$$

where $\mathbf{n} = (n, \dots, n)$ and

$$\begin{aligned} \boldsymbol{\Theta}_{\boldsymbol{\nu}} = \{\mathbf{q} = (q_1, \dots, q_d),\ q_i &= 1, 3, \dots,\ 2^{\nu_i} - 1 \text{ if } \nu_i > 0, \text{ and} \\ q_i &= 0, 1 \text{ if } \nu_i = 0\} \end{aligned}$$

(at level 0 we consider the indices corresponding to points sitting on the boundary). Bungartz ([23, Lemma 2.4], [21, Lemma 2.2]) investigated this case and showed the representation formula

$$v_{\boldsymbol{\nu}\mathbf{q}} = \int_\Omega \prod_{j=1}^d \bar{\varphi}_{\nu_j q_j}(\mathbf{x}) \frac{\partial^{2d} u(\mathbf{x})}{\partial x_1^2 \dots \partial x_d^2} d\mathbf{x}\,, \tag{3.21}$$

as well as estimates for hierarchical coefficients. The resulting hierarchical contributions satisfy

$$\|u_{\boldsymbol{\nu}}^W\|_{L^\infty(\Omega)} \le 2^{-d} 2^{-2|\boldsymbol{\nu}|_1} \cdot \left\| \frac{\partial^{2d} u}{\partial x_1^2 \dots \partial x_d^2} \right\|_{L^\infty(\Omega)}, \tag{3.22}$$

$$\|u_{\boldsymbol{\nu}}^W\|_{L^2(\Omega)} \le 3^{-d} 2^{-2|\boldsymbol{\nu}|_1} \cdot \left\| \frac{\partial^{2d} u}{\partial x_1^2 \dots \partial x_d^2} \right\|_{L^2(\Omega)}, \tag{3.23}$$

which yields (see [23, Lemma 3.1]) error estimates for the interpolation on a full grid:

$$\|u - u_n\|_{L^\infty(\Omega)} = O(2^{-2n}), \qquad \|u - u_n\|_{L^2(\Omega)} = O(2^{-2n}). \tag{3.24}$$

For the interpolant on sparse grids defned by

$$\tilde{u}_n := \sum_{|\boldsymbol{\nu}|_1 \le n+d-1} u_{\boldsymbol{\nu}}^W$$

it can be proved (see [23, Theorem 3.1]Theorem 3.1):

$$\|u - \tilde{u}_n\|_{L^\infty(\Omega)} = O(2^{-2n} \cdot n^{d-1}), \qquad \|u - \tilde{u}_n\|_{L^2(\Omega)} = O(2^{-2n} \cdot n^{d-1}). \tag{3.25}$$

Bungartz showed that the sparse grid interpolation operator is even optimal for the energy semi-norm in $H_0^1(\Omega)$ (order 2^{-n}). We will obtain all these results as the particular case $l = 0$.

Now, let us illustrate the hierarchical decomposition technique in the case of 1-forms in two dimensions. Consider the continuous 1-form

$$\omega = u_1(\mathbf{x})dx_1 + u_2(\mathbf{x})dx_2$$

with the interpolant

$$\mathcal{I}_{\boldsymbol{\nu}}\omega = \left(\mathcal{I}_{\boldsymbol{\nu}}^1 u_1(\mathbf{x})\right) dx_1 + \left(\mathcal{I}_{\boldsymbol{\nu}}^2 u_2(\mathbf{x})\right) dx_2.$$

The vector representative u_1 (corresponding to the direction 1) is approximated on level $\boldsymbol{\nu}$ by

$$\mathcal{I}_{\boldsymbol{\nu}}^1 u_1(\mathbf{x}) = \sum_{k_1=0}^{2^{\nu_1}-1} \sum_{k_2=0}^{2^{\nu_2}} \alpha_{\boldsymbol{\nu}\mathbf{k}} B_{\nu_1 k_1}(x_1) \varphi_{\nu_2 k_2}(x_2)$$

with the scaling coefficients

$$\alpha_{\boldsymbol{\nu}\mathbf{k}} = \alpha(x_{\nu_1 k_1}, x_{\nu_2 k_2}) = \int\limits_{k_1 h_{\nu_1}}^{(k_1+1)h_{\nu_1}} u_1(x_1, k_2 h_{\nu_2}) dx_1.$$

Change the order of summation

$$\mathcal{I}_{\boldsymbol{\nu}}^1 u_1(\mathbf{x}) = \sum_{k_2=0}^{2^{\nu_2}} \left(\sum_{k_1=0}^{2^{\nu_1}-1} \alpha_{\boldsymbol{\nu}\mathbf{k}} B_{\nu_1 k_1}(x_1) \right) \varphi_{\nu_2 k_2}(x_2)$$

and apply the Haar-wavelet transform to the interior representations (see equations (3.5) and (3.7)):

$$\mathcal{I}_{\boldsymbol{\nu}}^1 u_1(\mathbf{x}) = \sum_{k_2=0}^{2^{\nu_2}} \left(\sum_{m_1=0}^{\nu_1} \sum_{q_1 \in \Theta_{m_1}} \tilde{\alpha}_{(m_1,\nu_2)(q_1,k_2)} \psi_{m_1 q_1}(x_1) \right) \varphi_{\nu_2 k_2}(x_2).$$

Change again the order of summation:

$$\mathcal{I}_{\boldsymbol{\nu}}^1 u_1(\mathbf{x}) = \sum_{m_1=0}^{\nu_1} \sum_{q_1 \in \Theta_{m_1}} \left(\sum_{k_2=0}^{2^{\nu_2}} \tilde{\alpha}_{(m_1,\nu_2)(q_1,k_2)} \varphi_{\nu_2 k_2}(x_2) \right) \psi_{m_1 q_1}(x_1)$$

and apply the hierarchical transformation to the interior representations (see equations (3.2) and (3.3)):

$$\mathcal{I}^1_{\boldsymbol{\nu}} u_1(\mathbf{x}) = \sum_{m_1=0}^{\nu_1} \sum_{q_1\in\Theta_{m_1}} \left(\sum_{m_2=0}^{\nu_2} \sum_{q_2\in\Theta_{m_2}} \beta_{(m_1,m_2)(q_1,q_2)} \varphi_{m_2q_2}(x_2) \right) \psi_{m_1q_1}(x_1).$$

Tersely speaking, the hierarchical representation is obtained from the wavelet transform in one direction followed by the hierarchical transformation in the other. Let us examine how the hierarchical coefficients $\beta_{\mathbf{mq}}$, with $\mathbf{m} = (m_1, m_2)$, and $\mathbf{q} = (q_1, q_2)$, can be represented. The "nodal" coefficients are the degrees of freedom

$$\alpha_{\boldsymbol{\nu}\mathbf{k}} = \int\limits_{k_1h_{\nu_1}}^{(k_1+1)h_{\nu_1}} u_1(x_1, k_2h_{\nu_2})dx_1.$$

After the wavelet transform in the first direction, we obtain cf. (3.8)

$$\tilde{\alpha}_{(m_1,\nu_2)(q_1,k_2)} = \int_0^1 u_1(x_1, k_2h_{\nu_2})\psi_{m_1q_1}(x_1)dx_1.$$

The function $\int_0^1 u_1(x_1, x_2)\psi_{m_1q_1}(x_1)dx_1$ is smooth in x_2, so we may apply the hierarchical transformation in the second direction, as well as the representation formula (3.21):

$$\beta_{\mathbf{mq}} = \int_0^1 \frac{\partial^2}{\partial x_2^2} \left(\int_0^1 u_1(x_1, x_2)\psi_{m_1q_1}(x_1)dx_1 \right) \varphi_{m_2q_2}(x_2)dx_2 \ .$$

As the Haar-wavelet is the derivative of a hat function, by (3.12) we have (using integration by parts)

$$\beta_{\mathbf{mq}} = \int_0^1 \frac{\partial^2}{\partial x_2^2} \left(2^{-(\nu_1+1)/2} \int_0^1 \frac{\partial u_1(x_1, x_2)}{\partial x_1} \varphi_{m_1q_1}(x_1)dx_1 \right) \varphi_{m_2q_2}(x_2)dx_2 \ ,$$

which yields

$$\beta_{\mathbf{mq}} = 2^{-(\nu_1+1)/2} \int\limits_{\Omega} \varphi_{m_1q_1}(x_1)\varphi_{m_2q_2}(x_2) \frac{\partial^3}{\partial x_1 \partial x_2^2} u_1(\mathbf{x})d\mathbf{x} \ .$$

We proceed with the same steps in deriving the hierarchical representation for the general case of l-forms in d-dimensions. Take $I = \{i_1, \ldots, i_l\} \subset \{1, \ldots, d\}$, the complement index set $J = \{1, \ldots, d\} \setminus I$ and denote by $\mathbf{k}_I = \{k_i \ : \quad i \in I\}$ the indices corresponding to I. The interpolant of the I-component of a smooth differential form can be written as

$$\mathcal{I}^I_{\boldsymbol{\nu}} u_I = \sum_{\mathbf{k}_J} \left(\sum_{\mathbf{k}_I} \alpha_{\boldsymbol{\nu}\mathbf{k}} \prod_{i\in I} B_{\nu_i k_i}(x_i) \right) \prod_{j \in J} \varphi_{\nu_j k_j}(x_j)$$

with the degree of freedom corresponding to the face $F = [\mathbf{x}_{\boldsymbol{\nu}\mathbf{k}}; e_I]$

$$\alpha_{\boldsymbol{\nu}\mathbf{k}} = \int\limits_{[0,1]^l} u_I(\mathbf{x}_I, \mathbf{x}_{\nu_J k_J}) \prod_{i \in I} B_{\nu_i k_i}(x_i) d\mathbf{x}_I.$$

The wavelet transform in l dimensions for the interior representations gives

$$\mathcal{I}^I_{\boldsymbol{\nu}} u_I = \sum_{\mathbf{k}_J} \left(\sum_{\mathbf{m}_I} \sum_{\mathbf{q}_I} \tilde{\alpha}_{(\mathbf{m}_I \boldsymbol{\nu}_J)(\mathbf{q}_I \mathbf{k}_J)} \prod_{i\in I} \psi_{m_i q_i}(x_i) \right) \prod_{j \in J} \varphi_{\nu_j k_j}(x_j).$$

Change the order of summation and apply the hierarchical transform, in order to obtain

$$\mathcal{I}^I_{\boldsymbol{\nu}} u_I = \sum_{\mathbf{m}_I} \sum_{\mathbf{q}_I} \left(\sum_{\mathbf{m}_J} \sum_{\mathbf{q}_J} \beta_{(\mathbf{m}_I \mathbf{m}_J)(\mathbf{q}_I \mathbf{q}_J)} \prod_{j\in J} \varphi_{m_j q_j}(x_j) \right) \prod_{i \in I} \psi_{m_i q_i}(x_i).$$

The coefficients $\tilde{\alpha}$ are given by

$$\tilde{\alpha}_{(\mathbf{m}_I \boldsymbol{\nu}_J)(\mathbf{q}_I \mathbf{k}_J)} = \int\limits_{[0,1]^l} u_I(\mathbf{x}_I, \mathbf{x}_{\nu_J k_J}) \prod_{i \in I} \psi_{m_i q_i}(x_i) d\mathbf{x}_I\, .$$

The function $\int_{[0,1]^l} u_I(\mathbf{x}_I, \mathbf{x}_J) \prod_{i\in I} \psi_{m_i q_i}(x_i) d\mathbf{x}_I$ being smooth in $\mathbf{x}_J$, we obtain by means of (3.21)

$$\beta_{(\mathbf{m}_I \mathbf{m}_J)(\mathbf{q}_I \mathbf{q}_J)} = \int\limits_{[0,1]^{d-l}} \prod_{j \in J} \frac{\partial^2}{\partial x_j^2} \left(\int\limits_{[0,1]^l} u_I(\mathbf{x}) \prod_{i\in I} \psi_{m_i q_i}(x_i) d\mathbf{x}_I \right) \prod_{j \in J} \varphi_{m_j q_j}(x_j) d\mathbf{x}_J.$$

Integrating by parts, we get the representation formula for the hierarchical coefficients $\beta_{\mathbf{mq}}$.

Lemma 3.2.1. *The hierarchical coefficients of the I-component of a Whitney l-form in d dimensions are*

$$\beta_{\mathbf{mq}} = 2^{-(|\mathbf{m}|_1+l)/2} \int\limits_{\Omega} \prod_{i\in I} \varphi_{m_i q_i}(x_i) \prod_{j\notin I} \bar{\varphi}_{m_j q_j}(x_j) D^I u_I(\mathbf{x}) d\mathbf{x} \,, \tag{3.26}$$

where the operator D^I is

$$D^I = \prod_{i\in I} \frac{\partial}{\partial x_i} \prod_{j\notin I} \frac{\partial^2}{\partial x_i^2} = \frac{\partial^{l+2(d-l)}}{\partial x_{i_1} \ldots \partial x_{i_l} \partial x_{j_1}^2 \ldots \partial x_{j_{d-l}}^2} \,,$$

and the indices m_i and q_i satisfy

$$\begin{aligned} 0 \le m_i \le \nu_i, &\quad \textit{for } i = 1,2,\ldots,d \\ q_i \in \Theta_{m_i}, &\quad \textit{for } i = 1,2,\ldots,d \,. \end{aligned}$$

The above construction provides the **hierarchical basis functions**

$$\Psi^I_{\boldsymbol{\nu}\mathbf{q}} := \prod_{i\in I} \psi_{\nu_i q_i} \prod_{j\notin I} \varphi_{\nu_j q_j}, \quad \mathbf{q} \in \boldsymbol{\Theta}_{\boldsymbol{\nu}} := \Theta_{\nu_1} \times \cdots \times \Theta_{\nu_d} \,. \tag{3.27}$$

Thus, the **hierarchical surpluses** (detail approximation functions) belonging to u_I are

$$\mathcal{H}^I_{\boldsymbol{\nu}} u_I := \sum_{\mathbf{q}\in\boldsymbol{\Theta}_{\nu}} \beta_{\boldsymbol{\nu}\mathbf{q}} \Psi^I_{\boldsymbol{\nu}\mathbf{q}} \,. \tag{3.28}$$

The coefficients $\beta_{\boldsymbol{\nu}\mathbf{q}}$ arise from Haar-wavelet transform in directions from I followed by hierarchical transform in the other directions. The detail spaces are

$$W^I_{\boldsymbol{\nu}} := \mathrm{Range}(\mathcal{H}^I_{\boldsymbol{\nu}}).$$

Then, the **hierarchical interpolation operator** is defined as

$$\mathcal{H}_{\boldsymbol{\nu}} \omega := \sum_I \left(\mathcal{H}^I_{\boldsymbol{\nu}} u_I \right) dx_I.$$

For the resolution $n \in \mathbb{N}$, denote $\mathbf{n} = (n, \ldots, n)$, $|\boldsymbol{\nu}|_\infty = \max\{\nu_1, \ldots \nu_d\}$, $|\boldsymbol{\nu}|_1 = \sum_i \nu_i$. The above construction means for the full grid interpolation operator

$$\mathcal{I}^I_{\mathbf{n}} = \sum_{|\boldsymbol{\nu}|_\infty \le n} \mathcal{H}^I_{\boldsymbol{\nu}}, \quad \mathcal{I}_{\mathbf{n}} = \sum_{|\boldsymbol{\nu}|_\infty \le n} \mathcal{H}_{\boldsymbol{\nu}}.$$

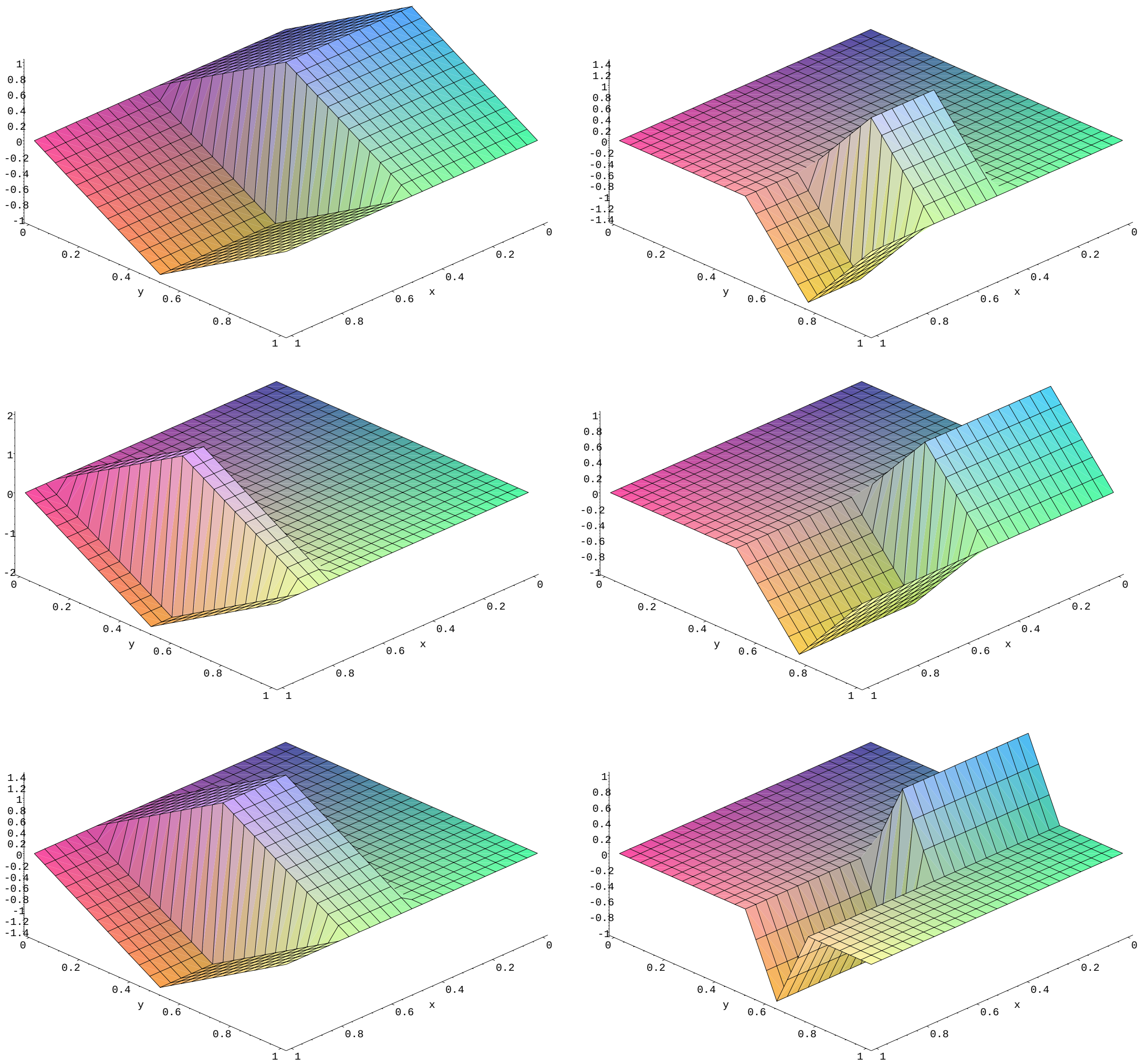


Figure 3.6: Hierarchical basis functions for the first component of an 1-form ($d = 2$, $n = 3$)

The **hierarchical decomposition** of the full grid interpolation space corresponding to the I-component is

$$V_{\mathbf{n}}^{I,l} = \bigoplus_{|\boldsymbol{\nu}|_\infty \leq n} W_{\boldsymbol{\nu}}^I. \tag{3.29}$$

By virtue of the tensor product approach, it is easy to see (cf. Figure 3.7) that the hierarchical interpolation operator is a combination of nodal interpolation operators on anisotropic full grids

$$\mathcal{H}_{\boldsymbol{\nu}}^I = \sum_{\boldsymbol{\epsilon} \in \{0,1\}^d} (-1)^{|\boldsymbol{\epsilon}|_1} \mathcal{I}_{\boldsymbol{\nu}-\boldsymbol{\epsilon}}^I \quad , \quad \mathcal{H}_{\boldsymbol{\nu}} = \sum_{\boldsymbol{\epsilon} \in \{0,1\}^d} (-1)^{|\boldsymbol{\epsilon}|_1} \mathcal{I}_{\boldsymbol{\nu}-\boldsymbol{\epsilon}}. \tag{3.30}$$

In detail, this means

$$\begin{aligned} \mathcal{H}_{\boldsymbol{\nu}}^I &= \mathcal{I}_{\boldsymbol{\nu}}^I - \mathcal{I}_{\boldsymbol{\nu}-(1,0,\ldots,0)}^I - \mathcal{I}_{\boldsymbol{\nu}-(0,1,\ldots,0)}^I - \ldots - \mathcal{I}_{\boldsymbol{\nu}-(0,\ldots,0,1)}^I + \\ &\quad + \mathcal{I}_{\boldsymbol{\nu}-(1,1,0,\ldots,0)}^I + \mathcal{I}_{\boldsymbol{\nu}-(1,0,1,\ldots,0)}^I + \ldots + (-1)^d \mathcal{I}_{\boldsymbol{\nu}-(1,1,\ldots,1)}^I. \end{aligned}$$

Together with the commuting diagram property for the “nodal” interpolation operator on anisotropic tensor product meshes (2.8), this yields, for sufficiently smooth forms:

$$\mathbf{d} \circ \mathcal{H}_{\boldsymbol{\nu}} = \mathcal{H}_{\boldsymbol{\nu}} \circ \mathbf{d}\,, \tag{3.31}$$

i.e. the commuting diagram property holds for the hierarchical interpolation operator. This is important for the estimation the interpolation error in $\mathbf{H}(d, \Omega)$.

In the figure 3.7 we sketch some scale and detail spaces together with basis functions corresponding to the first component of differential 1-forms in two dimensions.

Equations (3.4) and (3.9) provide estimates of the norms of basis functions

$$\left\| \Psi_{\boldsymbol{\nu}\mathbf{q}}^I \right\|_{L^2(\Omega)}^2 = \left(\frac{2}{3} \right)^{d-l} 2^{-|\boldsymbol{\nu}_{I'}|_1} \tag{3.32}$$

where $|\boldsymbol{\nu}_I|_1 := \sum_{i \in I} \nu_i$ and $|\boldsymbol{\nu}_{I'}|_1 := \sum_{i \notin I} \nu_i$.

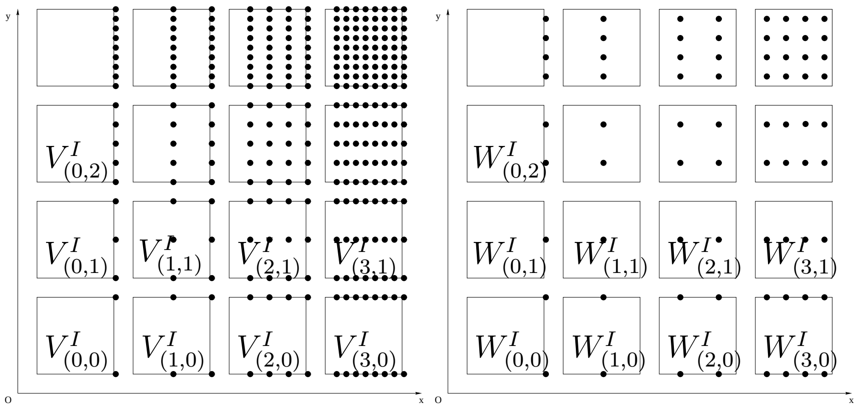


Figure 3.7: Scale and detail spaces for first component of a 1-form ($d = 2$, $n = 3$,$I = \{1\}$). Each box function is identified with the grid point sitting at the right end of its support (edge). Each Haar-wavelet and each hat function is associated with the grid point located in the middle of its support.

We have seen how the hierarchical multilevel principle gives rise to hierarchical base for Whitney l-forms in d dimensions. Selecting particular basis functions, we obtain the sparse grid spaces in the next chapter. Then, we investigate the properties of the sparse grid interpolation operator.

Chapter 4

Interpolation on Sparse Grids

Based on the hierarchical basis, we define the sparse grid interpolation operator. We prove the commuting diagram property and the existence of discrete sparse potentials. Then, we estimate the hierarchical contributions of detail spaces. The error estimate for the interpolation justifies the choice of the sparse grid. Approximate interpolation is needed for the Galerkin scheme in partial differential equations. The combination technique and a two point quadrature rule ensure that a similar error estimate as for the exact interpolation holds.

4.1 Interpolation Operator

In the previous chapter we have introduced the hierarchical basis for Whitney l-forms in d dimensions. We established the hierarchical decomposition of the full grid interpolation space (3.29). Now, we are in a position to define the sparse grid interpolation space and the associated interpolation operator. This is done in full analogy with the sparse grid interpolation space for Lagrangian finite elements.

As mentioned in the previous section, we mainly deal with differential forms through the components of their vector proxies. In a sense, a sparse Whitney l-form $\omega_n \in V_n^l$ will arise by discretizing each of the $\binom{d}{l}$ components of the vector proxies separately.

Definition 4.1.1. *The space for the I-component of the vector proxy corre-*

sponding to sparse Whitney l-forms at resolution n is defined by

$$V_n^I := \bigoplus_{|\boldsymbol{\nu}|_1 \le n+d-1} W_{\boldsymbol{\nu}}^I\,, \tag{4.1}$$

where $|\boldsymbol{\nu}|_1 = \nu_1 + \nu_2 + \cdots + \nu_d$.

Definition 4.1.2. *The* **sparse grid interpolation operator** *is defined by*

$$\mathcal{I}_n^I := \sum_{|\boldsymbol{\nu}|_1 \le n+d-1} \mathcal{H}_{\boldsymbol{\nu}}^I, \qquad \mathcal{I}_n := \sum_{|\boldsymbol{\nu}|_1 \le n+d-1} \mathcal{H}_{\boldsymbol{\nu}}. \tag{4.2}$$

Definition 4.1.3. *We call* $V_n^l :=$ Range$(\mathcal{I}_n)$ *the space of Whitney-l-forms on a sparse grid of resolution n in d dimensions.*

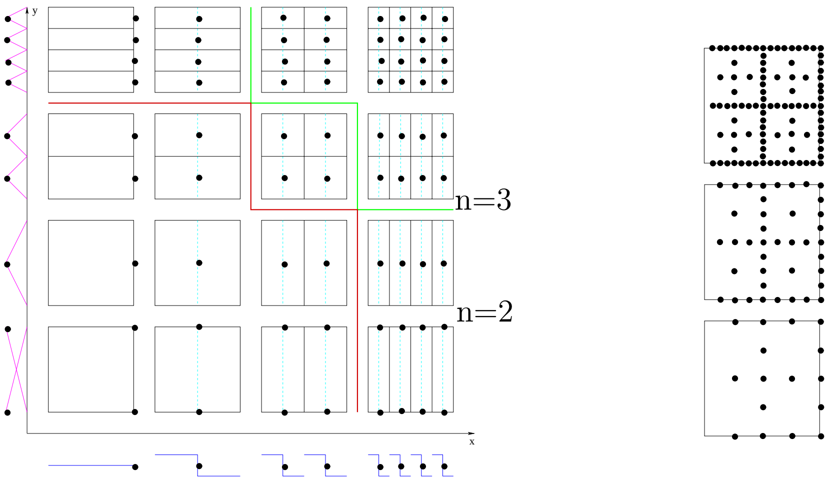


Figure 4.1: Supports of basis functions generating sparse grid spaces ($d = 2$, $n = 2, 3$ and $l = 1$)

In Figure 4.1 we sketch the detail spaces together with basis functions corresponding to the first component of differential 1-forms in two dimensions.

Firstly, the combination formula for the Lagrangian elements on sparse grids can be easily extended to the Whitney forms. The idea is to represent any sparse grid function as a linear combination of its interpolants on the regular (anisotrop) full grids $\Omega_{\boldsymbol{\nu}}$ with $|I|_1 = n + d - s$, for $s = 1, \ldots, d$.

Proposition 4.1.1. *The sparse grid interpolation operator can be written over the combination formulas:*

$$\mathcal{I}_n^I u_I = \sum_{s=1}^{d} (-1)^{s-1} \binom{d-1}{s-1} \sum_{|\boldsymbol{\nu}|_1 = n+d-s} \mathcal{I}_{\boldsymbol{\nu}}^I u_I \tag{4.3}$$

and

$$\mathcal{I}_n = \sum_{s=1}^{d} (-1)^{s-1} \binom{d-1}{s-1} \sum_{|\boldsymbol{\nu}|_1 = n+d-s} \mathcal{I}_{\boldsymbol{\nu}} \,. \tag{4.4}$$

Proof. These formulas are consequences of the definition of the sparse grid interpolation operator and of the remark that

$$\mathcal{H}_{\boldsymbol{\nu}}^I = \sum_{\boldsymbol{\epsilon} \in \{0,1\}^d} (-1)^{|\boldsymbol{\epsilon}|_1} \mathcal{I}_{\boldsymbol{\nu}-\boldsymbol{\epsilon}}^I \quad \text{and} \quad \mathcal{H}_{\boldsymbol{\nu}} = \sum_{\boldsymbol{\epsilon} \in \{0,1\}^d} (-1)^{|\boldsymbol{\epsilon}|_1} \mathcal{I}_{\boldsymbol{\nu}-\boldsymbol{\epsilon}} \,. \tag{4.5}$$

Indeed, (4.5) means in detail

$$\begin{aligned} \mathcal{H}_{\boldsymbol{\nu}} &= \mathcal{I}_{\boldsymbol{\nu}} - \mathcal{I}_{\boldsymbol{\nu}-(1,0,\dots,0)} - \mathcal{I}_{\boldsymbol{\nu}-(0,1,\dots,0)} - \dots - \mathcal{I}_{\boldsymbol{\nu}-(0,\dots,0,1)} + \\ &\quad + \mathcal{I}_{\boldsymbol{\nu}-(1,1,0,\dots,0)} + \mathcal{I}_{\boldsymbol{\nu}-(1,0,1,\dots,0)} + \dots + (-1)^d \mathcal{I}_{\boldsymbol{\nu}-(1,1,\dots,1)} \,. \end{aligned}$$

We replace it in the definition of the sparse grid interpolation operator (4.2) and rearrange the summation terms

$$\begin{aligned} \mathcal{I}_n &= \sum_{|\boldsymbol{\nu}|_1 \le n+d-1} \mathcal{H}_{\boldsymbol{\nu}} = \sum_{s=1}^{d} \sum_{|\boldsymbol{\nu}|_1 = n+d-s} \mathcal{H}_{\boldsymbol{\nu}} \\ &= \sum_{s=1}^{d} \sum_{|\boldsymbol{\nu}|_1 = n+d-s} \sum_{\boldsymbol{\epsilon} \in \{0,1\}^d} (-1)^{|\boldsymbol{\epsilon}|_1} \mathcal{I}_{\boldsymbol{\nu}-\boldsymbol{\epsilon}} = \sum_{s=1}^{d} c(s) \sum_{|\boldsymbol{\nu}|_1 = n+d-s} \mathcal{I}_{\boldsymbol{\nu}} \,. \end{aligned}$$

The coefficient $c(s)$ is

$$c(s) = (-1)^{s-1} \binom{d}{s-1} + (-1)^{s-2} \binom{d}{s-2} + \dots + (-1) \binom{d}{1} + \binom{d}{0} ,$$

that is, by a known identity for binomial coefficients [53],

$$c(s) = \sum_{k \le s-1} (-1)^k \binom{d}{k} = (-1)^{s-1} \binom{d-1}{s-1} .$$

In conclusion, the identities (4.4) and (4.3) hold. □

Fig. 4.2 shows the involved grids in the case of an 1-form in two dimensions on a sparse grid of resolution $n = 3$:

$$\mathcal{I}_3 = \mathcal{I}_{(1\,3)} + \mathcal{I}_{(2\,2)} + \mathcal{I}_{(3\,1)} - \mathcal{I}_{(1\,2)} - \mathcal{I}_{(2\,1)} \,.$$

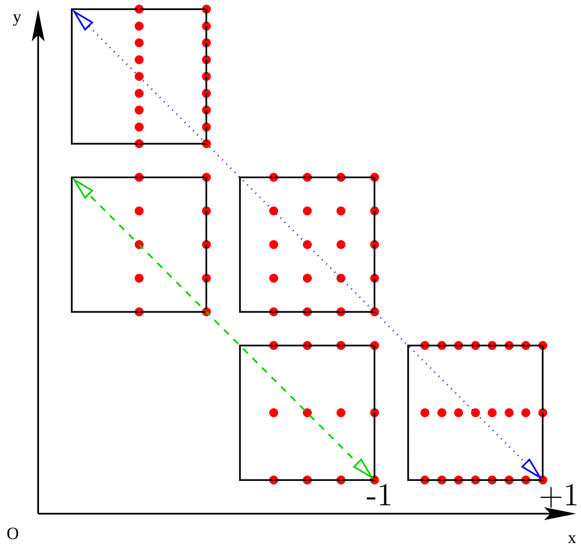


Figure 4.2: Grids involved in combination technique for $d = 2$, $l = 1$, $n = 3$, $I = \{1\}$

Secondly, we show that the two principal properties of the Whitney elements, i.e. the commuting diagram and the exact sequence properties, still hold in the sparse grid case.

Proposition 4.1.2. *The commuting diagram property*

$$\mathbf{d} \circ \mathcal{I}_n = \mathcal{I}_n \circ \mathbf{d} \tag{4.6}$$

and the exact sequence property

$$V_n^l \cap ker\{\mathbf{d}\} = \mathbf{d}V_n^{l-1} \tag{4.7}$$

hold true for the sparse grid interpolation operator and sufficiently smooth differential forms.

Proof. The first assertion follows from the definition of the sparse grid interpolation operator (4.2) and the commuting diagram property for the hierarchical interpolation operator (3.31).

We focus on the second assertion. As the derivative of the hat function is a Haar-wavelet, the definitions of V_n^l and of the exterior derivative $\mathbf{d}$ instantly provide

$$V_n^l \cap \ker\{\mathbf{d}\} \supset \mathbf{d}V_n^{l-1}\,.$$

We prove the reverse inclusion. Consider a closed Whitney l-form ω_n, that is $\mathbf{d}\omega_n = 0$. We construct a discrete potential η_n on the sparse grid, that is $\eta_n \in V_n^{l-1}$ and $\mathbf{d}\eta_n = \omega$. Using the combination formula (4.4), we can wite ω_n as

$$\omega_n = \mathcal{I}_n\omega_n = \sum_{s=1}^{d}(-1)^{s-1}\binom{d-1}{s-1}\sum_{|\boldsymbol{\nu}|_1=n+d-s}\mathcal{I}_{\boldsymbol{\nu}}\omega_n\,.$$

The commuting diagram for anisotrop full grids (3.17) gives

$$\mathbf{d}\left(\mathcal{I}_{\boldsymbol{\nu}}\omega_n\right) = \mathcal{I}_{\boldsymbol{\nu}}(\mathbf{d}\omega_n) = 0\,,$$

so, by the existence of the discrete potentials for the full grids, there is $\eta_{\boldsymbol{\nu}} \in V_{\boldsymbol{\nu}}^{l-1}$ such that $\mathbf{d}\eta_{\boldsymbol{\nu}} = \mathcal{I}_{\boldsymbol{\nu}}\omega_n$. We construct the discrete $(l-1)$-form η on the sparse grid by means of the combination technique and the obtained $\eta_{\boldsymbol{\nu}}$:

$$\eta_n := \sum_{s=1}^{d}(-1)^{s-1}\binom{d-1}{s-1}\sum_{|\boldsymbol{\nu}|_1=n+d-s}\eta_{\boldsymbol{\nu}}\,.$$

Clearly $\eta_n \in V_n^{l-1}$ and $\mathbf{d}\eta_n = \omega_n$, which completes the proof. □

Thirdly, the representation formula of hierarchical coefficients (3.26) allows us to estimate the hierarchical contributions of detail spaces.

Lemma 4.1.1. *The hierarchical coefficients decay according to*

$$|\beta_{\boldsymbol{\nu}\mathbf{q}}| \;\leq\; 2^{-l}3^{-\frac{d}{2}}\cdot 2^{-(|\boldsymbol{\nu}_I|_1-l)}\cdot 2^{-(3|\boldsymbol{\nu}_{I'}|_1-d+l)/2}\left\|D^I u_I\right\|_{L^2(\mathrm{supp}(\psi^I_{\boldsymbol{\nu}\mathbf{q}}))}\,. \tag{4.8}$$

Proof. Equations (3.27), (3.32) and the definition of $\bar{\varphi}_{\nu_j q_j}$, (3.1) lead to (4.8). □

Proposition 4.1.3. *The L^2-norms of details are bounded by*

$$\left\|\mathcal{H}^I_{\boldsymbol{\nu}}u_I\right\|_{L^2(\Omega)} \;\leq\; 3^{-(d-l/2)}\cdot 2^{-2|\boldsymbol{\nu}_{I'}|_1-|\boldsymbol{\nu}_I|_1}\cdot\left\|D^I u_I\right\|_{L^2(\Omega)}\,. \tag{4.9}$$

Proof. The hierarchical basis functions corresponding to a fixed level have disjoint supports, and equation (3.32) gives:

$$\begin{aligned}
\left\|\mathcal{H}^I_{\boldsymbol{\nu}} u_I\right\|^2_{L^2(\Omega)} &= \left\|\sum_{\mathbf{q}\in\boldsymbol{\Theta}_{\boldsymbol{\nu}}} \beta^I_{\boldsymbol{\nu}\mathbf{q}} \boldsymbol{\Psi}^I_{\boldsymbol{\nu}\mathbf{q}}\right\|^2_{L^2(\Omega)} = \sum_{\mathbf{q}\in\boldsymbol{\Theta}_{\boldsymbol{\nu}}} |\beta^I_{\boldsymbol{\nu}\mathbf{q}}|^2 \cdot \left\|\boldsymbol{\Psi}^I_{\boldsymbol{\nu}\mathbf{q}}\right\|^2_{L^2(\Omega)} = \\
&= \sum_{\mathbf{q}\in\boldsymbol{\Theta}_{\boldsymbol{\nu}}} |\beta^I_{\boldsymbol{\nu}\mathbf{q}}|^2 \cdot \left(\frac{2}{3}\right)^{(d-l)} 2^{-|\boldsymbol{\nu}_{I'}|_1} \\
&\leq \sum_{\mathbf{q}\in\boldsymbol{\Theta}_{\boldsymbol{\nu}}} 3^{-l} 9^{-(d-l)} \cdot 2^{-4|\boldsymbol{\nu}_{I'}|_1 - 2|\boldsymbol{\nu}_I|_1} \cdot \left\|D^I u_I|\mathrm{supp}(\psi^I_{\boldsymbol{\nu}\mathbf{q}})\right\|^2_{L^2(\Omega)} ,
\end{aligned}$$

where we used (4.8). So

$$\left\|\mathcal{H}^I_{\boldsymbol{\nu}} u_I\right\|^2_{L^2(\Omega)} \leq 3^{-l} 9^{-(d-l)} \cdot 2^{-4|\boldsymbol{\nu}_{I'}|_1 - 2|\boldsymbol{\nu}_I|_1} \cdot \left\|D^I u_I\right\|^2_{L^2(\Omega)} ,$$

which proves the inequality (4.9). □

The commuting diagram property and the error estimates for the L^2-norm are all the tools needed to derive error estimates in other norms:

$$\begin{aligned}
\|\omega - \mathcal{I}_n\omega\|^2_{\mathbf{H}(d,\Omega)} &= \|\omega - \mathcal{I}_n\omega\|^2_{L^2(\Omega)} + \|\mathbf{d}(\omega - \mathcal{I}_n\omega)\|^2_{L^2(\Omega)} = \\
&= \|\omega - \mathcal{I}_n\omega\|^2_{L^2(\Omega)} + \|\mathbf{d}\omega - \mathcal{I}_n\mathbf{d}\omega\|^2_{L^2(\Omega)} .
\end{aligned}$$

We emphasize the role of the commuting diagram property in the derivation of the error estimate in $\mathbf{H}(\mathbf{d},\Omega)$.

The rationale behind the definition of the sparse grid interpolation (4.1) and (4.2) is *the fast decay of the hierarchical coefficients.* The bulk of detail spaces do not contribute much to the approximation. We may dispose of many details without letting the convergence rate deteriorate too much. Each component of the vector field could be discretized on its optimal sparse grid in the sense of Bungartz [23]. But our aim is conforming finite elements in $\mathbf{H}(\mathbf{d},\Omega)$, so we must discretize all components on the *same grid.* It is not difficult to see that a suitable grid to work on is the classical sparse grid [23, 92]. We take into account only the detail spaces $W^I_{\boldsymbol{\nu}}$ with $|\boldsymbol{\nu}|_1 \leq n+d-1$. The following estimates of the interpolation error justify this.
For the L^2-norm we have by definition

$$\|\omega - \mathcal{I}_n\omega\|^2_{L^2(\Omega)} = \sum_I \left\|u_I - \mathcal{I}^I_n u_I\right\|^2_{L^2(\Omega)} ,$$

and the triangle inequality gives

$$\left\|u_I - \mathcal{I}_n^I u_I\right\|_{L^2(\Omega)} \leq \sum_{|\boldsymbol{\nu}|_1 \geq n+d} \left\|\mathcal{H}_{\boldsymbol{\nu}}^I u_I\right\|_{L^2(\Omega)} .$$

Using the estimates of the hierarchical contributions (4.9), we obtain

$$\begin{aligned} \left\|u_I - \mathcal{I}_n^I u_I\right\|_{L^2(\Omega)} &\leq \sum_{|\boldsymbol{\nu}|_1 \geq n+d} 3^{-(d-l/2)} \cdot 2^{-2|\boldsymbol{\nu}_{I'}|_1 - |\boldsymbol{\nu}_I|_1} \cdot \left\|D^I u_I\right\|_{L^2(\Omega)} \\ &\leq 3^{-(d-l/2)} \cdot \left\|D^I u_I\right\|_{L^2(\Omega)} \sum_{|\boldsymbol{\nu}|_1 \geq n+d} 2^{-2|\boldsymbol{\nu}|_1 + |\boldsymbol{\nu}_I|_1} . \qquad (4.10) \end{aligned}$$

All depends on the evaluation of the sum

$$S(n,l,d) = \sum_{|\boldsymbol{\nu}|_1 \geq n+d} 2^{-2|\boldsymbol{\nu}|_1 + |\boldsymbol{\nu}_I|_1} = \sum_{k=n+d}^{\infty} 2^{-2k} \sum_{|\boldsymbol{\nu}|_1 = k} 2^{|\boldsymbol{\nu}_I|_1} , \qquad (4.11)$$

which we intend to estimate. To this end, we present three lemmata.

Lemma 4.1.2. *For* $1 \leq n < m$ *it holds*

$$\sum_{k=0}^{n} \binom{m}{k} \leq 2n! \binom{m}{n}.$$

Proof. Induction on n. For $n = 1$ we have

$$\binom{m}{0} + \binom{m}{1} = 1 + m < 2m$$

and the assertion is true.
Suppose that the proposition is true for $n-1$ and prove it for $n \geq 2$. Indeed,

$$\sum_{k=0}^{n} \binom{m}{k} = \sum_{k=0}^{n-1} \binom{m}{k} + \binom{m}{k} \leq 2(n-1)! \binom{m}{n-1} + \binom{m}{k} ,$$

by the induction hypothesis, so

$$\sum_{k=0}^{n} \binom{m}{k} \leq 2\binom{m}{k} \left(\frac{n!}{m-n+1} + \frac{1}{2} \right) ,$$

which is less than $2\binom{m}{k} \cdot n!$, since $n \geq 2$ and $m - n + 1 \geq 2$. Hence, the assertion is true for n. By induction on n, we conclude that the lemma holds. □

Lemma 4.1.3. *It holds*

$$\sum_{\substack{\nu_1+\ldots+\nu_d=k\\ 0\le\nu_i, i=1,\ldots,d}} 1 = \binom{d+k-1}{d-1}.$$

Proof. The lemma is equivalent to a known problem from combinatorics: in how many ways can you place k objects in d boxes, allowing void boxes ? First, remember the answer for the same problem with the requirement that there is no void box, which is just the definition of the combinations with repetitions:

$$\sum_{\substack{\nu_1+\ldots+\nu_d=k\\ 1\le\nu_i, i=1,\ldots,d}} 1 = \binom{k-1}{d-1}$$

If we allow void boxes, then we can leave p, with $0 \le p \le d-1$, void boxes and distribute the k objects in the remaining $d-p$ boxes as before:

$$\sum_{\substack{\nu_1+\ldots+\nu_d=k\\ 0\le\nu_i, i=1,\ldots,d}} 1 = \sum_{p=0}^{d-1} \binom{d}{p}\binom{k-1}{d-p-1},$$

which is just the Vandermonde convolution [53, vol. 1, p. 58]. □

In the following lemma, the conditions on running indices arise by setting the degrees of freedom from the boundary to zero.

Lemma 4.1.4. *For $1 \le l \le d$, it holds*

$$S_k(l,d) := \sum_{\nu_1+\ldots+\nu_d=k} 2^{\nu_1+\ldots+\nu_l} \le 2^k \binom{k+l-1}{l-1},$$

where $0 \le \nu_i$, for $i = 0, \ldots, l$, and $1 \le \nu_j$, for $j = l+1, \ldots, d$.

Proof. We use a backward induction argument, beginning with $l = d$. Indeed, using Lemma 4.1.3, we obtain:

$$S_k(d,d) = \sum_{|\boldsymbol{\nu}|_1=k} 2^k = 2^k \binom{d+k-1}{d-1},$$

and $S_k(d-1,d-1) \leq 2^k\binom{d+k-2}{d-2}$. Now, the next term is

$$\begin{aligned} S_k(d-1,d) &= \sum_{\nu_d=1}^{k} S_{k-\nu_d}(d-1,d-1) \leq \sum_{s=1}^{k} 2^{k-s}\binom{d-2+k-s}{d-2} \\ &\leq \binom{d-2+k-1}{d-2} 2^k \sum_{s=1}^{k} 2^{-s} \leq 2^k\binom{d-2+k-1}{d-2}. \end{aligned}$$

By induction, it is easy to see that

$$S_k(d-t,d) \leq 2^k\binom{d-t-1+k-t}{d-t-1} \qquad \text{for } 1 \leq t \leq d-1$$

so

$$S_k(l,d) \leq 2^k\binom{l-1+k-d+l}{l-1} \leq 2^k\binom{k+l-1}{l-1},$$

since $l \leq d$. □

Let us turn now to the crucial sum (4.11). We distinguish three cases: $l=0$, $l=1$, and $2 \leq l \leq d$.

We begin by the particular case $l=0$, when the sum (4.11) is

$$S(n,0,d) = \sum_{k=n+d}^{\infty} 2^{-2k}\binom{k-1}{d-1}.$$

Using the trick from the proof of Lemma 3.3 in [23, pp. 27-28], we get

$$S(n,0,d) \leq 2^{-2n}2^{-2d}2\sum_{k=0}^{d-1}\binom{n+d-1}{k}. \tag{4.12}$$

Apply Lemma 4.1.2 for the last sum:

$$\begin{aligned} S(n,0,d) &\leq 2^{-2n}2^{-2d}4(d-1)!\binom{n+d-1}{d-1} = \\ & \quad 4\cdot 2^{-2d}2^{-2n}(n+1)(n+2)\cdots(n+d-1). \end{aligned}$$

In the second case, when $l=1$, Lemma 4.1.4 gives $S_k(1,d) \leq 2^k$, and hence $S(n,l,d) \leq 2^{-(n+d)}$.

Finally, for the case $2 \leq l$, we use Lemma 4.1.4 and the same trick from [23]:

$$\begin{aligned} S(n,l,d) &\leq \sum_{k=n+d}^{\infty} 2^{-k}\binom{k+l-1}{l-1} = 2^{-(n+d)}\sum_{i=0}^{\infty} 2^{-i}\binom{n+d+i+l-1}{l-1} \\ &\leq 2^{-(n+d)} 2 \sum_{j=0}^{l-1}\binom{n+d+l-1}{j}. \end{aligned}$$

Lemma 4.1.2 tells us then that the sum (4.11) can be estimated by

$$S(n,l,d) \leq 4 \cdot 2^{-(n+d)}(n+d+1)\cdots(n+d+l-1)\,. \tag{4.13}$$

As a consequence of equations (4.10), (4.11), (4.12), and (4.13), we obtain the estimation of interpolation error.

Theorem 4.1.1. *If the smoothness conditions for the differential l-form ω are met, then the upper bound for the L^2-norm of the interpolation error holds:*

$$\|\omega - \mathcal{I}_n\omega\|_0 \leq g_n^{d,l} \cdot \left(\sum_I \left\|D^I u_I\right\|_0^2\right)^{1/2}, \tag{4.14}$$

where

$$g_n^{d,l} = 3^{-(d-l/2)} \cdot \begin{cases} 2^{-2(n+d)} \cdot (n+1)\cdots(n+d-1) & \textit{for} \quad l=0 \\ 2^{-(n+d)} & \textit{for} \quad l=1 \\ 2^{-(n+d)}(n+d+1)\cdots(n+d+l-1) & \textit{for} \quad 2 \leq l \leq d \end{cases}$$

and D^I is defined in Lemma 3.2.1.

For 0-forms we obtained the same error estimation as for Lagrange finite elements (see [23] p.28). For d-forms we have a purely scalar case again and approximation based on Haar-wavelets. For the L^∞-norm, this corresponds to a known error estimate (see [64] p.25). From the result for 1-forms we obtain an estimate of the error in the $H_0^1(\Omega)$-norm (see [23] p.28) as a consequence of the commuting diagram property.

For fixed dimension, the convergence is the best for the 0-forms and slowly deteriorates with the increase of the order of the differential forms. Denoting $h_n = 2^{-n}$, we have for l forms with $l \geq 2$ convergence of order $\mathcal{O}\left(h_n|\log(h_n)|^{l-1}\right)$, whereas for the 0-forms we have convergence of order

$\mathcal{O}\left(h_n^2|\log(h_n)|^{d-1}\right)$. At a first glance, it seems that the approximation would be better for increasing dimension. This is not the case, since we demand $n \geq d$ and generally, we have in the formula for $g_n^{d,l}$ a sort of factorial.

Due to the commuting diagram property (4.6), we obtain automatically the corresponding result for the $\mathbf{H}(\mathbf{d},\Omega)$-norm.

Theorem 4.1.2. *If the smoothness conditions for the differential l-form ω are met, then the upper bound for the $\mathbf{H}(\mathbf{d},\Omega)$-norm of the interpolation error holds:*

$$\|\omega - \mathcal{I}_n\omega\|_{\mathbf{H}(d,\Omega)} \leq \left(\left(g_n^{d,l}\right)^2 + \left(g_n^{d,l+1}\right)^2\right)^{1/2} \cdot \left(\sum_I \left\|D^I u_I\right\|_0^2\right)^{1/2} . \qquad (4.15)$$

Denote the total number of the sparse grid points by $b_{d,n}$ and the number of the interior points by $a_{d,n}$. Bungartz showed in [23] that

$$\begin{aligned} a_{d,n} &= \sum_{i=0}^{n-1} 2^i \binom{d-1+i}{d-1} \\ &= (-1)^d + 2^n \sum_{i=0}^{d-1} \binom{n+d-1}{i} \cdot (-2)^{d-1-i} \\ &= 2^n \cdot \left(\frac{n^{d-1}}{(d-1)!} + \mathcal{O}(n^{d-2})\right) . \end{aligned}$$

and

$$\frac{a_{d,n}}{b_{d,n}} \longrightarrow 1\,, \text{ for } n \rightarrow \infty \text{ and fixed } d.$$

Note that the number of points in the full grid of resolution n is $(1+2^n)^d$. We refer to [23, p.25-31] for a detailed comparison between the number of points in the full and in the sparse grid. Fig. 4.3 illustrates the curse of dimensionality and the remedy proposed by the sparse grid approach.

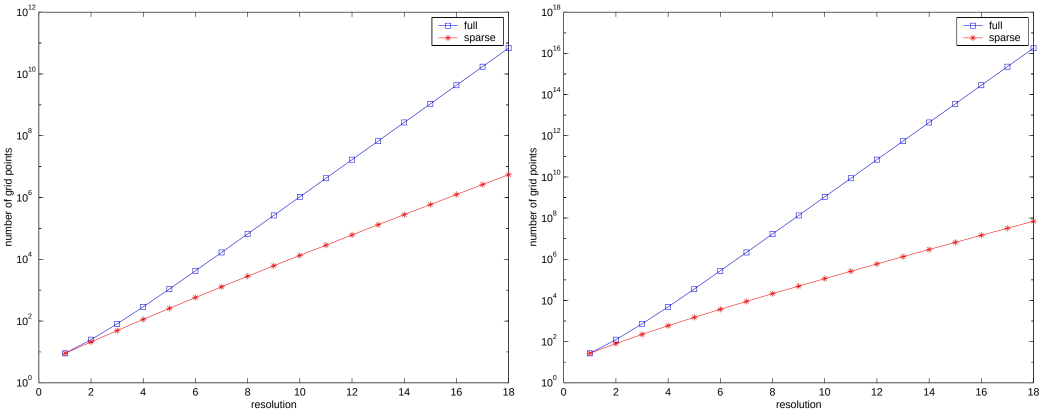


Figure 4.3: Number of full and sparse grid points for $d = 2$ and $d = 3$

4.2 Approximate Interpolation

In the course of a Galerkin approximation of a partial differential equation, we usually need to evaluate some right hand side $(\omega, \eta)_{L^2(\Omega)}$, with ω an l-form, for test forms η. For instance, consider the $\mathbf{H}(\mathbf{d}, \Omega)$-elliptic variational problem (2.4) from Sect. 2.2 for constants coefficients $\alpha = \beta = 1$: Seek l-form ξ such that

$$(\mathbf{d}\xi, \mathbf{d}\eta)_{L^2(\Omega)} + (\xi, \eta)_{L^2(\Omega)} = (\omega, \eta)_{L^2(\Omega)} \quad \forall l\text{-forms } \eta \, . \tag{4.16}$$

When performing a Galerkin-discretization based on the space of Whitney-l-forms on sparse grids, only the inner product of two discrete forms is algorithmically available. This suggests to replace the l-form ω by its interpolant $\mathcal{I}_n^{(1)}\omega$ first [21]. However, in general, for $l > 0$, it is impossible to determine the exact interpolant, because of the integrals that occur in the definition (3.16) of the degrees of freedom. This is an important new aspect that crops up when moving from 0-forms to general l-forms. We have to resign to the use of approximate interpolation.

Strang's lemma (see [77, Sect. 4.3] or [25, Sect. 4.1]) tells us what the approximate interpolation has to satisfy: in order to preserve the overall order of convergence that we get from (4.14) via Cea's lemma, the approximate interpolation has to fulfill an estimate that is at least as strong as (4.14).

Given the *smooth* differential l–form ω by its components u_I (which are smooth functions), we aim to compute an appropriate replacement for $\mathcal{I}_n^{(1)}\omega$

with minimal effort, i.e. with a number of operations proportional to the dimension of the sparse grid space. The naive application of an (adaptive) quadrature rule for the direct evaluation of the degrees of freedom (integrals over l–faces) is too expensive [33]. We introduce an approximate interpolation operator with optimal complexity in terms of evaluations. To this end, we prove the error estimates directly, based on estimates for the norms of the detail functions (hierarchical surpluses) and not for the hierarchical coefficients.

Recalling (4.2), our task can be reduced to devising suitable approximations for the hierarchical surplus operators $\mathcal{H}^I_{\boldsymbol{\nu}}$, $|\boldsymbol{\nu}|_1 \leq n+d-1$. Induction with respect to the dimension d and equation (3.30) together with (3.18) show that

$$\mathcal{H}^I_{\boldsymbol{\nu}} = \mathop{\circ}_{i\in I}(\mathcal{I}^{\text{const}}_{\nu_i} - \mathcal{I}^{\text{const}}_{\nu_i-1}) \mathop{\circ}_{j\notin I}(\mathcal{I}^{\text{lin}}_{\nu_j} - \mathcal{I}^{\text{lin}}_{\nu_j-1}) \,.$$

We replace the "exact" interpolation operators $\mathcal{I}^{\text{const}}_{\nu_i}$ with those based on the two point trapezoidal rule $\tilde{\mathcal{I}}^{\text{const}}_{\nu_i}$

$$\tilde{u}_\nu(t) = \tilde{\mathcal{I}}^{\text{const}}_{\nu} u(t) := \sum_{k=0}^{2^\nu-1} \tilde{a}_{\nu k} B_{\nu k}(t), \qquad \tilde{a}_{\nu k} = \frac{u(k2^{-\nu}) + u((k+1)2^{-\nu})}{2} \cdot 2^{-\nu}$$

and we get

$$\tilde{\mathcal{H}}^I_{\boldsymbol{\nu}} := \mathop{\circ}_{i\in I} \tilde{\mathcal{H}}_{\nu_i} \mathop{\circ}_{j\notin I} \mathcal{H}_{\nu_j} := \mathop{\circ}_{i\in I}(\tilde{\mathcal{I}}^{\text{const}}_{\nu_i} - \tilde{\mathcal{I}}^{\text{const}}_{\nu_i-1}) \mathop{\circ}_{j\notin I}(\mathcal{I}^{\text{lin}}_{\nu_j} - \mathcal{I}^{\text{lin}}_{\nu_j-1}).$$

This operator was already treated in [29, 64]. Note that for the resulting approximate interpolation operator $\tilde{\mathcal{H}}_{\boldsymbol{\nu}}$, the commuting diagram property (3.31) is no longer ensured.

We prove that the interpolation error estimates on sparse grids (4.14) hold for this approximate interpolation operator only with a different dependence on dimension d. The main tool for the derivation of the estimates in the previous two sections was Prop. 4.1.3 concerning the norms of details. An equality similar to (4.9) for the approximate interpolation operator is instrumental in getting the interpolation estimates. Taking the cue from [64, page 24], by an induction argument with respect to the dimension of the space d, we have

$$\left\|\tilde{\mathcal{H}}^I_{\boldsymbol{\nu}} u_I\right\|_{L^2(\Omega)} \;\leq\; 3^l \cdot 2^{-|\boldsymbol{\nu}_I|_1} \cdot 5^{d-l} \cdot 2^{-2|\boldsymbol{\nu}_{I'}|_1} \left\|D^I u_I\right\|_{L^2(\Omega)},$$

that is

$$\left\|\tilde{\mathcal{H}}^I_{\boldsymbol{\nu}} u_I\right\|_{L^2(\Omega)} \;\leq\; C \cdot 2^{-2|\boldsymbol{\nu}_{I'}|_1 - |\boldsymbol{\nu}_I|_1} \left\|D^I u_I\right\|_{L^2(\Omega)} .$$

Following the steps of the preceding two sections, it is easy to prove that the error estimates (4.14) hold for the approximate interpolation operator, only with a different dependence on dimension d. Indeed, instead of the term $3^{-(d-l/2)}$ in the expression of $g_n^{d,l}$, we have $5^d(3/5)^l$.

Theorem 4.2.1. *If the smoothness conditions for the differential l-form ω are met, then the upper bound for the L^2-norm of the error involving approximation holds:*

$$\left\|\omega - \tilde{\mathcal{I}}_n \omega\right\|_0 \leq \tilde{g}_n^{d,l} \cdot \left(\sum_I \left\|D^I u_I\right\|_0\right)^{1/2}, \tag{4.17}$$

where

$$\tilde{g}_n^{d,l} = 3^l \cdot 5^{d-l} \cdot \begin{cases} 2^{-2(n+d)} \cdot (n+1)\cdots(n+d-1) & \textit{for} \quad l = 0 \\ 2^{-(n+d)} & \textit{for} \quad l = 1 \\ 2^{-(n+d)}(n+d+1)\cdots(n+d+l-1) & \textit{for} \quad 2 \leq l \leq d \end{cases}$$

and D^I is defined in Lemma 3.2.1.

This means that the l-dimensional trapezoidal rule with 2^l points can be used to compute the degrees of freedom for any l-face. This is remarkable, as for large faces the quadrature error can be considerable. In sum, the number of evaluations of u_I is only a small multiple of the dimension of V_n^I. Note that, in terms of implementation, this approach forces us to make use of the combination technique [42, 64] for the evaluation of the approximate interpolation operator. We give further details in Chapter 6.2.

Figure 4.5 presents the behavior of the L^2-error for 0-, 1- and 2-forms in the two and three dimensional case, together with the theoretically estimated convergence curves. The interpolated differential form ω has all components $3\pi^2 \cdot \sin(\pi x) \cdot \sin(\pi y) \cdot \sin(\pi z)$. The reference curves show the constants $\tilde{g}_n^{d,l}$ from Theorem 4.2.1. Figure 4.6 illustrates the importance of the smoothness of the interpolated form. Here, the interpolated differential form ω has the components $u_i = x_i \cdot e^{-\left|\frac{1}{2} - ||x||\right|}$. Convergence still holds, but the order deteriorates.

We end this section by some considerations on the visualization on sparse grids, since this subject is close related to the interpolation. It is known (see also Fig. 4.3) that data sets on sparse grids at fine resolutions cannot be handled on uniform grids any more. There are visualization toolkits that work directly on sparse grids [51, 81–83]. The sparse grid data set of level $n = 12$, corresponding to the previous 0-form $\omega = x_1 \cdot e^{-\left|\frac{1}{2}-||x||\right|}$ in $d = 3$ dimensions was rendered by Mathias Hopf[1] in Fig. 4.4 with an X-ray shading method (left) and with multiple semitransparent shaded ISO-surfaces (right). The computations were done on the PC cluster "Kepler" of the University of Tübingen. This cluster consists of 96 dual PIII nodes connected with Myrinet, and two additional front-end nodes.

The computations were done on the PC cluster "Kepler" of the University of Tübingen with a visualization toolkit that works directly on sparse grids. The cluster consists of 96 dual PIII nodes connected with Myrinet, and two additional front-end nodes.

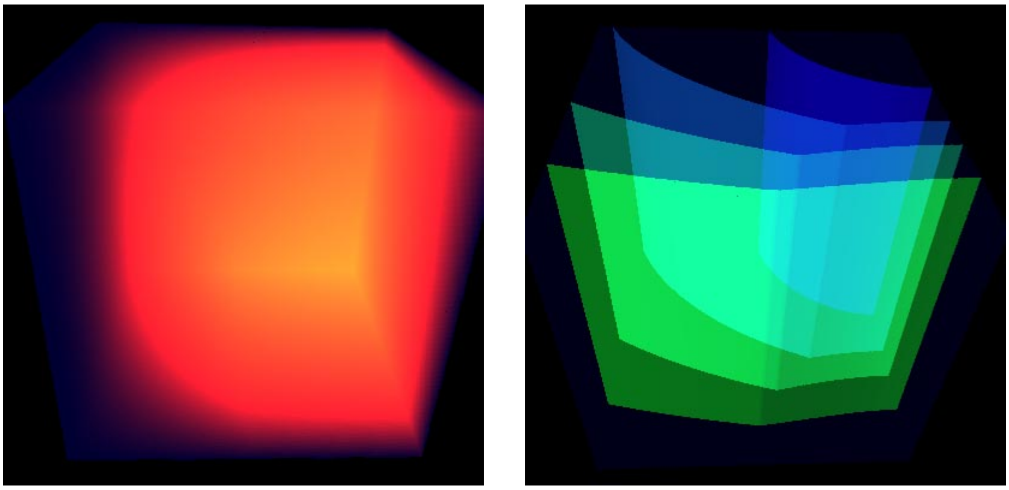


Figure 4.4: A sparse grid data set rendered with an X-ray shading method (left) and with multiple semitransparent shaded ISO-surfaces (right)

[1] Mathias Hopf is at the Visualization and Interactive Systems Group, IfI, University of Stuttgart

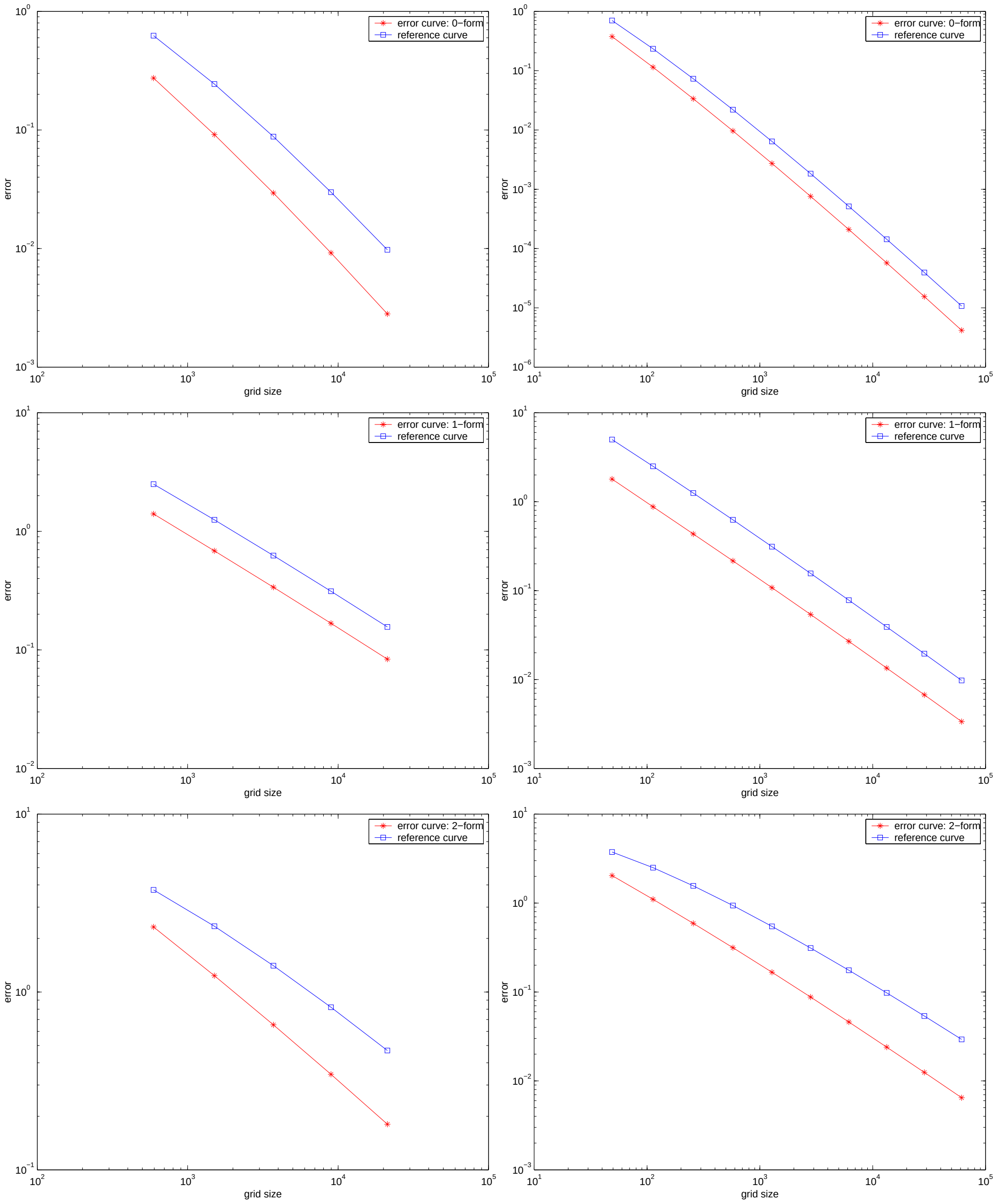


Figure 4.5: Behavior of L^2-error for 0-, 1- and 2-forms with enough smoothness: $u_i = 3\pi^2 \cdot \sin(\pi x) \cdot \sin(\pi y) \cdot \sin(\pi z)$, left for $d = 3$ and right for $d = 2$

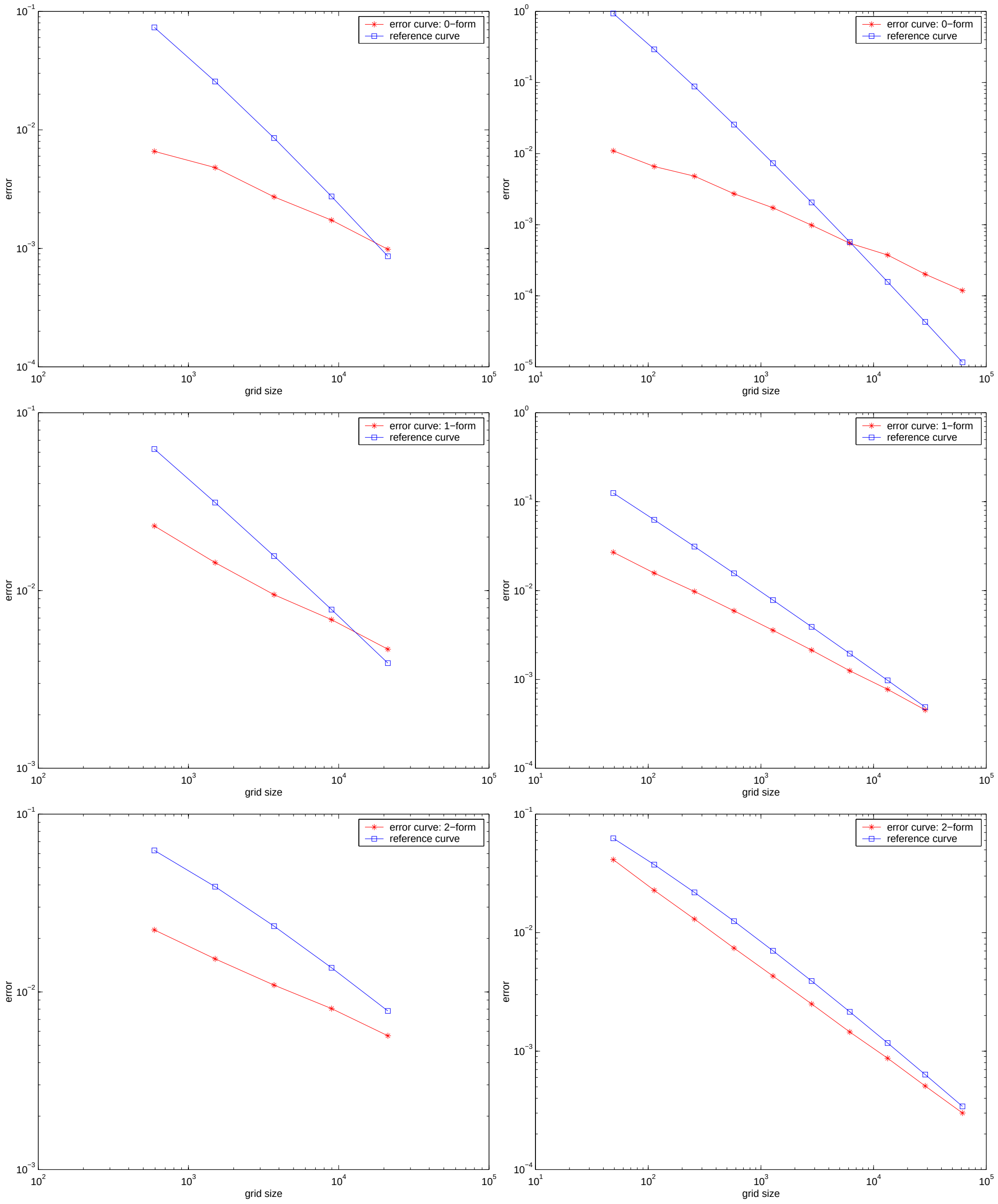


Figure 4.6: Behavior of L^2-error for 0-, 1- and 2-forms for a differential form that is not smooth enogh: $u_i = x_i \cdot e^{-\left|\frac{1}{2}-||x||\right|}$, left for $d = 3$, right for $d = 2$

Selecting particular basis functions, we constructed the sparse grid spaces. Then, we defined and investigated the properties of the sparse grid interpolation operator. We considered approximate interpolation in order to handle Galerkin schemes in partial differential equations. In the next chapter we will deal with the discretization of second order mixed problems by means of Whitney forms on sparse grids

Chapter 5

Second Order Mixed Problems

Sparse grid interpolation operators for Whitney l-forms in d dimensions and their approximation properties are at our disposal. Now, we focus on two problems from the theory of the mixed discretization of second order elliptic PDEs. We are concerned by the stability conditions of the discretization (by sparse Whitney forms) of the primal and of the dual problem below. The existence of stable potentials is a sufficient condition, which we can establish for particular cases. Rigorous results could be obtained for three dimensions.

5.1 Formulation and Stability Conditions

In light of the theory of the mixed discretization of second order elliptic PDEs we deal with the following two problems:
The primal variational problem:

$$\begin{array}{c}\text{Find an } l\text{-form } \omega \text{ and an } (l-1)\text{-form } \eta \text{ such that:}\\ \begin{array}{rcrcrl}(\mathbf{d}\omega, \mathbf{d}\zeta)_0 & + & (\mathbf{d}\eta, \zeta)_0 & = & f(\zeta) & \text{for all } l\text{-forms } \zeta\\ (\omega, \mathbf{d}\xi)_0 & & & = & 0 & \text{for all } (l-1)\text{-forms } \xi\,,\end{array}\end{array} \tag{5.1}$$

The dual variational problem:

$$\begin{array}{c}\text{Find an } l\text{-form } \omega \text{ and a closed } (l+1)\text{-form } \eta \text{ such that:}\\ \begin{array}{rcrcrl}(\omega, \zeta)_0 & + & (\eta, \mathbf{d}\zeta)_0 & = & f(\zeta) & \text{for all } l\text{-forms } \zeta\\ (\mathbf{d}\omega, \xi)_0 & & & = & g(\xi) & \text{for all closed } (l+1)\text{-forms } \xi\,.\end{array}\end{array} \tag{5.2}$$

We consider only zero Dirichlet boundary conditions in this chapter. We just write formal the right hand side in the above formulations $f(\zeta)$, $g(\xi)$ with functionals f, g, since we concentrate in the following only on the stability of the discretization of these problems. Sufficient conditions for the stability of the discretization are well known [18, 68]. For the primal problem, it is enough to prove the so-called “ellipticity on the kernel”.

Proposition 5.1.1 (Ellipticity on the kernel). *Denoting*

$$V^l_{n\perp} := \left\{ \zeta \in V^l_n, \textit{ with } (\zeta, \mathbf{d}\eta)_{L^2(\Omega)} = 0 \; \forall \eta \in V^{l-1}_n \right\} ,$$

there is a constant C depending on d and l, but not on the resolution of the sparse grid n, such that

$$\|\zeta\|_0 \leq C \|\mathbf{d}\zeta\|_0 \textit{ for all } \zeta \in V^l_{n\perp} .$$

For the dual problem, the stability is ensured by the so called LBB-condition.

Proposition 5.1.2 (LBB-condition). *There is a constant β depending on d and l, but not on the resolution of the sparse grid n, such that*

$$\sup_{\omega \in V^l_n .} \frac{(\mathbf{d}\omega, \eta)_0}{\|\omega\|_{\mathcal{H}(\mathbf{d})}} \geq \beta \|\eta\|_0 , \textit{ for all } \eta \in V^{l+1}_n \textit{closed}$$

In the next section, we prove the LBB-condition in the case $l = d - 1$. Then, we prove the ellipticity on the kernel in the particular case $d = 3$ and $l = 1$.
The crucial tool is a conjecture on the existence of stable potentials:

Conjecture 5.1.1 (Existence of stable potentials). *Given a sparse Whitney $(l+1)$-form $\omega \in V^{l+1}_n$ that is closed,i.e. $\mathbf{d}\omega = 0$, there is a sparse Whitney l-form $\eta \in V^l_n$ with $\mathbf{d}\eta = \omega$ such that*

$$\|\eta\|_0 \leq C \|\omega\|_0 ,$$

and the constant C does not depend on the discretization level n or on ω.

We show that this condition ensures the ellipticity on the kernel. Indeed, take ζ from $V^l_{n\perp}$ and consider its exterior derivative, which is a sparse Whitney $(l + 1)$-form, $\omega = \mathbf{d}\zeta \in V^{l+1}_n$. The conjecture gives then a stable potential $\eta \in V^l_n$, that is $\mathbf{d}\eta = \omega$ and $\|\eta\|_0 \leq C \|\omega\|_0$. It is enough to prove that the potential η has a larger norm than the original $\|\zeta\|_{L^2(\Omega)}$, that is $\|\eta\|_{L^2(\Omega)} \geq \|\zeta\|_{L^2(\Omega)}$. But $\mathbf{d}\eta = \omega = \mathbf{d}\zeta$, and the exact sequence property tells us that the difference $\zeta-\eta$ is the derivative of a sparse Whitney $(l-1)$-form: $\zeta-\eta=\mathbf{d}f$, with $f\in V_n^{l-1}$. Now, we can write

$$\|\zeta\|^2_{L^2(\Omega)} \;=\; (\zeta,\zeta)_{L^2(\Omega)}=(\zeta,\eta)_{L^2(\Omega)}+(\zeta,\mathbf{d}f)_{L^2(\Omega)}\;.$$

The last term in sum is zero, because ζ belongs to $V^l_{n\perp}$. Hence,

$$\|\zeta\|^2_{L^2(\Omega)} \;=\; (\zeta,\eta)_{L^2(\Omega)}\le\|\zeta\|_{L^2(\Omega)}\,\|\zeta\|_{L^2(\Omega)}\;,$$

which ends the proof.

The existence of the stable potentials in the space of Whitney forms on sparse grids in the general case is still open. In the next section, we establish it for $l=d-1$, see Theorem 5.2.1. In the proof of this theorem, we illustrate how the existence of stable potentials implies the LBB-condition, too. In Sect. 5.3, we prove Proposition 5.1.1 in the three dimensional case for $l=1$. In the last section of this chapter, we show the existence of the stable potentials for $l=0$ in general d dimensions.

5.2 LBB-condition for $d-1$ forms

We concentrate on the proof of the discrete LBB condition in the case $l=d-1$, for general dimension d. All d-forms in d dimensions are closed, and in order to emphasize that the d-form is in fact a function, we change the notation, writing f_n instead of η in the LBB formula:

$$\sup_{\omega_n\in V_n^{d-1}}\;\frac{(\mathbf{d}\omega_n,f_n)_{L^2(\Omega)}}{\|\omega_n\|_{\mathcal{H}(\mathbf{d})}}\ge\beta\,\|f_n\|_{L^2(\Omega)}\;\text{ for all discrete } d\text{-forms } f_n\in V_n^d. \tag{5.3}$$

Here, V_n^d is used as notation for the space of the Whitney d-forms living on the sparse grid of resolution n, according to definition 4.1.3. If $d=3$, then $\mathbf{d}$ stands for the divergence operator in (5.3). We prove the existence of the discrete LBB-constant β depending neither on n, nor on f_n in an entirely discrete setting. Conversely, the usual technique [18, Sect. II.2], [69, Sect. 10] relies on lifting properties of the continuous Laplacian along with interpolation estimates.

Our approach is much simpler, because we have at our disposal two special tools: the existence of discrete potentials for Whitney forms and the orthogonality of the Haar-wavelets. The first means that given f_n in V_n^d, there exists ω_n in V_n^{d-1} such that

$$\mathbf{d}\omega_n = f_n \ . \tag{5.4}$$

The second tool can be employed to show the stability of the potential

$$\|\omega_n\|_{L^2(\Omega)} \leq (3d)^{-1} \|f_n\|_{L^2(\Omega)} \ . \tag{5.5}$$

Then the left hand side of (5.3) is greater than

$$\frac{\|f_n\|^2_{L^2(\Omega)}}{\sqrt{\|\omega_n\|^2_{L^2(\Omega)} + \|f_n\|^2_{L^2(\Omega)}}} \geq \frac{1}{\sqrt{1+(3d)^{-1}}} \|f_n\|_{L^2(\Omega)} \,,$$

so there exists an LBB-constant $\beta \geq \sqrt{\frac{3d}{3d+1}}$. It remains to prove the following theorem:

Theorem 5.2.1. *For each f_n from the space of Whitney d-forms on the sparse grid of resolution n, there exists a corresponding Whitney $(d-1)$-form ω_n such that* (5.4) *and* (5.5) *hold.*

Proof. Consider a sparse Whitney d-form f_n from V_n^d, that is, cf. (4.2) and (3.28),

$$f_n = \sum_{|\boldsymbol{\nu}|_1 \leq n+d-1} \sum_{\mathbf{q}\in\Theta_{\boldsymbol{\nu}}} \beta_{\boldsymbol{\nu}\mathbf{q}} \Psi_{\boldsymbol{\nu}\mathbf{q}}^{\{1,\dots,d\}}.$$

Thanks to the orthonormality of the Haar-wavelets $\Psi_{\boldsymbol{\nu}\mathbf{q}}^{\{1,\dots,d\}}$, we have

$$\|f_n\|^2_{L^2(\Omega)} = \sum_{|\boldsymbol{\nu}|_1 \leq n+d-1} \sum_{\mathbf{q}\in\Theta_{\boldsymbol{\nu}}} |\beta_{\boldsymbol{\nu}\mathbf{q}}|^2. \tag{5.6}$$

Now, we define the sparse Whitney $(d-1)$-form $\omega_n := \sum_{i=1}^{d} v_n^i dx_{i'}$, with $i' = \{1,\dots,d\} \setminus \{i\}$, and the components of the vector proxy

$$v_n^i(\mathbf{x}) := \frac{1}{d} \sum_{|\boldsymbol{\nu}|_1 \leq n+d-1} \sum_{\mathbf{q}\in\Theta_{\boldsymbol{\nu}}} \beta_{\boldsymbol{\nu}\mathbf{q}} 2^{-(\nu_i+1)/2} \varphi_{\nu_i q_i}(x_i) \prod_{j\neq i} \psi_{\nu_j q_j}(x_j). \tag{5.7}$$

As the derivative of the hat function is a Haar wavelet, equation (3.10) ensures that ω_n is a potential for f_n, that is $\mathbf{d}\omega_n = f_n$. In the following, we prove the inequality (5.5).

Denoting $\boldsymbol{\nu}' = \boldsymbol{\nu} \setminus \nu_i$ and $\mathbf{q}' = \mathbf{q} \setminus q_i$, we can rewrite the i-th component of the vector proxy (for a fixed i):

$$v_n^i(\mathbf{x}) = \frac{1}{d} \sum_{|\boldsymbol{\nu}'|_1 \leq n+d-1} \sum_{\mathbf{q}' \in \Theta_{\boldsymbol{\nu}'}} v_{\boldsymbol{\nu}'\mathbf{q}'}(x_i) \prod_{j \neq i} \psi_{\nu_j q_j}(x_j),$$

where

$$v_{\boldsymbol{\nu}'\mathbf{q}'}(x_i) := \sum_{\nu_i=0}^{n+d-1-|\boldsymbol{\nu}'|} \sum_{q_i \in \theta_{\nu_i}} \beta_{\boldsymbol{\nu}\mathbf{q}} 2^{-(\nu_i+1)/2} \varphi_{\nu_i q_i}(x_i).$$

Using again the orthonormality of the Haar-wavelets $\Psi_{\boldsymbol{\nu}'\mathbf{q}'}^{\{1,\ldots,d\}\setminus\{i\}}$, we obtain

$$\left\|v_n^i\right\|_{L^2(\Omega)}^2 = \frac{1}{d^2} \int_0^1 \sum_{\substack{|\boldsymbol{\nu}'|_1 \leq n+d-1 \\ \mathbf{q}' \in \Theta_{\boldsymbol{\nu}'}}} |v_{\boldsymbol{\nu}'\mathbf{q}'}(x_i)|^2 dx_i = \frac{1}{d^2} \sum_{\substack{|\boldsymbol{\nu}'|_1 \leq n+d-1 \\ \mathbf{q}' \in \Theta_{\boldsymbol{\nu}'}}} \left\|v_{\boldsymbol{\nu}'\mathbf{q}'}\right\|_{L^2(0,1)}^2.$$

Now, we consider $\left\|v_{\boldsymbol{\nu}'\mathbf{q}'}\right\|_{L^2(0,1)}^2$ and use the Cauchy-Schwarz inequality for the sum:

$$\begin{aligned} \left\|v_{\boldsymbol{\nu}'\mathbf{q}'}\right\|_{L^2(0,1)}^2 &= \int_0^1 \left[\sum_{\substack{\nu_i=0 \\ q_i \in \theta_{\nu_i}}}^{n+d-1-|\boldsymbol{\nu}'|_1} \beta_{\boldsymbol{\nu}\mathbf{q}} 2^{-(\nu_i+1)/2} \varphi_{\nu_i q_i}(x_i) \right]^2 dx_i \leq \\ &\leq \left(\sum_{\substack{\nu_i=0 \\ q_i \in \theta_{\nu_i}}}^{n+d-1-|\boldsymbol{\nu}'|_1} |\beta_{\boldsymbol{\nu}\mathbf{q}}|^2 \right) \left(\sum_{\substack{\nu_i=0 \\ q_i \in \theta_{\nu_i}}}^{n+d-1-|\boldsymbol{\nu}'|_1} 2^{-(\nu_i+1)} \left\|\varphi_{\nu_i q_i}\right\|_{L^2(0,1)}^2 \right). \end{aligned}$$

Replace the norm of the hat by its concrete value (3.4) and obtain

$$\begin{aligned} \left\|v_{\boldsymbol{\nu}'\mathbf{q}'}\right\|_{L^2(0,1)}^2 &\leq \left(\sum_{\substack{\nu_i=0 \\ q_i \in \theta_{\nu_i}}}^{n+d-1-|\boldsymbol{\nu}'|_1} |\beta_{\boldsymbol{\nu}\mathbf{q}}|^2 \right) \left(\tfrac{1}{3} \sum_{\nu_i=0}^{n+d-1-|\boldsymbol{\nu}'|_1} \tfrac{1}{2} 2^{-\nu_i} \right) \\ &\leq \tfrac{1}{3} \sum_{\substack{\nu_i=0 \\ q_i \in \theta_{\nu_i}}}^{n+d-1-|\boldsymbol{\nu}'|_1} |\beta_{\boldsymbol{\nu}\mathbf{q}}|^2. \end{aligned}$$

Together with equation (5.6), this gives for the components of the vector proxy

$$\left\|v_n^i\right\|_{L^2(\Omega)}^2 \le \frac{1}{3d^2} \sum_{\substack{|\boldsymbol{\nu}'|_1 \le n+d-1 \\ \mathbf{q}' \in \Theta_{\boldsymbol{\nu}'}}} |\beta_{\boldsymbol{\nu}\mathbf{q}}|^2 \le \frac{1}{3d^2} \|f_n\|_{L^2(\Omega)}^2 ,$$

so the norm of the sparse potential is

$$\|\omega_n\|_{L^2(\Omega)}^2 = \sum_{i=1}^{d} \left\|v_n^i\right\|_{L^2(\Omega)}^2 \le \frac{1}{3d} \|f_n\|_{L^2(\Omega)}^2 ,$$

from which we conclude that the estimate (5.5) holds. □

By Lagrange multipliers technique, the LBB-constant β can be numerically computed as $\sqrt{\lambda}$, where λ is the smallest solution of the generalized eigenvalue problem $BA^{-1}B^t x = \lambda M x$. The matrices A and M are the discrete counterparts of the scalar products in $\boldsymbol{H}(\mathrm{div};\Omega)$ and $L^2(\Omega)$, respectively, and $B = MD$, where D represents the exterior derivative operator. We solved numerically this eigenvalue problem by the power method for $\mu u = (BA^{-1}B^t)^{-1} M u$, with $\mu = 1/\lambda$. The termination criterion was $|\mu_{\text{old}} - \mu_{\text{new}}|/|\mu_{\text{new}}| \le 10^{-4}$. For the solution of the inner system $BA^{-1}B^t x = y$, we used the Uzawa method, where a few cg-steps were enough for approximately inverting A.

We present in table 5.1 the results for the two, three and four dimensional case together with the lower bound $\beta_d = \sqrt{\frac{3d}{1+3d}}$. The numerical results confirm that the LBB-constant β stay upon the obtained lower bound β_d.

n	5	6	7	8	9	10	11	12	β_d
$d=2$	0.978	0.978	0.978	0.978	0.978	0.978	0.978	0.978	0.926
$d=3$	0.989	0.987	0.987	0.987	0.987	0.987	0.987		0.946
$d=4$	0.997	0.995	0.995						0.961

Table 5.1: Approximation of the LBB constant β

5.3 Ellipticity on Kernel in three Dimensions

Now, we restrict ourselves to the case $d = 3$ and $l = 1$. In light of the last observation in Sect. 5.1, it is enough to prove the existence of stable potentials.

Theorem 5.3.1. *For $d = 3$, given the sparse Whitney 2-form $\omega \in V_n^2$ with $\mathbf{d}\omega = 0$, there is a sparse Whitney 1-form $\eta \in V_n^1$ with $\mathbf{d}\eta = \omega$ such that*

$$\|\eta\|_0 \leq \frac{1}{\sqrt{6}} \|\omega\|_0$$

Proof. Let us begin by some notations. Consider the closed Whitney 2-form

$$\omega = v^{1,2} dx_1 \wedge dx_2 + v^{2,3} dx_2 \wedge dx_3 + v^{3,1} dx_3 \wedge dx_1$$

with the components of the vector proxy

$$\begin{aligned}
v^{1,2} &= \sum_{\substack{|\boldsymbol{\nu}|_1 \leq n+d-1 \\ \mathbf{q} \in \Theta_{\boldsymbol{\nu}}}} \beta^{1,2}_{\boldsymbol{\nu}\mathbf{q}} \Psi^{\{1,2\}}_{\boldsymbol{\nu}\mathbf{q}}\,, \\
v^{2,3} &= \sum_{\substack{|\boldsymbol{\nu}|_1 \leq n+d-1 \\ \mathbf{q} \in \Theta_{\boldsymbol{\nu}}}} \beta^{2,3}_{\boldsymbol{\nu}\mathbf{q}} \Psi^{\{2,3\}}_{\boldsymbol{\nu}\mathbf{q}}\,, \\
v^{3,1} &= \sum_{\substack{|\boldsymbol{\nu}|_1 \leq n+d-1 \\ \mathbf{q} \in \Theta_{\boldsymbol{\nu}}}} \beta^{3,1}_{\boldsymbol{\nu}\mathbf{q}} \Psi^{\{3,1\}}_{\boldsymbol{\nu}\mathbf{q}}\,.
\end{aligned}$$

In order to simplify the notation, we drop the upperscripts in the lengthy formulas. Denote

$$\begin{aligned}
\beta^{1,2}_{\boldsymbol{\nu}\mathbf{q}} &= a_{\boldsymbol{\nu}\mathbf{q}}\,, & 2^{(1+\nu_3)/2} &= \gamma_{\nu_3}\,, \\
\beta^{2,3}_{\boldsymbol{\nu}\mathbf{q}} &= b_{\boldsymbol{\nu}\mathbf{q}}\,, & 2^{(1+\nu_1)/2} &= \alpha_{\nu_1}\,, \\
\beta^{3,1}_{\boldsymbol{\nu}\mathbf{q}} &= c_{\boldsymbol{\nu}\mathbf{q}}\,, & 2^{(1+\nu_2)/2} &= \beta_{\nu_2}\,.
\end{aligned}$$

Note that a, b, c varies with $\boldsymbol{\nu}$ and $\mathbf{q}$, but α, β, γ depend only on ν_1, ν_2, ν_3, respectively. The condition that ω is closed, $\mathbf{d}\omega = 0$, gives then

$$\gamma_{\nu_3} a_{\boldsymbol{\nu}\mathbf{q}} + \alpha_{\nu_1} b_{\boldsymbol{\nu}\mathbf{q}} + \beta_{\nu_2} c_{\boldsymbol{\nu}\mathbf{q}} = 0. \tag{5.8}$$

for all admissible $\boldsymbol{\nu}$, $\mathbf{q}$.

We have to construct the potential of ω, that is a Whitney 1–form

$$\eta = u^1 dx_1 + u^2 dx_2 + u^3 dx_3\,.$$

Consider the components of the vector proxy corresponding to the potential η

$$u^i = \sum_{\substack{|\boldsymbol{\nu}|_1 \leq n+d-1 \\ \mathbf{q}\in\Theta_{\boldsymbol{\nu}}}} \beta^i_{\boldsymbol{\nu}\mathbf{q}} \Psi^{\{i\}}_{\boldsymbol{\nu}\mathbf{q}}\,,$$

and denote (remembering the preceding convention)

$$\beta^1_{\boldsymbol{\nu}\mathbf{q}} = t_{\boldsymbol{\nu}\mathbf{q}}\,, \quad \beta^2_{\boldsymbol{\nu}\mathbf{q}} = y_{\boldsymbol{\nu}\mathbf{q}}\,, \quad \beta^3_{\boldsymbol{\nu}\mathbf{q}} = z_{\boldsymbol{\nu}\mathbf{q}}\,.$$

The potential condition $\mathbf{d}\eta = \omega$ shows that, at least for all $\nu_1, \nu_2, \nu_3 \geq 1$ and all $\mathbf{q}$, we have to demand

$$\begin{aligned} \alpha_{\nu_1} y_{\boldsymbol{\nu}\mathbf{q}} - \beta_{\nu_2} t_{\boldsymbol{\nu}\mathbf{q}} &= a_{\boldsymbol{\nu}\mathbf{q}} \\ \beta_{\nu_2} z_{\boldsymbol{\nu}\mathbf{q}} - \gamma_{\nu_3} y_{\boldsymbol{\nu}\mathbf{q}} &= b_{\boldsymbol{\nu}\mathbf{q}} \\ \gamma_{\nu_3} t_{\boldsymbol{\nu}\mathbf{q}} - \alpha_{\nu_1} z_{\boldsymbol{\nu}\mathbf{q}} &= c_{\boldsymbol{\nu}\mathbf{q}}\,. \end{aligned}$$

Together with equation (5.8), this tells us that we have a parameter at our disposal. Let us set $t_{\boldsymbol{\nu}\mathbf{q}} = 0$ for all $\boldsymbol{\nu}$ and $\mathbf{q}$. Remark that imposing Dirichlet zero boundary conditions eliminates a lot of sum terms corresponding to the boundary.

With the last notations we have:

$$\begin{aligned} u^1 &= 0\,, & v^{2,3} &= \sum_{\boldsymbol{\nu}\mathbf{q}} (\beta_{\nu_2} z_{\boldsymbol{\nu}\mathbf{q}} - \gamma_{\nu_3} y_{\boldsymbol{\nu}\mathbf{q}}) \Psi^{\{2,3\}}_{\boldsymbol{\nu}\mathbf{q}}\,, \\ u^2 &= \sum_{\boldsymbol{\nu}\mathbf{q}} y_{\boldsymbol{\nu}\mathbf{q}} \Psi^{\{2\}}_{\boldsymbol{\nu}\mathbf{q}}\,, & v^{1,2} &= \sum_{\boldsymbol{\nu}\mathbf{q}} \alpha_{\nu_1} y_{\boldsymbol{\nu}\mathbf{q}} \Psi^{\{1,2\}}_{\boldsymbol{\nu}\mathbf{q}}\,, \\ u^3 &= \sum_{\boldsymbol{\nu}\mathbf{q}} z_{\boldsymbol{\nu}\mathbf{q}} \Psi^{\{3\}}_{\boldsymbol{\nu}\mathbf{q}}\,, & v^{3,1} &= \sum_{\boldsymbol{\nu}\mathbf{q}} (-\alpha_{\nu_1}) z_{\boldsymbol{\nu}\mathbf{q}} \Psi^{\{3,1\}}_{\boldsymbol{\nu}\mathbf{q}}\,. \end{aligned}$$

If we prove that $\|u^2\|^2_{L^2(\Omega)} \leq C\,\|v^{1,2}\|^2_{L^2(\Omega)}$ and $\|u^3\|^2_{L^2(\Omega)} \leq C\,\|v^{3,1}\|^2_{L^2(\Omega)}$, then the theorem holds. Consider the second component of the potential

$$u^2(\mathbf{x}) = \sum_{\substack{\nu_2=1, \\ q_2\in\Theta_{\nu_2}}}^{n+d-1} \left(\sum_{\substack{\nu_1+\nu_3\leq n+d-1-\nu_2 \\ q_1,\, q_2}} y_{\boldsymbol{\nu}\mathbf{q}} \varphi_{\nu_1 q_1}(x_1) \varphi_{\nu_3 q_3}(x_3) \right) \psi_{\nu_2 q_2}(x_2)\,.$$

The orthogonality of $\psi_{\nu_2 q_2}$ gives

$$
\begin{aligned}
\left\|u^2\right\|^2_{L^2(\Omega)} &= \int\limits_0^1 \int\limits_0^1 \sum_{\substack{\nu_2=1,\\ q_2\in\Theta_{\nu_2}}}^{n+d-1} \left(\sum_{\substack{\nu_1+\nu_3\leq n+d-1-\nu_2\\ q_1,\, q_2}} y_{\boldsymbol{\nu}\mathbf{q}} \varphi_{\nu_1 q_1}(x_1) \varphi_{\nu_3 q_3}(x_3) \right)^2 dx_1 dx_3 \\
&= \sum_{\substack{\nu_2=1,\\ q_2\in\Theta_{\nu_2}}}^{n+d-1} \int\limits_0^1 \int\limits_0^1 \left(\sum_{\substack{\nu_1+\nu_3\leq n+d-1-\nu_2\\ q_1,\, q_2}} y_{\boldsymbol{\nu}\mathbf{q}} \varphi_{\nu_1 q_1}(x_1) \varphi_{\nu_3 q_3}(x_3) \right)^2 dx_1 dx_3 \,.
\end{aligned}
$$

Consider the integrand and use the Cauchy-Schwarz inequality for sums:

$$
\begin{aligned}
&\left(\sum_{\substack{\nu_1+\nu_3\leq n+d-1-\nu_2\\ q_1,\, q_2}} y_{\boldsymbol{\nu}\mathbf{q}} \varphi_{\nu_1 q_1} \varphi_{\nu_3 q_3} \right)^2 \\
&\qquad = \left[\sum_{\substack{\nu_1=1\\ q_1\in\Theta_{\nu_1}}}^{n+d-1-\nu_2} \left(\alpha_{\nu_1} \sum_{\substack{\nu_3=0\\ q_3\in\Theta_{\nu_3}}}^{n+d-1-\nu_2-\nu_1} y_{\boldsymbol{\nu}\mathbf{q}} \varphi_{\nu_3 q_3} \right) \frac{1}{\alpha_{\nu_1}} \varphi_{\nu_1 q_1} \right]^2 \leq \\
&\qquad \leq \sum_{\substack{\nu_1=1\\ q_1\in\Theta_{\nu_1}}}^{n+d-1-\nu_2} \alpha^2_{\nu_1} \left(\sum_{\substack{\nu_3=0\\ q_3\in\Theta_{\nu_3}}}^{n+d-1-\nu_2-\nu_1} y_{\boldsymbol{\nu}\mathbf{q}} \varphi_{\nu_3 q_3} \right)^2 \left(\sum_{\substack{\nu_1=1\\ q_1\in\Theta_{\nu_1}}}^{n+d-1-\nu_2} \frac{1}{\alpha^2_{\nu_1}} \varphi^2_{\nu_1 q_1} \right) .
\end{aligned}
$$

Hence, the squared norm of the second component of the potential is

$$
\begin{aligned}
\left\|u^2\right\|^2_{L^2(\Omega)} \leq \sum_{\substack{\nu_2=1,\\ q_2\in\Theta_{\nu_2}}}^{n+d-1} & \left(\sum_{\substack{\nu_1=1\\ q_1\in\Theta_{\nu_1}}}^{n+d-1-\nu_2} \alpha^2_{\nu_1} \int\limits_0^1 \left(\sum_{\substack{\nu_3=0\\ q_3\in\Theta_{\nu_3}}}^{n+d-1-\nu_2-\nu_1} y_{\boldsymbol{\nu}\mathbf{q}} \varphi_{\nu_3 q_3}(x_3) \right)^2 dx_3 \right. \\
& \left. \cdot \sum_{\substack{\nu_1=1\\ q_1\in\Theta_{\nu_1}}}^{n+d-1-\nu_2} \frac{1}{\alpha^2_{\nu_1}} \left\|\varphi_{\nu_1 q_1}\right\|^2_{L^2(0,1)} \right) .
\end{aligned}
$$

But we can compute the last sum

$$\begin{aligned}\sum_{\substack{\nu_1=1\\ q_1\in\Theta_{\nu_1}}}^{n+d-1-\nu_2} \frac{1}{\alpha_{\nu_1}^2}\left\|\varphi_{\nu_1 q_1}\right\|_{L^2(0,1)}^2 &= \sum_{\nu_1=1}^{n+d-1-\nu_2} \frac{1}{\alpha_{\nu_1}^2}\frac{2}{3}2^{-\nu_1}2^{\nu_1-1} = \frac{1}{3}\sum_{\nu_1=1}^{n+d-1-\nu_2} 2^{-(\nu_1+1)}\\ &\leq \frac{1}{6}, \qquad (5.9)\end{aligned}$$

and consequently

$$\left\|u^2\right\|_{L^2(\Omega)}^2 \leq \frac{1}{6}\sum_{\substack{\nu_1+\nu_2\leq n+d-1\\ q_1\in\Theta_{\nu_1},q_2\in\Theta_{\nu_2}}} \alpha_{\nu_1}^2\int_0^1\left(\sum_{\substack{\nu_3=0\\ q_3\in\Theta_{\nu_3}}}^{n+d-1-\nu_2-\nu_1} y_{\boldsymbol{\nu}\mathbf{q}}\varphi_{\nu_3 q_3}(x_3)\right)^2 dx_3\,.$$

Taking into account again the orthogonality of Haar wavelets $\psi_{\nu_1 q1}\psi_{\nu_2 q_2}$ for the expression of $v^{1,2}$, we get exactly the previous sum:

$$\begin{aligned}\left\|v^{1,2}\right\|_{L^2(\Omega)}^2 &= \\ \int_\Omega\left[\sum_{\substack{\nu_1+\nu_2\leq n+d-1\\ q_1\in\Theta_{\nu_1},q_2\in\Theta_{\nu_2}}}\left(\alpha_{\nu_1}\sum_{\substack{\nu_3=0\\ q_3\in\Theta_{\nu_3}}}^{n+d-1-\nu_2-\nu_1} y_{\boldsymbol{\nu}\mathbf{q}}\varphi_{\nu_3 q_3}(x_3)\right)\psi_{\nu_1 q_1}(x_1)\psi_{\nu_2 q_2}(x_2)\right]^2 d\mathbf{x}\\ &= \int_0^1\sum_{\substack{\nu_1+\nu_2\leq n+d-1\\ q_1\in\Theta_{\nu_1},q_2\in\Theta_{\nu_2}}}\alpha_{\nu_1}^2\left(\sum_{\substack{\nu_3=0\\ q_3\in\Theta_{\nu_3}}}^{n+d-1-\nu_2-\nu_1} y_{\boldsymbol{\nu}\mathbf{q}}\varphi_{\nu_3 q_3}(x_3)\right)^2 dx_3\,.\end{aligned}$$

Hence,

$$\left\|u^2\right\|_{L^2(\Omega)}^2 \leq \frac{1}{6}\left\|v^{1,2}\right\|_{L^2(\Omega)}^2.$$

Analogous manipulations provide the inequality $\left\|u^3\right\|_{L^2(\Omega)}^2 \leq \frac{1}{6}\left\|v^{3,1}\right\|_{L^2(\Omega)}^2$. This amounts to the contention of the theorem

$$\|\eta\|_0 \leq C\,\|\omega\|_0\;,$$

with the stability constant C satisfying

$$C \leq \frac{1}{\sqrt{6}} = 0.4082\,.$$

Remark that C could be even smaller, because of the remaining term $\|v^{2,3}\|^2_{L^2(\Omega)}$.
□

Now, we aim at a numerical evaluation of C. Denote $A = 1/C^2$. The Lagrange multipliers technique for the problem

$$\inf\Big\{(\mathbf{d}\eta,\mathbf{d}\eta)_{L^2(\Omega)}\,;\quad \eta\in V^1_{n\perp},\ (\eta,\eta)_{L^2(\Omega)}=1\Big\}=A$$

instantly gives

$$\left\{\begin{array}{lllllll} \frac{1}{2}\,(\mathbf{d}\eta,\mathbf{d}\zeta)_{L^2(\Omega)} & + & (\zeta,\mathbf{d}f)_{L^2(\Omega)} & - & \lambda\,(\eta,\zeta)_{L^2(\Omega)} & = & 0 \qquad \text{for all } \zeta\in V^1_n \\ \quad(\eta,\mathbf{d}g)_{L^2(\Omega)} & & & & & = & 0 \qquad \text{for all } g\in V^0_n. \end{array}\right.$$

Set $\zeta=\mathbf{d}f$ in the first equation, then $\mathbf{d}f=0$. Hence, we have to approximate the first non-zero eigenvalue of the generalized eigenvalue problem for the **curl** operator with zero Dirichlet boundary conditions:

$$(\mathbf{d}\eta,\mathbf{d}\eta)_{L^2(\Omega)}=\lambda\,(\eta,\zeta)_{L^2(\Omega)}\,,\quad \eta,\zeta\in V^1_n.$$

This was done using the PPINVIT–algorithm of Hiptmair and Neymeyr [50] with a CG-solver obtaining

$$A\simeq 19.7393 \qquad\text{and}\qquad C\simeq 0.2251.$$

5.4 Existence of stable potentials

In the previous sections we proved Prop. 5.1.1 for the particular cases $l = d-1$ or $d=3$ and $l=1$. We focus now on the case $l=0$ in d dimensions and we finish this chapter with a discussion on possibilities to complete the proof for general l, d.

For the sake of simplicity, we consider again Dirichlet zero boundary conditions. The next theorem is the Friedrich's inequality. New is the dependence on the dimension, known that the constant in the traditional Friedrich's inequality $C\approx \mathrm{diam}(\Omega)$.

Theorem 5.4.1. *Given the sparse Whitney 1-form $\omega\in V^1_n$ with $\mathbf{d}\omega=0$, there is a sparse Whitney 0-form $\eta\in V^0_n$ with $\mathbf{d}\eta=\omega$ such that*

$$\|\eta\|_0\le\frac{1}{\sqrt{6d}}\,\|\omega\|_0$$

Proof. Consider a Whitney 1-form $\omega \in V_n^1$, that is, cf. (4.2) and (3.28)

$$\omega = \sum_{i=1}^{d} w^i(\mathbf{x}) dx_i \,,$$

with the components of the vector proxy

$$w^i(\mathbf{x}) = \sum_{|\boldsymbol{\nu}|_1 \le n+d-1} \sum_{\mathbf{q} \in \Theta_{\boldsymbol{\nu}}} \beta_{\boldsymbol{\nu}\mathbf{q}} \Psi^i_{\boldsymbol{\nu}\mathbf{q}}(\mathbf{x}) \,.$$

Denote $\lambda_i = (\nu_i \ q_i)$ and $\boldsymbol{\lambda} = (\boldsymbol{\nu} \ \mathbf{q})$. In order to simplify the notations, we omit the summation limits in the following. Hence, the components are

$$w^i(\mathbf{x}) = \sum_{\boldsymbol{\lambda}} \beta^i_{\boldsymbol{\lambda}} \psi_{\lambda_i}(x_i) \prod_{j \neq i} \varphi_{\lambda_j}(x_j) \,.$$

Prop. 4.1.2 ensures the existence of the discrete potential $\eta \in V_n^0$, $\eta = u(\mathbf{x})$, such that $\mathbf{d}\eta = \omega$. The potential

$$u(\mathbf{x}) = \sum_{\boldsymbol{\lambda}} \gamma_{\boldsymbol{\lambda}} \prod_{j=1}^{d} \varphi_{\lambda_j}(x_j) \,,$$

has the gradient

$$\begin{aligned} \mathbf{d}\eta &= \sum_{i=1}^{d} \sum_{\boldsymbol{\lambda}} \gamma_{\boldsymbol{\lambda}} \varphi'_{\lambda_i}(x_i) \prod_{j \neq i} \varphi_{\lambda_j}(x_j) dx_i \\ &= \sum_{i=1}^{d} \sum_{\boldsymbol{\lambda}} \gamma_{\boldsymbol{\lambda}} 2^{(\nu_i+1)/2} \psi_{\lambda_i}(x_i) \prod_{j \neq i} \varphi_{\lambda_j}(x_j) x_i \,, \end{aligned}$$

as the derivative of the hat is a Haar-wavelet (3.10). Denoting $a_{\lambda_i} = a_{\nu_i} = 2^{(\nu_i+1)/2}$, the equality $\mathbf{d}\eta = \omega$ reduces to

$$a_{\lambda_i} \cdot \gamma_{\boldsymbol{\lambda}} = \beta^i_{\boldsymbol{\lambda}} \text{ for all } i \text{ and } \boldsymbol{\lambda} \,. \tag{5.10}$$

Remark that the norm of the form is determined by the norm of the components

$$\|\omega\|^2_{L^2(\Omega)} = \sum_{i=1}^{d} \left\| w^i \right\|^2_{L^2(\Omega)} \,,$$

and rewrite the components of the vector proxy as

$$w^i(\mathbf{x}) = \sum_{\lambda_i} \left(\sum_{\boldsymbol{\lambda}'} \beta^i_{\boldsymbol{\lambda}} \prod_{j \neq i} \varphi_{\lambda_j}(x_j) \right) \psi_{\lambda_i}(x_i)\,,$$

where $\boldsymbol{\lambda}' = \boldsymbol{\lambda} \setminus \lambda_i$ for fixed i. The orthogonality of the Haar-wavelets gives

$$\left\| w^i \right\|^2_{L^2(\Omega)} = \sum_{\lambda_i} \int_0^1 \cdots \int_0^1 \left(\sum_{\boldsymbol{\lambda}'} \beta^i_{\boldsymbol{\lambda}} \prod_{j \neq i} \varphi_{\lambda_j}(x_j) \right)^2 \prod_{j \neq i} dx_j\,. \tag{5.11}$$

Consider now the squared norm of the potential

$$\|\eta\|^2_{L^2(\Omega)} = \int_\Omega \left(\sum_{\boldsymbol{\lambda}} \gamma_{\boldsymbol{\lambda}} \prod_{j=1}^{d} \varphi_{\lambda_j}(x_j) \right)^2 .$$

Fix $i \in \{1, 2, \ldots, d\}$, rewrite the previous interior sum, and use the Cauchy-Schwarz inequality:

$$\left[\sum_{\lambda_i} \left(\sum_{\boldsymbol{\lambda}'} \gamma_{\boldsymbol{\lambda}} \prod_{j} \varphi_{\lambda_j}(x_j) \right) \varphi_{\lambda_i}(x_i) \right]^2 \leq$$

$$\sum_{\lambda_i} \left(\sum_{\boldsymbol{\lambda}'} a_{\lambda_i} \cdot \gamma_{\boldsymbol{\lambda}} \prod_{j} \varphi_{\lambda_j}(x_j) \right)^2 \cdot \sum_{\lambda_i} \left(\frac{1}{a_{\lambda_i}} \varphi_{\lambda_i}(x_i) \right)^2 .$$

Hence, the squared norm of the potential $\|\eta\|^2_{L^2(\Omega)}$ is less than

$$\int_0^1 \cdots \int_0^1 \sum_{\lambda_i} \left(\sum_{\boldsymbol{\lambda}'} a_{\lambda_i} \cdot \gamma_{\boldsymbol{\lambda}} \prod_{j \neq i} \varphi_{\lambda_j}(x_j) \right)^2 \prod_{j \neq i} dx_j \cdot \int_0^1 \sum_{\lambda_i} \left(\frac{1}{a_{\lambda_i}} \varphi_{\lambda_i}(x_i) \right)^2 dx_i\,.$$

We have seen in the previous section, inequality (5.9), that the last integral is less than 1/6. Using the equality (5.10), we get

$$\|\eta\|^2_{L^2(\Omega)} \leq \int_0^1 \cdots \int_0^1 \sum_{\lambda_i} \left(\sum_{\boldsymbol{\lambda}'} \beta^i_{\boldsymbol{\lambda}} \prod_{j \neq i} \varphi_{\lambda_j}(x_j) \right)^2 \prod_{j \neq i} dx_j \cdot \frac{1}{6}\,. \tag{5.12}$$

The relations (5.11) and (5.12) show that

$$6\,\|\eta\|^2_{L^2(\Omega)} \leq \left\| w^i \right\|^2_{L^2(\Omega)}\,.$$

As i was arbitrary fixed, this concludes the proof. □

The general case of l-forms in d dimensions is much more complex. Consider a sparse Whitney $(l+1)$-form $\omega \in V_n^{l+1}$, with $\mathbf{d}\omega = 0$, that is

$$\omega = \sum_{|J|=l+1} w^J dx_J \,,$$

where

$$w^J(\mathbf{x}) = \sum_{\lambda} \beta_\lambda^J \Psi_{\boldsymbol{\nu}\mathbf{q}}^J(\mathbf{x}) \,.$$

The existence of the discrete potentials ensures the solvability of the system

$$\sum_{t=1}^{l+1} (-1)^{t+1} a_{\lambda_{j_t}} \gamma_\lambda^{J\setminus\{j_t\}} = \beta_\lambda^J \,, \tag{5.13}$$

for the coefficients γ_λ^I of the potential. In the previous particular cases we could write down the solution. Then, we used some computational tricks on it. In the general case, this approach is thwarted by the generality of the exterior derivative operator $\mathbf{d}$.

The challenge resides exactly from the description of the solution of system (5.10) and not in the instability of the hierarchical hat functions. Biorthogonal spline-wavelets systems on intervals are known [26, 27]. There is the possibility of the construction of such systems on sparse grids, as in [75]. For the full grid case, Urban [28, 84] constructed stable wavelet bases in $\boldsymbol{H}(\mathrm{div};\Omega)$ and $\boldsymbol{H}(\mathbf{curl};\Omega)$. This construction is close related to the solution of the system (5.13), such that it is not clear how to convey it to the general case. However, we computed numerically the stability constant C, as described in the final of Sect. 5.3. The results for $d = 4$ give existence of stable discrete potentials in the general case.

n	5	6	7	8	9	10	11	12
$l = 1$	0.2247	0.2251	0.2251	0.2251	0.2251	0.2251	0.2251	0.2251
$l = 2$	0.3182	0.3183	0.3183	0.3183	0.3183	0.3183	0.3183	0.3183

n	4	5	6	7	8
$l = 1$	0.1834	0.1837	0.1838	0.1838	0.1838
$l = 2$	0.2247	0.2247	0.2251	0.2251	0.2251
$l = 3$	0.3178	0.3183	0.3183	0.3183	0.3183

Table 5.2: Numerically computed stability constant C in the cases $d = 3$ (top) and $d = 4$ (bottom)

Chapter 6

Algorithms and Data Structures

The two pervasive ideas in the theory of the sparse grids, namely the hierarchical decomposition and the reduction to the one dimensional case via tensor product, feature prominently in the algorithms. The unidirectional scheme is the basis of the multilevel transformations. The combination technique is the key to the approximate interpolation. As the hierarchical bases lead to dense system matrices, the mass matrix multiplication is intricate and needs particular attention. The stiffness matrix multiplication combines the exterior derivative operator and the mass matrix. Then we present in the next chapter the multigrid method for the Whitney forms on sparse grids as a multiplicative scheme from the family of the semicoarsening techniques. For the multigrid solver we need stencils for mass and stiffness matrices corresponding to the most general case of the anisotropic full grids. Generality refers to three parameters: the dimension d of the space, the order l of the differential form, and the multi-dimensional resolution n of the grid. In the final section we examine the computational costs of the multigrid algorithm and study its convergence empirically.

First, we describe and motivate the algorithms in a general form. and avoid any consideration on programming language and data structure. We postpone technical and implementation details until the last two sections of this chapter. However, we use some `C++`-terminology, as "container" and "iterator", but their meaning will be clear in the context. They should be understood rather as general concepts than in the strict `C++` sense.

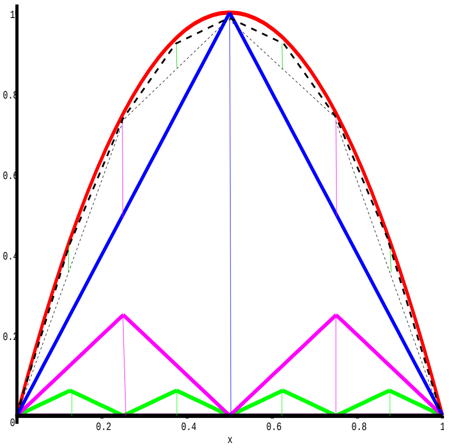


Figure 6.1: Hierarchical Coefficients

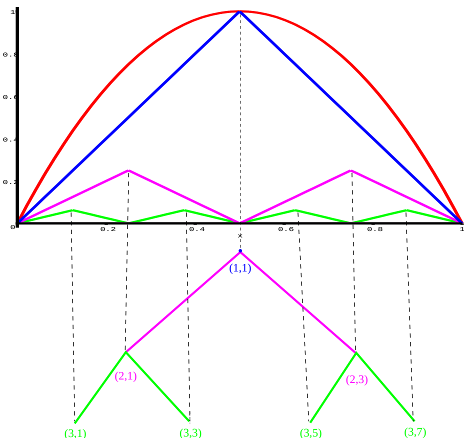


Figure 6.2: Hierarchical Data Structure

6.1 Multilevel Transforms

The hierarchical decomposition is reflected in the algorithms as the natural tree structure of data. The tensor product mirrors as the unidirectional scheme.

For the theory, the advantage of the hierarchical representation is the quick decay (depending on the local smoothness of the function to interpolate) of the hierarchical coefficients (Fig. 6.1). This property can be used for the adaptive representation of functions. In this context, the data structure is naturally hierarchical, see Fig. 6.2. We may think the data structure in terms of trees, independently of the concrete data organization.

The tensor product enables us to reduce the algorithms to one-dimensional considerations. Consider first that we have to represent hierarchically an l-form in d-dimensions on an anisotropic full grid of resolution $\mathbf{n}$. With the notations from Sect. 3.2, we identify each face $F = [x_{\boldsymbol{\nu}\mathbf{q}}; \vec{e}_{i_1}, \vec{e}_{i_2}, \ldots, \vec{e}_{i_l}]$ with its corresponding (nodal) degree of freedom $\alpha_F := \int\limits_F \omega$. As suggested in Sect. 3.2, we must work on the components of the Whitney form and we must first perform the wavelet transform and then the hierarchical transform. Both steps follow the unidirectional principle developed in [6, 8, 19, 23, 29]: reduce the task to the one dimensional case by recursion. In order to describe such an algorithm, we use two main tools: a container keeping the relevant points, and an iterator which runs over points in the hierarchical order. In the tree terminology, this means the level-order traversal [73]. Moreover, given a point of the current grid, we can extract a $(d-1)$-dimensional and an 1-dimensional orthogonal subgrids containing that point. Imagine that we just cut the grid over a line parallel to one of the coordinate axes or over the corresponding orthogonal $(d-1)$-plane. The set of all points forms a container, and each of the sections is a container similar to the original one, adapted to its dimension. Starting from an initial point, we can iterate over the points on the line (the "right" loop in the following) or over the points on the $(d-1)$-plane (the "left" loop).

Whatever the multilevel transformation is, it follows the unidirectional scheme. We describe this scheme intuitively as a traversal of the data structure, before we give a formal description in Fig. 6.4.

```
if ($d == 1$)
    • one dimensional transform;
else
    • run over the ``right'' loop (last coordinate is varying):
      for each point on this line take the orthogonal ($d-1$)-section
      and apply recursion ($d := d-1$);
    • run over the ``left'' loop (first $d-1$ coordinates are varying):
      for each point on this ($d-1$)-plane take the orthogonal one
      dimensional section and apply recursion ($d := 1$);
```

In the last two steps, we run over the same points but we look at the grid from different perspectives. As example, we present the history of the algorithm in the case $d = 2$, and $\mathbf{n} = (2, 3)$ only with interior points $(\boldsymbol{\nu}\mathbf{q}) =$ (level index). Compare to Fig. 6.3.

- $(1,1)(1,1)$ run on line parallel to Ox: $(1,1)(1,1)$, $(2,1)(1,1)$, $(1,1)(3,1)$
- $(1,2)(1,1)$ run on line parallel to Ox: $(1,2)(1,1)$, $(2,2)(1,1)$, $(1,2)(3,1)$
- $(1,2)(1,3)$ run on line parallel to Ox: $(1,2)(1,3)$, $(2,2)(1,3)$, $(1,2)(3,3)$
- $(1,3)(1,1)$ run on line parallel to Ox: $(1,3)(1,1)$, $(2,3)(1,1)$, $(1,3)(3,1)$
- $(1,3)(1,3)$ run on line parallel to Ox: $(1,3)(1,3)$, $(2,3)(1,3)$, $(1,3)(3,3)$
- $(1,3)(1,5)$ run on line parallel to Ox: $(1,3)(1,5)$, $(2,3)(1,5)$, $(1,3)(3,5)$
- $(1,3)(1,7)$ run on line parallel to Ox: $(1,3)(1,7)$, $(2,3)(1,7)$, $(1,3)(3,7)$

Now, change perspective:

- $(1,1)(1,1)$ run on line parallel to Oy: seven points
- $(2,1)(1,1)$ run on line parallel to Oy: seven points
- $(2,1)(3,1)$ run on line parallel to Oy: seven points

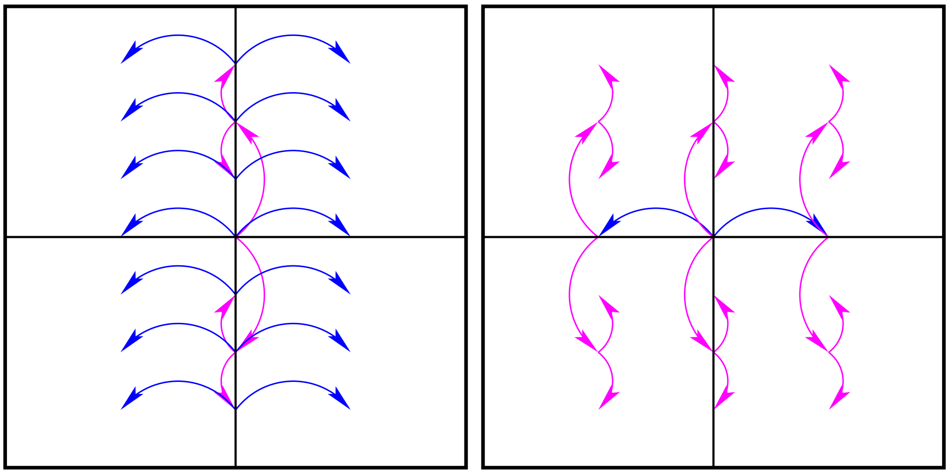


Figure 6.3: First (y) and second (x) perspective for $d = 2, \mathbf{n} = (2, 3)$

In the pseudocode description of the unidirectional scheme (Fig. 6.4), we denote the grid extraction functions by `subgrid2()` and `subgrid1()` for the $(d-1)$- and the 1-dimensional variants, respectively. More precisely, starting from a point `P` which corresponds to $(\boldsymbol{\nu}\mathbf{q})$, `subgrid2()` extracts the subgrid in the $(d-1)$-dimensional plane containing the point `P` and being parallel to the first $d-1$ directions. In the traversal of this subgrid, the $\nu_1, \ldots, \nu_{d-1}$ and $q_1, \ldots, q_{d-1}$ vary, whereas ν_d and q_d are fixed. On the contrary, the function `subgrid1()` extracts the subgrid in the line containing the point `P` and being parallel to the last direction. In the traversal of this subgrid, the $\nu_1, \ldots, \nu_{d-1}$ and $q_1, \ldots, q_{d-1}$ are fixed, whereas ν_d and q_d vary. The point `P` indicated by an iterator `y` may be obtained by `y`$\rightarrow$`P`. The beginning of a container is returned by `begin()` function, while `end()` points immediately after the last entry in the container.

Hence, we have to provide the implementation of the hierarchical, the wavelet, the transpose and the inverse transforms **only for the one dimensional case**.

In the description of the hierarchical transforms, we follow the guiding lines of [90] and [6]. The *hierarchical neighbors* of a node i are the two geometrical nearest grid points higher situated in the hierarchy (at a lower discretization level). Suppose that we have two functions *left* and *right* that provide the hierarchical neighbors. Then the hierarchical coefficients are

```
transform(int $d$, container $u$)
  {
    if($d == 1$) { ... // 1D transform }
    else
      {
        container right = u.subgrid1(u.begin()→P)
        iterator rit = right.begin()
        while( rit != right.end() ) {
            container left = u.subgrid2(rit→P)
            transform($d-1$, left)
            ++rit
          }
        container left = u.subgrid2(u.begin()→P)
        iterator lit = left.begin()
        while( lit != left.end() ) {
            container right = u.subgrid1(lit→P)
            transform(1, right)
            ++lit
          }
      }
  }
```

Figure 6.4: Unidirectional scheme

computed from the nodal coefficients by

$$v_i = u_i - \frac{1}{2}(u_{left(i)} + u_{right(i)}) \,, \tag{6.1}$$

whereas the backward transformation is

$$u_i = v_i + \frac{1}{2}(u_{left(i)} + u_{right(i)}) \,. \tag{6.2}$$

With an iterator which traverses the tree data structure in level-order, we copy the nodal values in a vector u. That is, we transform the hierarchical structure to a linear one. Now, the hierarchical neighbors of the element i (at level l less than the resolution n) are just the $i - 2^{n-l}$ and $i + 2^{n-l}$ elements. The hierarchical representation arises from the nodal coefficients over the bottom-up loop:

```
for $l$ from $n$ by 1 downto 1 do
    for $i$ from $2^{n-l}$ by $2^{n-l+1}$ to $2^n - 1$ do
        $u[i] := u[i] - \frac{1}{2}(u[i - 2^{n-l}] + u[i + 2^{n-l}])$   .
```

The back transformation is the top-down loop:

```
for $l$ from 1 by 1 to $n - 1$ do
    for $i$ from $2^{n-l-1}$ by $2^{n-l+1}$ to $2^n - 1$ do
        $u[i] := u[i] + \frac{1}{2}(u[i - 2^{n-l-1}] + u[i + 2^{n-l-1}])$   .
```

The Haar-wavelet transform follows the known pyramid algorithm [76, 78], schematically presented in Fig. 6.5. At the bottom of a pyramid reside the scale coefficients, and at the top sit the wavelet coefficients. The wavelet transform acts bottom-up, whereas the reverse transform goes top-down. In order to avoid rounding error, we do not norm the Haar-wavelets. For a

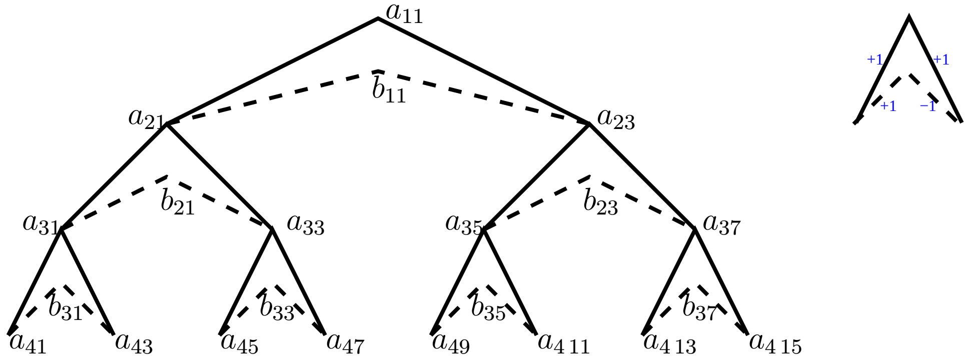


Figure 6.5: Pyramid algorithm

vector u, the wavelet transform reads

```
for $l$ from $n$ by 1 downto 1 do
      for $i$ from $2^{n-l}$ by $2^{n-l+1}$ to $2^n - 1$ do
            $a := u[i]$
            $u[i] := \frac{1}{2}(u[i] - u[i + 2^{n-l}])$
            $u[i + 2^{n-l}] := \frac{1}{2}(a + u[i + 2^{n-l}])$  ,
```

whereas the scale transform is:

```
for $l$ from 1 by 1 to n do
      for $i$ from $2^{n-l}$ by $2^{n-l+1}$ to $2^n - 1$ do
            $a := u[i]$
            $u[i] := u[i] + u[i + 2^{n-l}]$
            $u[i + 2^{n-l}] := -a + u[i + 2^{n-l}]$  .
```

The Haar-wavelets are orthogonal, so the scale and the wavelet transform are transposed to each other, up to a scaling due to the use of the not normed basis functions. We finish this section by the transpose (with respect to the Euclidian inner product) of the hierarchical and the nodal transformations. The transpose of the hierarchical transform is:

```
for $l$ from 1 by 1 to n do
      for $i$ from $2^{n-l}$ by $2^{n-l+1}$ to $2^n - 1$ do
            $u[i - 2^{n-l}] := u[i - 2^{n-l}] - \frac{1}{2}u[i]$
            $u[i + 2^{n-l}] := u[i + 2^{n-l}] - \frac{1}{2}u[i]$
```

and the transpose of the nodal transform is:

```
for $l$ from $n$ by 1 downto 1 do
      for $i$ from $2^{n-l}$ by $2^{n-l+1}$ to $2^n - 1$ do
            $u[i - 2^{n-l}] := u[i - 2^{n-l}] + \frac{1}{2}u[i]$
            $u[i + 2^{n-l}] := u[i + 2^{n-l}] + \frac{1}{2}u[i]$.
```

6.2 Interpolation

In the previous section, we used containers corresponding to full grids, as we were concerned by anisotrop full grids. Now, we deal with the interpolation on sparse grids, so the containers and the iterators correspond to the sparse grids, as in Fig. 6.6. The general unidirectional scheme (`Algorithm transform`) applies to the sparse grid in the case of 0-forms. In the case of

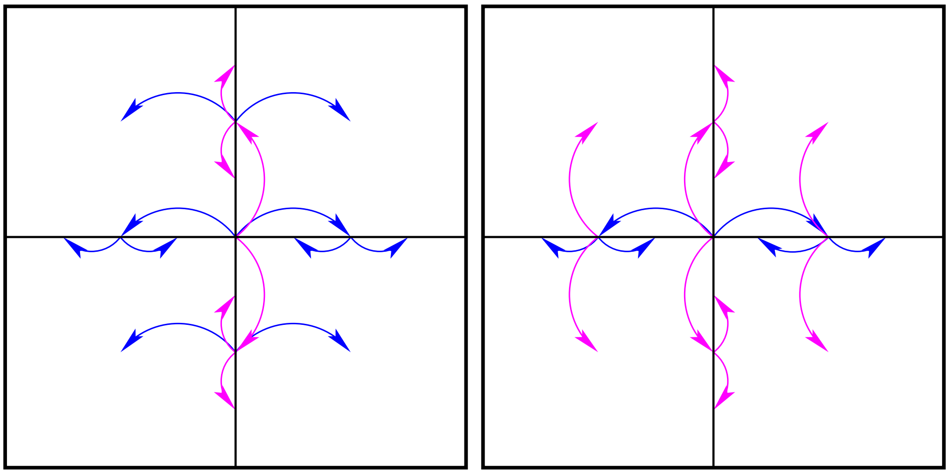


Figure 6.6: First (y) and Second (x) Perspective for $d = 2, n = 3$

general l-forms $(1 \leq l \leq d)$, the degrees of freedom are integrals over l-faces. If we have all this "nodal" coefficients at our disposal, we can apply the wavelet transform for the directions from I and the hierarchical transform for the other directions, obtaining the "hierarchical" coefficients. But this algorithm needs the storage of *all* degrees of freedom corresponding to the *full grid*. If we refer only to the sparse grid, we drop out essential information. The nodal coefficients are **not** details, they are necessary. Let us explain in detail this new aspect that crops up when moving from 0-forms to general l-forms.

We are analytically given an l-form ω in d dimensions, for instance the right hand side of some partial differential equation, as in Sect. 4.2. The task is to represent ω hierarchically on the sparse grid. In the case of 0-forms, the nodal coefficients from the sparse grid are enough for the interpolation. The degrees of freedom are in this case just the values of the function in the grid points. Replacing the degrees of freedom for l-forms $(l > 0)$ by these nodal values of functions leads to an extreme wavelet crime (in the sense of Strang [78]), which destroys the convergence. Indeed, since the degrees of freedom are integrals over faces, we are not allowed to approximate them only by the value in one point. The elements have different dimensions, so a naive one or two point quadrature rule cannot work. Bluntly speaking, if we do so, we squander essential information.

A solution may be an *adaptive quadrature rule*: for larger faces, employ more evaluations points. As proved in [33], for 1-forms at level m, the trapezoidal rule with 2^{n-m} points on each edge of length 2^{-m} is enough for the convergence. We do not present here this approach, since its computational complexity is a major drawback. Let us see why.

For 0-forms we need $2^n\left(\frac{n^{d-1}}{(d-1)!}+O(n^{d-2})\right)$ function evaluations, as showed in [23]. For one coordinate of the vector representative of an l-form ($1 \leq l \leq d-1$), we need a number of function evaluations equal with 2^{ln} times the number of points in the sparse grid in a $(d-l)$-dimensional cube, that is

$$2^{ln} \times 2^n\left(\frac{n^{d-l-1}}{(d-l-1)!}+O(n^{d-l-2})\right).$$

For $l=d$, we need 2^{dn} function evaluations, as for the full grid. The values or the points are not stored, so *no extra memory is required.*

Suppose that we perform a Galerkin-discretization of the equation (4.16) from Sect. 4.2, based on the space of Whitney-l-forms on sparse grids, and we have at our disposal some efficient solver, as the multigrid method. In this case, the adaptive quadrature rule, used for the evaluation of the right hand side would *destroy the rapid convergence in the multilevel algorithm.* This is the reason of looking for another way to evaluate the interpolation operator.

In Sect. 4.2, we presented the approximate interpolation and we proved that the convergence order from Theorem 4.1.1 holds for this operator, too. We mentioned that the implementation relies on the combination technique [42, 63, 64, 70].

The combination formula (4.3) tells us that the interpolant $\mathcal{I}_n^I u_I$ of u_I into the sparse grid space V_n^I is just a linear combination of the interpolants of u_I into full anisotropic spaces $V_{\boldsymbol{\nu}}^I = \text{Range}(\mathcal{I}_{\boldsymbol{\nu}}^I)$, with $|\boldsymbol{\nu}|_1 = n+d-s$. The dimension of these spaces is of order $O(2^n)$. For each component $\mathcal{I}_{\boldsymbol{\nu}}^I u_I$, we perform the wavelet transform in directions from I and the hierarchical transform for the other directions. In this way, we obtain the hierarchical representation of $\mathcal{I}_{\boldsymbol{\nu}}^I u_I$. The addition of two functions defined on different grids is now just the addition of single coefficients. This is due to the fact that *the hierarchical basis functions that are associated with the grid points which belong to both grids agree.* We simply add the hierarchical representations, in order to construct the interpolant of u_I in the hierarchical basis. Remark that we collect only the hierarchical coefficients corresponding to sparse grid points. But the others **can** be dropped, because they are indeed details, yet. We stress that *the use of combination technique avoids the premature elimination of details and the squander of information.*

The approximate interpolation operator from Sect. 4.2 emerges by replacing the "exact" interpolation operators $\mathcal{I}_\nu^{\mathrm{const}}$ with those based on the two point trapezoidal rule $\tilde{\mathcal{I}}_\nu^{\mathrm{const}}$. Hence, for the computation of $\tilde{\mathcal{I}}_{\boldsymbol{\nu}}^I u_I$, we need only the values of u_I in the sparse grid points. On each anisotropic mesh, we apply the two point trapezoidal rule, using an unidirectional scheme. We present an example in Fig. 6.7, where the focus is on the 2-form in 2 dimensions given by the vector proxy $u(x,y) = 4x(1-x)\cdot 4y(1-y)$.

We pursue the example presenting in Fig. 6.8 the steps of the interpolation algorithm based on the combination technique. For each interpolant $\mathcal{I}_{\boldsymbol{\nu}}^I u_I$ we apply first the wavelet transform in the x-direction and then in the y-direction. Finally, we simply add the coefficients. The hierarchical coefficients corresponding to points not belonging to the sparse grid are just 0, because the associated basis functions are not in the sparse grid space. The computation of the nodal coefficients on the sparse grid is not as simple as for the hierarchical coefficients. It needs another interpolation, from low to high level grids. Dropping the nodal coefficients corresponding to points not existing in the sparse grid is not allowed: they are not details, as the bottom line of Fig. 6.8 clearly illustrates. This is the reason to exploit the implicit form of the nodal representation, given by the combination formula (4.4).

We want to describe the interpolation algorithm via combination technique as pseudocode. In order to represent a Whitney form on a sparse grid in a computer code, we have to store the coefficients of some kind of basis representation. The most sensible choice is the hierarchical basis representation. For a discrete l-form on the sparse grid $\omega_n \in \mathcal{S}_n$ it reads

$$\omega_n = \sum_I \left(\sum_{|\boldsymbol{\nu}|_1 \le n+d-1} \sum_{\mathbf{q}\in\Theta_{\boldsymbol{\nu}}} \beta^I_{\boldsymbol{\nu}\mathbf{q}} \Psi^I_{\boldsymbol{\nu}\mathbf{q}} \right) dx_I \,. \tag{6.3}$$

The $\beta^I_{\boldsymbol{\nu}\mathbf{q}}$ are the hierarchical coefficients and they uniquely determine ω. Assuming a sequential ordering of the index sets, the hierarchical coefficient can be lumped into a vector $\vec{\omega} = (\beta^I_{\boldsymbol{\nu}\mathbf{q}})_{\boldsymbol{\nu},\mathbf{q},I}$. In fact, the arrow $\vec{}$ can be regarded as an operator that maps the Whitney l-form ω_n to its hierarchical coefficients.

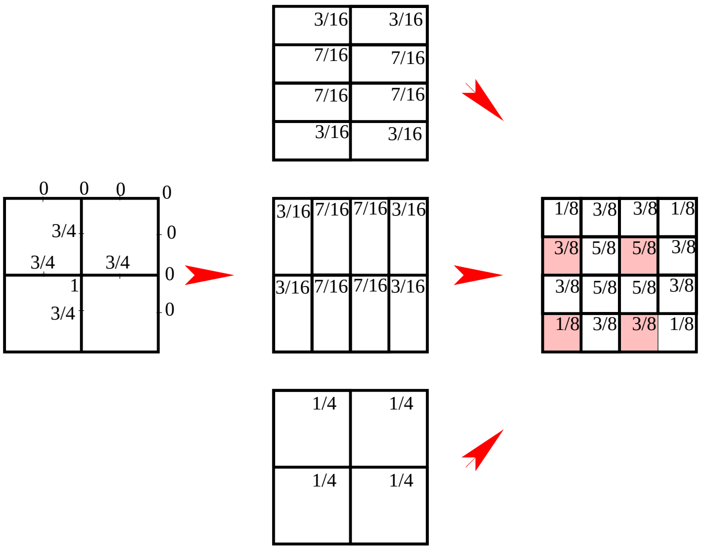


Figure 6.7: Interpolation on sparse grids of 2-forms in two dimensions

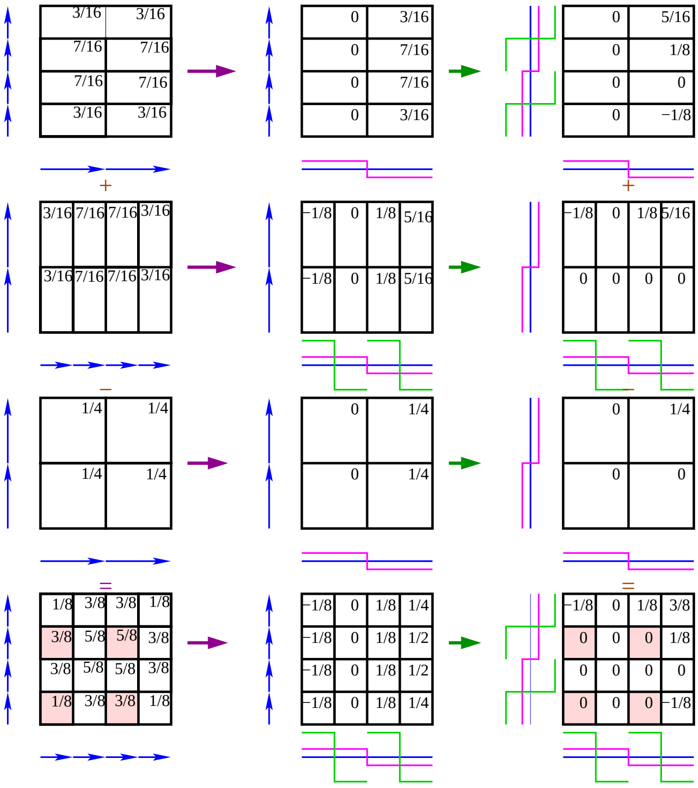


Figure 6.8: Combination technique for 2-forms in the case of $d = 2$ and $n = 2$

Definition 6.2.1. *By $\mathcal{S}_n$ and $\mathcal{A}_{\boldsymbol{\nu}}$ we refer to the spaces of coefficient vectors corresponding to the sparse grid and the anisotropic full grid discretization of resolutions n and $\boldsymbol{\nu}$, respectively.*

We don't discuss here the data structure used to store the coefficients $\beta^I_{\boldsymbol{\nu}\mathbf{q}}$. Rather, we take for granted that such a data structure is available and offers an interface in the form of the function `access(`I`,`P`)`, which accesses the value corresponding to the index I and grid point $P = (\boldsymbol{\nu}\mathbf{q})$. As discussed in Sect. 3.2, by a grid point we mean a tuple $(\boldsymbol{\nu}\mathbf{q})$ of a level index $\boldsymbol{\nu}$ and a position index $\mathbf{q} \in \Theta_{\boldsymbol{\nu}}$. A crucial assumption is that it takes a constant amount of time to retrieve any coefficient from the data structure. For instance, hash tables come close to meeting this requirement [71, Sect. 6.2]. We present our particular choice in Sect. 6.4.

Suppose that a reading procedure provides us with the values of the components of the (continuous) differential form in the grid points. Then, for the construction of the Whitney forms living on anisotropic full grids from the combination formula (4.4), we need an extraction procedure `extract` that amounts to mere copying of certain components of the coefficient vectors .

```
extract($\vec{\rho} \in \mathcal{S}_n$, $\vec{\lambda} \in \mathcal{A}_{\boldsymbol{\nu}}$)

  {
    for each point $P$ in the grid of $\mathcal{A}_{\boldsymbol{\nu}}$ and each index $I$
          $\vec{\lambda}$.access($I$,$P$) = $\vec{\rho}$($I$,$P$)
  }
```

Figure 6.9: Extraction of the hierarchical coefficients corresponding to an anisotropic full grid

For a Whitney form on an anisotropic full grid declarated simply by $\mathcal{A}_{\boldsymbol{\nu}}\ \ \lambda(\vec{\lambda})$, we have the function `trap_2p()`, which follows the unidirectional scheme and which, for the directions with constant interpolation, uses the two point trapezoidal rule. The transformation to the hierarchical coefficients `to_wh()` follows the unidirectional scheme, too (as discussed in the previous section). In the one dimensional case, it uses the wavelet transform for the directions with interpolation by constants and the hierarchical transform otherwise. In Fig. 6.10 we describe the combination technique in pseudocode. Note that the addition of Whitney forms living on a sparse grid and an underlying anisotropic full grid is done in such a way that the added degrees of freedom agree. The vector $\vec{\omega} \in \mathcal{S}_n$ is initialized by zero, and $\vec{\rho} \in \mathcal{S}_n$ contains the values of the components of the continous differential form in the grid points.

```
combin_tech(const &$\vec{\rho} \in \mathcal{S}_n$, &$\vec{\omega} \in \mathcal{S}_n$)

  {
    for each $s = 1, 2, \ldots, d$ {

        for each level $\boldsymbol{\nu}$ such that $|\boldsymbol{\nu}|_1 = n + d - s$:

            extract($\vec{\rho}$,$\vec{\lambda}$); $\mathcal{A}_{\boldsymbol{\nu}}$ $\lambda(\vec{\lambda})$
            $\lambda$.trap_2p()
            $\lambda$.to_wh()
            $\omega$ += $(-1)^{s-1}\binom{d-1}{s-1}$ $\lambda$
        }
  }
```

Figure 6.10: Combination technique procedure

```
interp_to_full(const $\vec{\rho} \in \mathcal{S}_n$, &$\vec{\omega} \in \mathcal{A}_{\mathbf{n}}$)

  {
    for each $s = 1, 2, \ldots, d$ {

        for each level $\boldsymbol{\nu}$ such that $|\boldsymbol{\nu}|_1 = n + d - s$:

            extract($\vec{\rho}$,$\vec{\lambda}$); $\mathcal{A}_{\boldsymbol{\nu}}$ $\lambda(\vec{\lambda})$
            $\lambda$.trap_2p()
            $\lambda$.an_to_is()
            $\omega$ += $\lambda$
        }
  }
```

Figure 6.11: Procedure for the interpolation from the sparse to the full grid

We finish this section by the computation of the error produced by the approximate interpolation operator. In order to evaluate numerically the interpolation error, we compute first by `interp_to_full` the values of the

Whitney form in the full grid points. This can be done directly, using the implicit form of the nodal representation (4.4). We start again from the values of the component of the (continuous) differential form in the grid points and we write in a vector corresponding to a Whitney form on the full isotropic grid of resolution n. In Fig. 6.11, `an_to_is()` interpolates from the anisotropic to the isotropic full grid. This can be easily done by recursion. For simplicity, we give here only a crude description:

- `for each point in the` $(d-1)$`-dimensional anisotropic full grid perform 1D-interpolation (fill in the` d`-direction)`
- `if` $d > 1$`, for each point in the obtained 1D full structure apply recursion to the` $(d-1)$`-dimensional case.`

Once we know the Whitney form on the full grid, we employ a 2-point Gauss quadrature formula in order to obtain the L^2-norm of error: $\|\omega - \mathcal{I}_n\omega\|_{L^2(\Omega)}$. We rely on the tensor product structure for a recursive unidirectional algorithm.

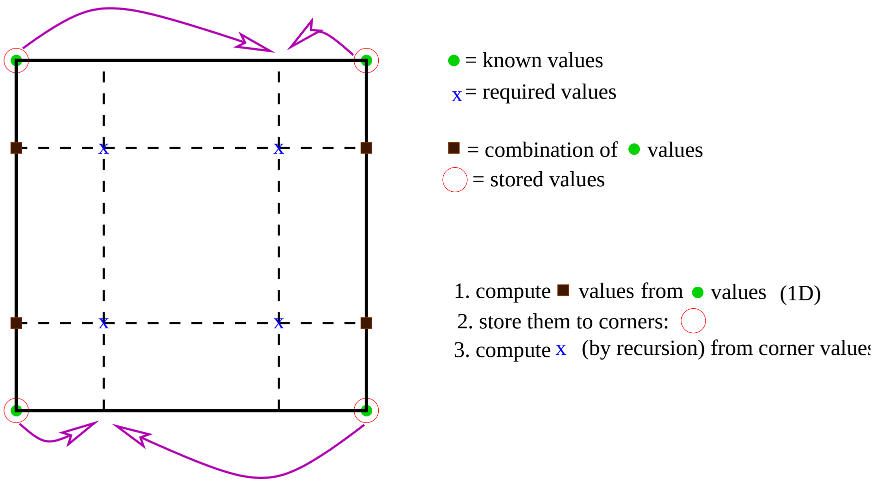

● = known values
x = required values
■ = combination of ● values
○ = stored values
1. compute ■ values from ● values (1D)
2. store them to corners: ○
3. compute x (by recursion) from corner values

Figure 6.12: Algorithm schema for interpolation in Gauss points

We don't need supplementary storage space for the Gauss quadrature points. Compute first the local error, that is the corresponding integral on each hyper-cube, and then simply sum over all elements in the grid. For one hyper-cube, we compute by the unidirectional scheme the values of all

coordinate functions of $\mathcal{I}_n\omega$ in the 2^d Gauss points and we store them in the vertices (see figure 6.12 for the two dimensional case).
In the next run over the vertices of the hyper-cube, we apply the Gauss quadrature rule with 2^d points for d-dimensions. We obtain an order $O(2^{-4n})$ approximation of the error $\|\omega - \mathcal{I}_n\omega\|_{L^2(\Omega)}$. Working with the full grid means that we cannot perform the previous computations for very fine resolutions n. For reasons of storage capacity and computational time, we obtained the estimations only for $n = 12$ in the two dimensional case and for $n = 8$ in the three dimensional case. see Fig. 4.5, 4.6. This clearly illustrates the curse of dimensionality and the remedy proposed by the sparse grid approach (see Fig. 4.3).

6.3 Mass and Stiffness Matrix Multiplications

We want to discretize (2.4) by means of conforming finite elements on sparse grids, e.g. by means of sparse Whitney forms at a resolution n. Here, by resolution we mean the discretization level n in the definition 4.1.1 of the spaces of sparse Whitney forms. For the sake of simplicity we only consider the coefficients $\alpha = \beta = 1$ as in Sect. 4.2. Abbreviate the bilinear form in the variational problems by

$$a(\omega,\eta) := (\mathbf{d}\omega, \mathbf{d}\eta)_{L^2(\Omega)} + (\xi,\eta)_{L^2(\Omega)} \quad \text{and} \quad b(\eta) := (\xi,\eta)_{L^2(\Omega)} \,.$$

The homogeneous Dirichlet boundary conditions can easily be taken into account: since the degrees of freedom for a Whitney l-form are defined as integrals over the corresponding l-faces (see Sect. 2.2), all we have to do is to set the degrees of freedom located on the boundary to zero. Thus, finite element subspaces of $\mathbf{H}_0(\mathbf{d},\Omega)$ can be obtained.

Plugging the representation (6.3) into the variational problem (5.1) and testing with all hierarchical basis functions, we obtain a large linear system

$$A_l\,\vec{\omega} = \vec{\eta}\,. \tag{6.4}$$

Here A_l is the matrix arising from plugging the hierarchical basis functions into the bilinear form $a(\cdot,\cdot)$. As $a(\cdot,\cdot)$ is the sum of two contributions, A_l can be split into a mass matrix M_l, corresponding to $(\omega,\eta) \mapsto (\omega,\eta)_{L^2(\Omega)}$, and a stiffness matrix S_l arising from $(\omega,\eta) \mapsto (\mathbf{d}\omega,\mathbf{d}\eta)_{L^2(\Omega)}$. As we use the hierarchical representation, these matrices **not sparse**. However, as we use iterative schemes for the solution of (6.4), we need not to compute them explicitly, but only need to multiply them with a vector of degrees of freedom.

To perform the multiplication of a vector with A_l in the case of general dimension d, we must apply the so-called uni-directional strategy as described for 0-forms (Lagrangian finite elements) in [23]. Let us first consider the mass matrix multiplication. Denoting $v_{\boldsymbol{\eta},\mathbf{r},I} := \Psi^I_{\boldsymbol{\eta}\mathbf{r}} dx_I$, we have to compute the L^2-scalar product

$$(M\vec{\omega})_{\boldsymbol{\eta},\mathbf{r},J} = (\omega_n, v_{\boldsymbol{\eta},\mathbf{r},J}) = \sum_{|\boldsymbol{\nu}|_1 \leq n+d-1} \sum_{\mathbf{q}\in\boldsymbol{\Theta}_{\boldsymbol{\nu}}} \beta^J_{\boldsymbol{\nu}\mathbf{q}} \left(\Psi^J_{\boldsymbol{\nu}\mathbf{q}}, \Psi^J_{\boldsymbol{\eta}\mathbf{r}}\right)_{L^2(\Omega)} ,$$

because only the components belonging to the same multi-index I are coupled. The tensor product structure enables us to write each scalar product in the above sum as

$$\left(\Psi^J_{\boldsymbol{\nu}\mathbf{q}}, \Psi^J_{\boldsymbol{\eta}\mathbf{r}}\right)_{L^2(\Omega)} = \prod_{j\in J} \left(\psi_{\nu_j q_j}, \psi_{\eta_j r_j}\right)_{L^2([0,1])} \cdot \prod_{i\notin J} \left(\varphi_{\nu_i q_i}, \varphi_{\eta_i r_i}\right)_{L^2([0,1])} .$$

As the Haar-wavelets are orthogonal, the first terms in the above product are 1 if $\nu_j = \eta_j$ and $q_j = r_j$, and 0 otherwise. Hence, the directions in which the I-component is constant causes no difficulties as far as multiplication with the mass matrix M_l is concerned. For each component of the vector proxy of the l-form, the problem reduces to a mass matrix multiplication for 0-forms in $d - l$ dimensions. Consequently, we can apply the well known algorithm for Lagrangian finite elements [6, Sect. 3.1] to the case of general Whitney l-forms.

Let us recall the fundamental idea [6, 19, 23] for the mass matrix multiplication in the one dimensional case. Test a Whitney 0-form u_h with a fixed hierarchical basis function $\varphi_{k\,j}$

$$(u_h, \varphi_{k\,j}) = (\sum_{l\,i} u_{l\,i}\varphi_{l\,i}\,,\, \varphi_{k\,j})$$

and decompose the last expression into a sum over the hat functions $\varphi_{l\,i}$, whose support **contain** the support of $\varphi_{k\,j}$, and a sum over the hat functions $\varphi_{l\,i}$, whose support **is contained** by the support of $\varphi_{k\,j}$.

$$(u_h, \varphi_{k\,j}) = (\sum_{(l\,i)\supset(k\,j)} u_{l\,i}\varphi_{l\,i}\,,\, \varphi_{k\,j}) + (\sum_{(l\,i)\subseteq(k\,j)} u_{l\,i}\varphi_{l\,i}\,,\, \varphi_{k\,j})\,. \tag{6.5}$$

By $(l\,i) \supset (k\,j)$ we mean the set of those $(l\,i)$ such that $\operatorname{supp}(\varphi_{l\,i}) \supset \operatorname{supp}(\varphi_{k\,j})$, and $(l\,i) \subseteq (k\,j)$ is the set of those $(l\,i)$ such that $\operatorname{supp}(\varphi_{l\,i}) \subseteq \operatorname{supp}(\varphi_{k\,j})$. Remark that the sums correspond to the hierarchical predecessors and, respectively, successors of the actual node $(k\,j)$. Note also, it is not important which sum the "middle" term $(\varphi_{k\,j}\,,\,\varphi_{k\,j})$ joins.

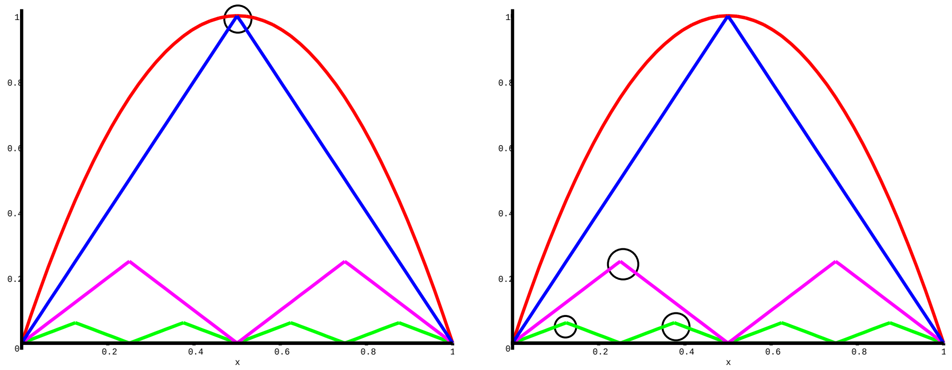


Figure 6.13: Nodes involved by `upper` and `under`

We present an example in Fig. 6.13 for $k = 2$ and $j = 1$: $(2\,1)$, $(3\,1)$, $(3\,3) \subseteq (2\,1)$, whereas $(1\,1) \supset (2\,1)$. Balder [6, p. 26] observed that for the L^2 scalar product it holds

$$\left(\sum_{(l\,i) \supset (k\,j)} u_{l\,i} \varphi_{l\,i}\,,\,\varphi_{k\,j} \right) = h_k \frac{1}{2} (u_{left(k,j)} + u_{right(k,j)})\,,$$

which remembers the backward transformation (6.2). Hence, the first sum can be computed during a top-down pass through the data structure in hierarchical order. It is named `upper` because it accumulates terms from the upper part of actual node. The second sum can be computed during a bottom-up pass through a copy of the data structure. It is named `under` because it accumulates terms sitting under the actual node. Afterwards, the two structures are added. Clearly, the data structure contains now the result of the mass matrix multiplication.

The same idea emerges in the d-dimensional case. We follow the guiding lines of [6, 8]. Test the sparse Whitney 0-form u_h with the d-dimensional hat function $\varphi_{\mathbf{kj}}$:

$$(u_h, \varphi_{\mathbf{kj}}) = \int\limits_{[0,1]^d} u_h \varphi_{\mathbf{k}\,\mathbf{j}} = \sum_{\substack{|\mathbf{l}|_1 \le n+d-1 \\ \mathbf{l} \in \mathbb{N}^d}} \sum_{\mathbf{i} \in \mathbf{I}_\mathbf{l}} u_{\mathbf{l}\,\mathbf{i}} \prod_{\nu=1}^{d} \int\limits_0^1 \varphi_{l_\nu i_\nu} \cdot \varphi_{k_\nu j_\nu} \,. \tag{6.6}$$

First, apply the unidirectional idea, that is the idea of the reduction to the one dimensional case. We isolate the terms corresponding to the last coordinate

$$\begin{aligned}(u_h, \varphi_{\mathbf{k},\mathbf{j}}) &= \\ &\sum_{\substack{|\mathbf{l}'|_1 \le n+d-2 \\ \mathbf{l}' \in \mathbb{N}^{d-1}\ \mathbf{i}' \in \mathbf{I}_{\mathbf{l}'}}} \left(\sum_{l_d=1}^{n+d-1-|\mathbf{l}'|_1} \sum_{i_d \in I_{l_d}} u_{(\mathbf{l}',l_d)(\mathbf{i}',i_d)} \int\limits_0^1 \varphi_{l_d i_d} \cdot \varphi_{k_d j_d} \right) \prod_{\nu=1}^{d-1} \int\limits_0^1 \varphi_{l_\nu i_\nu} \cdot \varphi_{k_\nu j_\nu} \\ &= \sum_{\substack{|\mathbf{l}'|_1 \le n+d-2 \\ \mathbf{l}' \in \mathbb{N}^{d-1}}} \sum_{\mathbf{i}' \in \mathbf{I}_{\mathbf{l}'}} v_{(\mathbf{l}',k_d)(\mathbf{i}',j_d)} \int\limits_{[0,1]^{d-1}} \varphi_{\mathbf{l}'\,\mathbf{i}'} \cdot \varphi_{\mathbf{k}'\,\mathbf{j}'} \,,\end{aligned} \tag{6.7}$$

where we denoted

$$v_{(\mathbf{l}',k_d)(\mathbf{i}',j_d)} = \sum_{l_d=1}^{n+d-1-|\mathbf{l}'|_1} \sum_{i_d \in I_{l_d}} u_{(\mathbf{l}',l_d)(\mathbf{i}',i_d)} \int\limits_0^1 \varphi_{l_d i_d} \cdot \varphi_{k_d j_d} \,.$$

This means that we sum first over the terms of an 1-dimensional case, and then over the terms corresponding to a $(d-1)$-dimensional case. Why should we prefer this order of summation? In fact, we can also sum in the reverse order. That is, we can write the formula (6.6) in another way:

$$\begin{aligned}(u_h, \varphi_{\mathbf{k},\mathbf{j}}) &= \\ &\sum_{l_d=1}^{n+d-2} \sum_{i_d \in I_{l_d}} \left(\sum_{\substack{|\mathbf{l}'|_1 \le n+d-1-l_d \\ \mathbf{l}' \in \mathbb{N}^{d-1}\ \mathbf{i}' \in \mathbf{I}_{\mathbf{l}'}}} u_{(\mathbf{l}',l_d)(\mathbf{i}',i_d)} \int\limits_{[0,1]^{d-1}} \varphi_{\mathbf{l}'\,\mathbf{i}'} \cdot \varphi_{\mathbf{k}'\,\mathbf{j}'} \right) \int\limits_0^1 \varphi_{l_d i_d} \cdot \varphi_{k_d j_d} = \\ &= \sum_{l_d=1}^{n+d-2} \sum_{i_d \in I_{l_d}} w_{(\mathbf{k}',l_d)(\mathbf{j}',i_d)} \int\limits_0^1 \varphi_{l_d i_d} \cdot \varphi_{k_d j_d} \,,\end{aligned} \tag{6.8}$$

where we denoted the sum corresponding to the first $d-1$ coordinates

$$w_{(\mathbf{k}',l_d)(\mathbf{j}',i_d)} = \sum_{\substack{|\mathbf{l}'|_1 \le n+d-1-l_d \\ \mathbf{l}' \in \mathbb{N}^{d-1}}} \sum_{\mathbf{i}' \in \mathbf{I}_{\mathbf{l}'}} u_{(\mathbf{l}',l_d)(\mathbf{i}',i_d)} \int\limits_{[0,1]^{d-1}} \varphi_{\mathbf{l}'\,\mathbf{i}'} \cdot \varphi_{\mathbf{k}'\,\mathbf{j}'} \,.$$

Unfortunately, we can not use directly neither (6.7) nor (6.8). The problem is that it is possible to have $|(\mathbf{l}', k_d)|_1 > n+d-1$ for the first case, or $|(\mathbf{k}', l_d)|_1 > n+d-1$ for the second. This means that the implementation of one of the two representation would need more storage points than actually existing, such that the sparse grid would degenerate in a full one, see [6, pp. 29-30].

The solution is to mix the two representations and to exploit the idea presented for the one dimensional case:

$$(u_h, \varphi_{\mathbf{k},\mathbf{j}}) = T_1 + T_2 \,.$$

The first term T_1 corresponds to the first order of summation, but restricting the one dimensional sum only to the hat functions $\varphi_{l_d i_d}$ whose support is contained in the support of $\varphi_{k_d j_d}$

$$T_1 = \tag{6.9}$$
$$\sum_{\substack{|\mathbf{l}'|_1 \le n+d-2 \\ \mathbf{l}' \in \mathbb{N}^{d-1}\ \mathbf{i}' \in \mathbf{I}_{\mathbf{l}'}}} \left(\sum_{\substack{1 \le l_d \le n+d-1-|\mathbf{l}'|_1 \\ i_d \in I_{l_d} \\ (l_d\, i_d) \subseteq (k_d\, j_d)}} u_{(\mathbf{l}',l_d)(\mathbf{i}',i_d)} \int\limits_0^1 \varphi_{l_d i_d} \cdot \varphi_{k_d j_d} \right) \prod_{\nu=1}^{d-1} \int\limits_0^1 \varphi_{l_\nu i_\nu} \cdot \varphi_{k_\nu j_\nu} \,.$$

The second term T_2 corresponds to the second order of summation, restricting the one dimensional sum only at hat functions $\varphi_{l_d i_d}$ whose support contains in the support of $\varphi_{k_d j_d}$

$$T_2 = \tag{6.10}$$
$$\sum_{\substack{1 \le l_d \le n+d-2 \\ i_d \in I_{l_d} \\ (l_d\, i_d) \supset (k_d\, j_d)}} \left(\sum_{\substack{|\mathbf{l}'|_1 \le n+d-1-l_d \\ \mathbf{l}' \in \mathbb{N}^{d-1}\ \mathbf{i}' \in \mathbf{I}_{\mathbf{l}'}}} u_{(\mathbf{l}',l_d)(\mathbf{i}',i_d)} \int\limits_{[0,1]^{d-1}} \varphi_{\mathbf{l}'\,\mathbf{i}'} \varphi_{\mathbf{k}'\,\mathbf{j}'} \right) \int\limits_0^1 \varphi_{l_d i_d} \varphi_{k_d j_d} \,.$$

The difficulty encountered before does not arise anymore: if $(l_d\, i_d) \supset (k_d\, j_d)$, then $l_d < k_d$, so $|\mathbf{k}'|_1 + l_d < |\mathbf{k}'|_1 + k_d \leq n + d - 1$ (just what we wanted for the second sum), and if $(l_d\, i_d) \subseteq (k_d\, j_d)$, then $l_d \geq k_d$, so $|\mathbf{l}'|_1 \leq n + d - 1 - l_d \leq n + d - 1 - k_d$, that is $|\mathbf{k}'|_1 + k_d \leq n + d - 1$ (appropriate for the first sum). We can therefore use a recursion schema in order to compute separately (6.9) and (6.10) once for the whole vector u. As the formulas show, the recursion schema should act differently on two copies of the vector, as presented in Figure 6.14, where `‘‘under’’` computes the interior sum in (6.9) and `‘‘upper’’` computes the exterior sum in (6.10). They are procedures that refer only to one dimensional grids. We sketch the data modification during the algorithm for the case $d = 2$, $n = 3$. Remark “changing perspective” in descending the tree. The recursion acts in following way for the first sum (see figure 6.6):

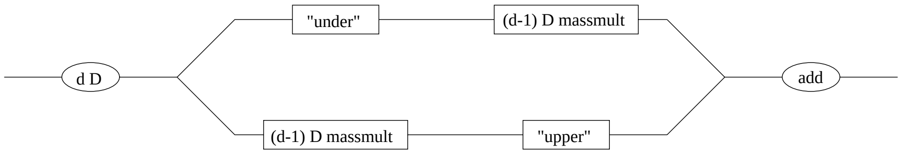


Figure 6.14: Recursion schema for mass matrix multiplication

- $(1,1)(1,1)$ `‘‘under’’` in direction y : write to positions

  (1,1)(1,1)
  (1,2)(1,1) (1,2)(1,3)
  (1,3)(1,1) (1,3)(1,3) (1,3)(1,5) (1,3)(1,7)

- $(2,1)(1,1)$ `‘‘under’’` in direction y : write to positions

  (2,1)(1,1)
  (2,2)(1,1) (2,2)(1,3)

- $(2,1)(3,1)$ `‘‘under’’` in direction y : write to positions

  (2,1)(3,1)
  (2,2)(3,1) (2,2)(3,1)

- $(3,1)(1,1)$ `‘‘under’’` in direction y : write to position (3,1)(1,1)
- $(3,1)(3,1)$ `‘‘under’’` in direction y : write to position (3,1)(3,1)
- $(3,1)(5,1)$ `‘‘under’’` in direction y : write to position (3,1)(5,1)
- $(3,1)(7,1)$ `‘‘under’’` in direction y : write to position (3,1)(7,1)

Now change perspective, in order to do recursion:

- $(1,1)(1,1)$ `massmult` in direction x: write to positions

  (1,1)(1,1)
  (2,1)(1,1) (2,1)(3,1)
  (3,1)(1,1) (3,1)(3,1) (3,1)(5,1) (3,1)(7,1)

- $(1,2)(1,1)$ `massmult` in direction x: write to positions

  (1,2)(1,1)
  (2,2)(1,1) (2,2)(3,1)

- $(1,2)(1,3)$ `massmult` in direction x: write to positions

  (1,2)(1,3)
  (2,2)(1,3) (2,2)(3,3)

- $(1,3)(1,1)$ `massmult` in direction x: write to position (1,3)(1,1)
- $(1,3)(1,3)$ `massmult` in direction x: write to position (1,3)(1,3)
- $(1,3)(1,5)$ `massmult` in direction x: write to position (1,3)(1,5)
- $(1,3)(1,7)$ `massmult` in direction x: write to position (1,3)(1,7)

The second sum is computed in a similar way. The only difference is that it applies first recursion and then `‘‘upper’’`. Changing the perspective may provide difficulties when the working structure is a tree. In that case, one should add a tree with the other perspective (it points to the same addresses corresponding to grid points, of course). Our version uses associative arrays or hash tables instead of trees and avoids this problem. We describe the mass matrix multiplication algorithm in pseudocode in Fig. 6.15. The parameters are a (sparse grid)container u and three integers: d and n refer to the dimension and resolution of the actual sparse grid, whereas s is related to the memory management. The latter will be explained in detail in Sect. 6.5. Also related to the memory management are the vectors of degrees of freedom $p[i]$ which appear in Fig. 6.15 as a global parameter in the addition step. Moreover, we use an auxiliary function `get1n()`, which returns the $|\cdot|_1$-norm of the fixed elements in the level of the actual container; the aim is to pass the correct resolution to the corresponding subgrid in recursion.

It is noteworthy that only the substructure to be used has to be copied, otherwise, the linear complexity of the algorithm is lost.

Concerning the multiplication by the stiffness matrix, there are new aspects that come into play because of the generality of l-forms. For the rest

```
massmult(container $u$, int $d$, $n$, $s$)

  {
    container v(u)
    container lu = u.subgrid2(u.begin()→P)
    iterator lit = lu.begin()
    while( lit != lu.end() ) {

        container ru = u.subgrid1(lit→P)
        under(ru, $n+d-1-$ru.get1n($d$), $s$)
        ++lit
      }
    if($d>1$) {

        container ru = u.subgrid1(u.begin()→P)
        iterator rit = ru.begin()
        while( rit != ru.end() ) {

            container lu = u.subgrid2(rit→P)
            massmult(lu, $d-1$, $n+1-$lu.get1n($d$), $s-1$)
            ++rit
          }
        container rv = v.subgrid1(v.begin()→P)
        iterator rit = rv.begin()
        while( rit != rv.end() ) {

            container lv = v.subgrid2(rit→P)
            massmult(lv, $d-1$, $n+1-$lv.get1n($d$), $s$)
            ++rit
          }
      }
    container lv = v.subgrid2(v.begin()→P)
    iterator lit = lv.begin()
    while( lit != lv.end() ) {

        container rv = v.subgrid1(lit→P)
        upper(rv, $n+d-1-$rv.get1n($d$), $s-d+1$)
        ++lit
      }
    $p[s-d]$ += $p[s-d+1]$ // addition step
  }
```

Figure 6.15: Algorithm for the multiplication by the mass matrix

of this section we concentrate on the multiplication by the stiffness matrix. We want to compute the L^2-scalar product

$$(d\omega_n, dv_{\boldsymbol{\eta},\mathbf{r},I}) =: (S\vec{\omega})_{\boldsymbol{\eta},\mathbf{r},I} \,.$$

As $\mathbf{d}V_n^l \subset V_n^{l+1}$, we can write D for the matrix of the exterior derivative on the sparse grid, and we have obviously

$$S\vec{\omega} = D^T M_{l+1} D\vec{\omega} \,.$$

As proved in [6] for the case of Lagrangian finite elements, the mass matrix multiplication algorithm needs a number of operations proportional to the number of grid points. Hence, it remains to design an algorithm with the same property for the computation of the exterior derivative of a Whitney form.

We begin by clarifying the ordering of the components of an l-form $\omega = \sum_I u_I dx_I$. Each component u_I is associated with the index $I = \{i_1, \ldots, i_l\} \subset \{1, \ldots, d\}$ with $i_1 < i_2 < \cdots < i_l$. By the definition of the exterior derivative

$$d\omega = \sum_I (du_I) \wedge dx_I = \sum_I \left(\sum_{k=1}^{d} \frac{\partial}{\partial x_k} u_I \, dx_k \right) \wedge dx_I \,,$$

so we have to insert k at its appropriate place in I, such that $I \cup \{k\}$ is a valid index. As $dx_k \wedge dx_I = 0$ if $k \in I$, for the Whitney l-from ω_n the exterior derivative is

$$d\omega_n = \sum_I \sum_{k \notin I} \left(\sum_{|\boldsymbol{\nu}|_1 \le n+d-1} \sum_{\mathbf{q} \in \Theta_{\boldsymbol{\nu}}} \beta^I_{\boldsymbol{\nu}\mathbf{q}} \frac{\partial}{\partial x_k} \Psi^I_{\boldsymbol{\nu}\mathbf{q}} \right) dx_k \wedge dx_I \,.$$

As the derivative of a hat function is a Haar-wavelet, we have

$$d\omega_n = \sum_I \sum_{k \notin I} \left(\sum_{|\boldsymbol{\nu}|_1 \le n+d-1} \sum_{\mathbf{q} \in \Theta_{\boldsymbol{\nu}}} \beta^I_{\boldsymbol{\nu}\mathbf{q}} \cdot 2^{(\nu_k+1)/2} \Psi^{I\cup\{k\}}_{\boldsymbol{\nu}\mathbf{q}} \right) dx_k \wedge dx_I \,.$$

Hence, we compute the vector of degrees of freedom $S\vec{\omega}$ corresponding to an $(l+1)$-form by the above sum. The conclusion is that we obtain the exterior derivative and the multiplication by the stiffness matrix in a number of operations that is proportional to the number of sparse grid points. In pseudocode notation, the algorithm for the computation of the discrete exterior derivative $\vec{\mu} = D\vec{\omega}$ is depicted in Fig. 6.16.

```
for each point $P = (\boldsymbol{\nu}, \mathbf{q})$ and index $I$
    for each $t \in \{1, \ldots, d\} \setminus I$ {
        sgn $=$ card$\{i \in I;\ i < t\}$
        $s := (-1)^{\texttt{sgn}}$; $J = I \cup \{t\}$
        $c_t$ = $s$ * $2^{(\nu_t+1)/2}$
        $\vec{\mu}$.access($J$,$P$) = $c_t$ * $\vec{\omega}$.access($I$,$P$)
    }
```

Figure 6.16: Algorithm for the evaluation of the exterior derivative

6.4 Software Abstractions

So far, we have examined theory and general algorithms for sparse Whitney forms. But the way from these concepts to an efficient code is still long. Now, we explain our choice of the data structure together with different details of the code design. We stress that the efficiency of the mass matrix multiplication algorithm can be destroyed if we do not pay attention to the data management. We describe a solution to this problem, in the context of our data abstraction.

The theory of the sparse grids is reflected in the algorithms through the intuitive hierarchical structure of data and the recursivity. The complex algorithms from the previous chapter need a flexible and modular design. Abstract data types and object oriented programming lend themselves as proper tools in this context [93]. Of course, we must be careful about the over-use of design features, as the overloading the arithmetic operators "+" and "$*$". As emphasized, the natural data structure for sparse grids are trees. Some codes used trees directly and incorporated numerical schemes into the algorithms that manipulate data structures [72, 92]. In the object oriented style, there is a separation of the tree traversal and the arithmetic operation. We adopt an approach closed to the ideas presented in [93]. We split the abstractions in six main building blocks of concepts: *vector, grid, form, container, iterator*, and *operator.* Apart from those, we have some other minor entities to describe points, stencils, and the pool of memory.

Design. The two main ideas are to separate the data storage from the algorithm and to get rid of pointers.

Vectors store large amounts of numerical data. They can be efficiently handled in the linear algebra operations. We used a C^{++}-expression templates library for dense linear algebra [57]. A lot of time is spent in the compilation of the code, due to the use of the templates, but the resulting code is as fast as if we had directly called BLAS routines [57].

Grids refer to the geometry, they contain points without numerical data. In the tree version, we need $3d$ pointers per node and procedures for the tree traversals. Hence, a lot of memory and time are spent for administration. Difficulties arise from the organization of the pointers (remember the change of the perspective in the algorithm for the mass matrix multiplication) and from the efficiency of the cache (the nodes sit away from each other in memory). The alternative to the pointers is the key-based addressing in terms of associative arrays or of hash tables [19, 38, 43, 71, 93]. The former at our disposal in the C^{++} Standard Template Library (STL), see [79]. For efficiency, our data structure is built upon a hash table. The implementation of a hash table as hash map was realized by SGI in public domain [74] and may be added to a future revision of the C^{++}-standard. A *hash table* (see [53] or [73] for details) is a linear array of cells or buckets. The key is used as address in the hash table where the set of data is stored. If different sets of data have the same key, collisions occur and buckets appear. The hash function links the data and the key. A *hash function* is a unary function and must be deterministic (that is, it must always return the same value whenever it is called with the same argument), but return values of the hash function should be as uniform as possible. Ideally, no two keys will hash to the same value, which is not possible, since the hash table has fever cells than there are possible keys.

Each node has a position in space. Its local coordinates correspond to a `level` and `index`. Both of them build an `Index` that is a key entry in the hash table. Hence, we map each point to the value stored in the hash table, which is an integer. This integer gives the corresponding position in a vector (see Figure 6.17).

The hash table, together with auxiliary functions (as `begin, end, find`) is stored in the `class Grid`. The pure virtual class `AdGrid` (adaptive grid) inherits the `Grid`. The adaptation can yield an anisotropic full grid `AFGrid` or a sparse grid `SpGrid`.

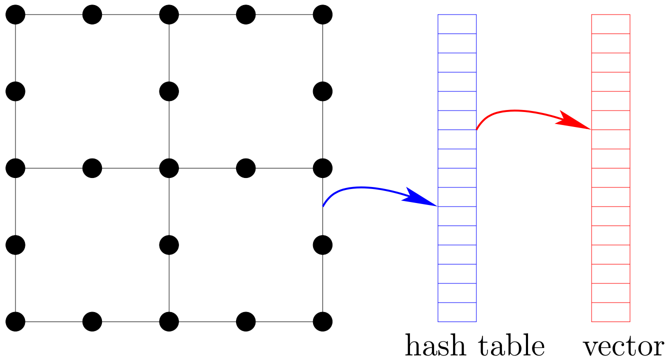
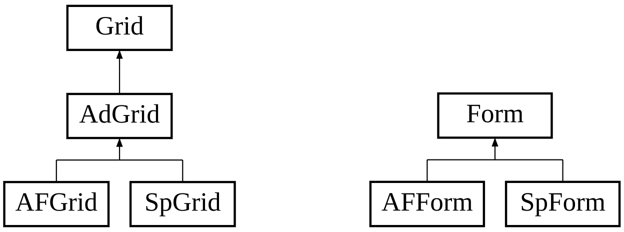


Figure 6.17: Acces to coefficients of a sparse grid function: the points are mapped to values in the hash table; these integers give the position in the data vector.

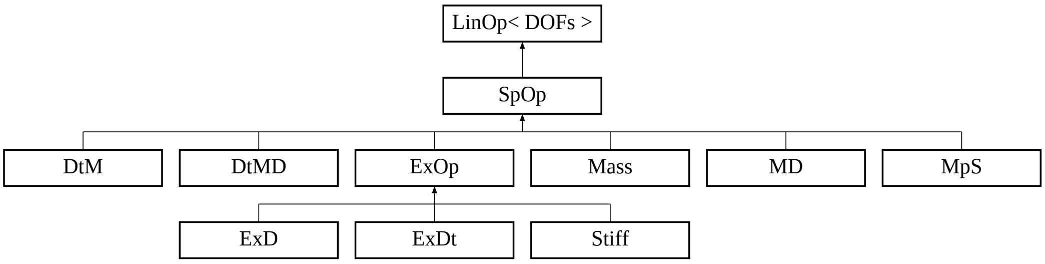


Forms bring together the geometry stored in the corresponding grid as a reference and the discretization vector as a handle. The class `Form` implements a general Whitney l-form, `AFForm` on an anisotropic grid, and `SpForm` on a sparse grid. Each of them posses the corresponding hierarchical transformations, too.

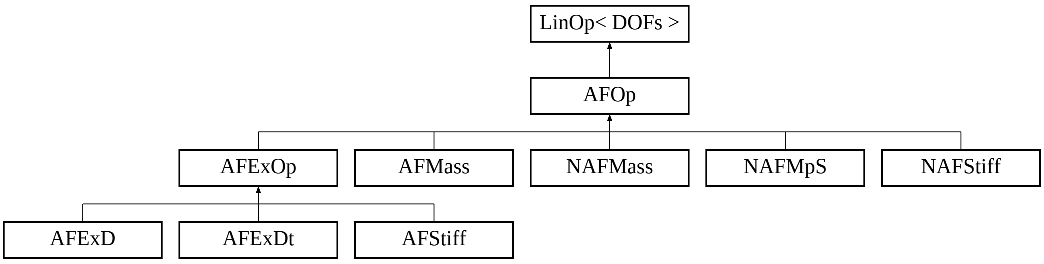
The *containers* and the corresponding *iterators* are fundamental in the algorithms presented in the previous chapter. They are tailored to the grids: sparse grid containers (`SPG`) and iterators (`SPG_iterator`) or anisotropic full grid containers (`AFG`) and iterators (`AFG_iterator`).

An `SPG` is the abstraction of a part of the tree of grid points. The set of elements in container is actually not stored once again, but only the address of the map with the data, the first element (offspring), the current resolution and a mask. These ellemnts allow the traversal of the container by the corresponding iterator, which selects the particular elements from the hash-table. The mask offer the possibility to extract subgrids of the `SPG`, namely sparse grids in lower dimensions. The tree traversing is done in a levelwise fashion. This means that the corresponding iterator will run over the points in lexicographic order. Reaching the end of the `SPG` means that the next index is the origin (offspring) of the `SPG`. We represent the end of the `SPG` by the end of the map, that is `SPG::end()` returns an `SPG_iterator` that points at the end of the map. The corresponding `SPG_iterator` inherits the class `iterator_traits` and it is only a forward iterator. The anisotropic counterparts of `SPG` and `SPG_iterator` are implemented analogously.

The *operators* are designed for the corresponding discretisations, too. They stand for matrices and act only by means of their `operator()` . They receive two arguments: a constant handle to the input vector and a handle to the result vector. The sparse operator class `SpOp` is pure virtual, inherited from the linear operator class `LinOp`. In the class hierarchy, `SpOp` is the father of the class `Mass`, which implements the mass matrix multiplication algorithm. The other main son of `SpOp` is the class `ExOp`, which refers to the exterior operators on sparse grids. Being pure virtual, it gives rise to the exterior derivative operator `ExD`, the transpose `ExDt`, and the stiffness matrix operator `Stiff`.



The operators corresponding to the anisotropic grid are similarly developed. Having the very general stencils at our disposal, we can perform the mass and the stiffness matrix operations on anisotropic full grids more efficiently. These operators are implemented in the classes `NAFMass` and `NAFStiff`. Moreover, the exterior derivative operator on anisotropic full grids and its transpose can be directly subordinated to the linear operator class `LinOp`.



We refer to comments contained in the code for more documentation on the classes.

Hash Statistics Essential for the performance of the code is the quality of the hash table, that is, the quality of the hash function. The hash table implementation of the STL uses linked lists for the resolution of collisions and automatic resizing. We rely on the hash function proposed by Griebel [38] and Schiekofer [71], which has been proved appropriate in the context of the adaptive sparse grids. Denoting as usual, the d dimensional level by $\boldsymbol{\nu}$ and the index by $\mathbf{q}$, the hash function is defined as

$$H(\boldsymbol{\nu}, \mathbf{q}) = \sum_{i=1}^{d} (2^{\nu_i} + q_i) \cdot P(i) \cdot P(43 + (d-2) \cdot 10 - i)\,,$$

where $P(i)$ is the i-th prime number. Now, we want a measurement of the quality of the hash table. First, we can look at the length b_j of the buckets j. Suppose that we access the data records randomly. Then, the probability to encounter a list of length m is

$$p(m) = \operatorname{card}\{j,\, b_j = m\} \cdot m/N\,,$$

where N is the total number of data records. With this notations, we can compute the average computational effort for accessing an entry as

$$E = \sum_{m=1}^{n} \frac{m+1}{2} p(m)\,.$$

The next tables present the statistics for $d = 4, 3$, and 2, respectively. For different levels, we give the number of buckets of length $|B|$ and the corresponding effort.

$l \backslash \lvert B \rvert$	1	2	3	4	5	N	effort
2	21	0	0	0	0	21	1
3	49	0	0	0	0	49	1
4	107	3	0	0	0	113	1.02655
5	236	9	1	0	0	257	1.04669
6	571	3	0	0	0	577	1.0052
7	1231	25	0	0	0	1281	1.01952
8	2635	88	2	0	0	2817	1.03337
9	6027	59	0	0	0	6145	1.0096
10	12927	193	0	0	0	13313	1.0145
11	28163	255	0	0	0	28673	1.00889
12	59999	721	0	0	0	61441	1.01173
13	126915	2079	0	0	0	131073	1.01586
14	268707	4911	0	0	0	278529	1.01763
15	556109	16855	2	0	0	589825	1.02859
16	1116085	63575	650	0	0	1245185	1.05262
17	2208381	204587	1294	1	0	2621441	1.07953
18	4237286	533139	66455	524	0	5505025	1.13363

Table 6.1: Number of buckets of different length for $d = 2$

$l \backslash \lvert B \rvert$	1	2	3	4	5	6	7	N	effort
3	219	3	0	0	0	0	0	225	1.01333
4	528	31	1	0	0	0	0	593	1.05734
5	1414	44	1	0	0	0	0	1505	1.03123
6	3633	40	0	0	0	0	0	3713	1.01077
7	8605	178	0	0	0	0	0	8961	1.01986
8	20093	578	0	0	0	0	0	21249	1.0272
9	44786	2414	17	0	0	0	0	49665	1.04963
10	94246	9551	443	3	0	0	0	114689	1.09502
11	189632	29018	4336	351	13	0	0	262145	1.16885
12	366677	67581	23158	5022	486	15	0	593921	1.29006

Table 6.2: Number of buckets of different length for $d = 3$

$l\backslash\|B\|$	1	2	3	4	5	6	7	N	effort
4	2703	33	0	0	0	0	0	2769	1.01192
5	7147	255	8	0	0	0	0	7681	1.03632
6	17464	1390	79	0	0	0	0	20481	1.07944
7	39000	5873	687	44	2	0	0	52993	1.15508
8	79339	18310	4581	907	107	4	0	133889	1.28849

Table 6.3: Number of buckets of different length for $d = 4$

6.5 Data Management in Mass Matrix Multiplication

The copy process in the mass matrix multiplication algorithm needs particular attention, otherwise it can destroy the complexity. We mention that we have to copy only a part of the vector, that is, only the values to be used later in the recursion. We present here our solution to this problem. We need exactly $d + 1$ copies. We show here the construction and the proof will be evident by induction.

We denote by N the length of the vector keeping the degrees of freedom. In the pictures, we figure a copy process by a vertical line with a bullet on top. Arrows and "+" signifies there an addition step. Copy and addition refer each time only to the degrees of freedom corresponding to the actual sparse grid, which can be t dimensional, with $1 \leq t \leq d$, the other vector entries being idle. The steps `under` and `upper` of the algorithm are denoted by "n" and "p", respectively.

First, consider the one dimensional case. We allocate two vectors $v[0]$ and $v[1]$ of length N, and suppose that the degrees of freedom are contained in $v[1]$. Following the algorithm, we make a copy of (in this case) the whole vector $v[0] := v[1]$. Then, we apply `under` on $v[0]$ and `upper` on $v[1]$. Finally, we add the two vectors, $v[0] := v[0] + v[1]$.

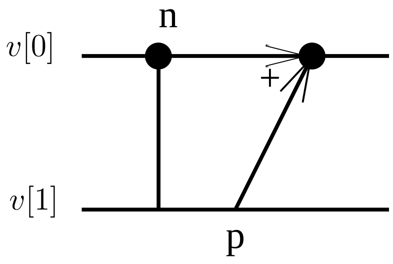


Figure 6.18: Map of the data modification for $d = 1$

Remark that the step `upper` reads and writes on the same vector. The procedure `under` can be considered to read from a vector and to write to the previous one.We express the application of `upper` to $v[s]$ by `upper(`s`)`, and we use `under(`s`)`, for the step `under` reading from $v[s]$ and writing to $v[s-1]$. The one dimensional `mass` procedure reads from $v[1]$ and writes in $v[0]$. In general, the block `mass(`d`,`s`)`, with $d \leq s$, reads from $v[s]$ and writes to $v[s-d]$.

Next, consider the two dimensional case. By recursion, we have two blocks of one dimensional `mass` acting on different vectors, connected by an application of `under` at the beginning, and a step of `upper`, followed by an addition at the end. With the previous notations, `mass(2,2)` perform first `under(2)`, then `mass(1,1)` and `mass(1,2)`, followed by `upper(1)`, and ends with the addition $v[0] := v[0] + v[1]$.

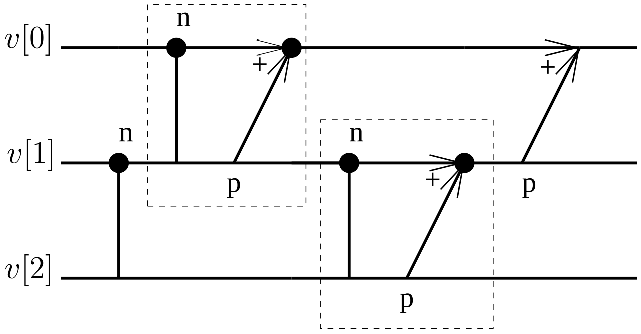


Figure 6.19: Map of the data modification for $d = 2$

Finally, we display in Fig. 6.20 the data modification for $d = 3$. By induction, we see at a glance that we copy each degree of freedom exactly $d + 1$ times.

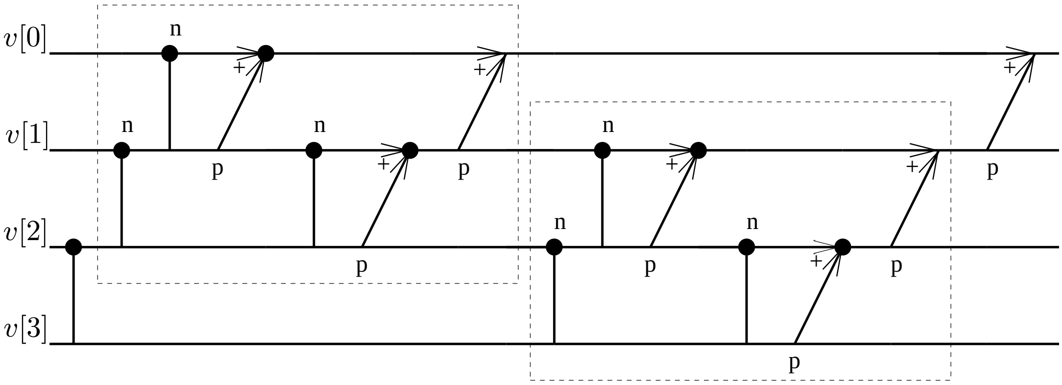


Figure 6.20: Map of the data modification for $d = 3$

In this chapter, the focus was on the general form of some algorithms involved in the sparse grid problems. The multilevel transformations illustrate the unidirectional principle. In the case of the general forms, the interpolation boils down to the approximate interpolation, done by the combination technique. We computed numerically the interpolation error on the full grid and we gave examples for smooth and not smooth forms, together with visualization links. Particular attentions wins the mass and the stiffness matrix multiplication algorithms. We presented in detail the multigrid scheme on sparse grids. Then, we addressed the automatical construction of stencils on anisotropic full grids in the most general setting. At the beggining, freedom was left in many aspects, as the choice of the data structure. In the last two sections, we explained our particular option, together with other implementation details. In the next chapter, the focus is on the multigrid method for sparse grid discretizations.

Chapter 7

Multigrid Solver

From the very beginning of the application of sparse grid schemes for the solution of elliptic boundary value problems, the fast solution of the resultinglinear systems of equations was an important issue. For standard finite element discretizations multigrid methods are known to provide the most efficient iterative solvers. The idea, in the form of multilevel Schwarz methods [88], instantly carries over to sparse grids, but it takes profound considerations to devise an efficient implementation. The latter is discussed in[38, 65], whereas a theoretical framework was provided in [39].

Second order elliptic boundary value problems represent a special case of boundary value problems for differential forms, namely the case of 0-forms. This indicates that it is possible to state a generic "elliptic" boundary value problem fitting any order of differential form as we described in 6.3 When using discrete differential forms on standard grids for the sake of Galerkin discretization, the guideline put forth above proved invaluable. Variants of the classical multigrid idea could be found that yield iterative solvers with a performance matching that of standard multigrid. These methods were presented in [3, 45] for the case of 2-forms, that is, $\boldsymbol{H}(\mathrm{div};\Omega)$-conforming finite elements, and in [4, 47] for discrete 1-forms, which are conforming in $\boldsymbol{H}(\mathbf{curl};\Omega)$. Now, we set out to rely on these ideas in the case of discrete differential forms on sparse grids.

7.1 Multigrid Method

The multigrid method for sparse grid discretizations of (2.4) uses the semicoarsening technique [22, 64]. It was implemented for the Lagrangian finite elements in the two and three dimensional case (see [64, 72], respectively).

The iterative scheme starts with an initial guess of solution ω_0. The residual is then $\rho_0(\eta) := b(\eta) - a(\omega, \eta)$, $\eta \in V_n^l$. The multigrid method can be viewed as a successive subspace correction scheme [37] with inexact subspace solvers on the subspaces $V_{\boldsymbol{\nu}}^l$ corresponding to anisotropic full grids of resolution $\boldsymbol{\nu}$. This gives rise to an outermost loop as described in Fig. 7.1.

```
For different $V_{\boldsymbol{\nu}}^l \subset V_n^l$:

  • Solve approximately $\zeta \in V_{\boldsymbol{\nu}}^l$:
    $a(\zeta, \eta) = \rho_i(\eta)$, for all $\eta \in V_{\boldsymbol{\nu}}^l \subset V_n^l$
  • update($\omega \in V_n^l, \rho \in V_n^l, \zeta \in V_{\boldsymbol{\nu}}^l$):

      * $\omega_{i+1} = \omega_i + \zeta$
      * $\rho_{i+1}(\eta) = b(\eta) - a(\omega_i + \zeta, \eta) = \rho_i(\eta) - a(\zeta, \eta)$
```

Figure 7.1: Exterior loop in multigrid

Note that the update of the solution renders the multigrid method a multiplicative scheme in the sense of Yserentant [89, 91]. The question is how to chose the spaces $V_{\boldsymbol{\nu}}^l$ and the ordering of the steps. Pflaum proposed the so-called Q-cycle for Lagrangian finite elements in two dimensions [64, 65]: it involves the spaces $V_{\boldsymbol{\nu}}^l \subset V_n^l$ on those anisotropic full subgrids that are not contained in any other anisotropic subgrid. They constitute the diagonal in the level scheme (see Fig. 7.2), i.e. the grids $\Omega_{\boldsymbol{\nu}}$ that come into play satisfy $|\boldsymbol{\nu}|_1 = n + d - 1$.

Then, at each step of the exterior loop approximately solve the residual equation for the correction by V-cycles based on semicoarsening in one direction. Schneider proposed in [72] to consider only one direction (S-cycle), whereas the Q-cycle of Pflaum involves all directions. Remember from Def. 6.2.1 that $\mathcal{S}_n$ and $\mathcal{A}_{\boldsymbol{\nu}}$ refer to the spaces of coefficient vectors corresponding to the sparse grid and the anisotropic full grid discretization of resolutions n and $\boldsymbol{\nu}$, respectively.

We give in Fig. 7.3 the formal description of the Q-cycle.

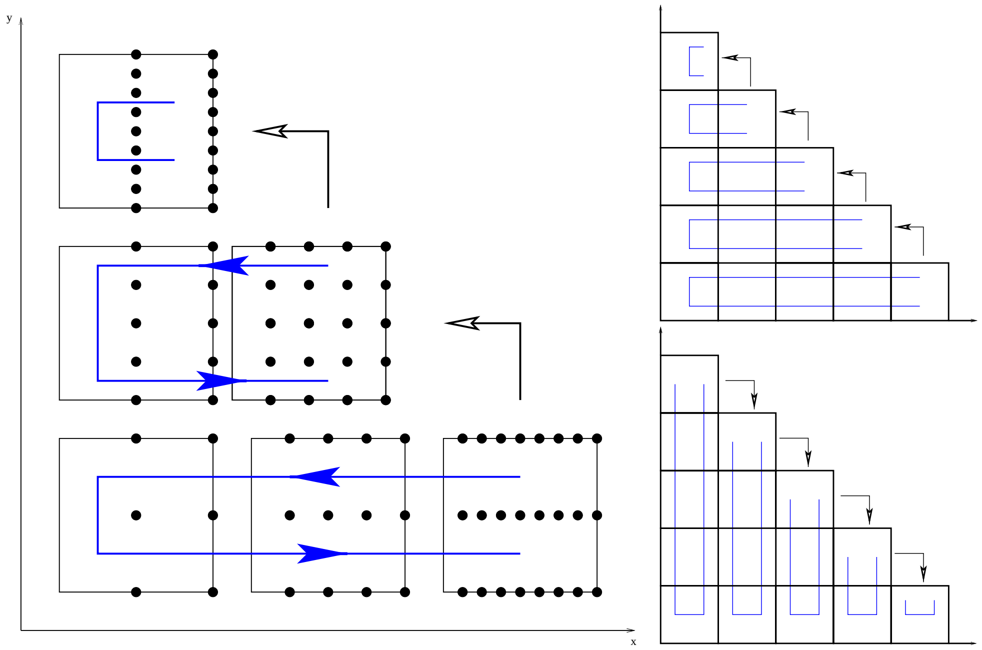


Figure 7.2: Q-cycle: one direction for $d = 2, n = 3, l = 1$ (left) and both directions for $d = 2, n = 5, l = 1$(right)

```
For each direction $j = 1, 2, \ldots, d$
    For levels $\boldsymbol{\nu}$ such that $|\boldsymbol{\nu}|_1 = n+d-1$:
        extloop_step($\vec{\omega}, \vec{\rho} \in \mathcal{S}_n$, $\boldsymbol{\nu}$, $j$)
```

Figure 7.3: Pseudocode for the Q-cycle

Let us study the the approximate correction on a single anisotropic full grid $\Omega_{\boldsymbol{\nu}}$ that forms the body of the exterior loop. This means that the solution ω and the residual ρ are updated with respect to the subspace $V^l_{\boldsymbol{\nu}}$. The computations can be split into three main steps (cf. Fig. 7.4): transfer of residual, approximate solve of residual equation, and transfer of correction back to the sparse grid.

In order to obtain the residual on the anisotropic grid in the nodal ba-

```
extloop_step($\vec{\omega}, \vec{\rho} \in \mathcal{S}_n$, $\boldsymbol{\nu}$, $j$)

  {
    extract($\vec{\rho}$, $\vec{\lambda}$)
    $\vec{\lambda} = H^T \vec{\lambda}$ // to nodal representation
    MGV($\nu_j$, $\vec{\zeta}$, $\vec{\lambda}$) // semicoarsening on full grid
    $\vec{\zeta} = H \vec{\zeta}$ // to hierarchical representation
    update($\vec{\omega}$, $\vec{\rho}$, $\vec{\zeta}$)
  }
```

Figure 7.4: One step in the exterior loop

sis we need two operations: the extraction of the hierarchical coefficients corresponding to the current anisotropic full grid and the transformation to the nodal coefficients of the residual on the anisotropic full grid. The first step, the extraction of the residual on an anisotropic full grid included in the sparse grid, is trivial in the hierarchical representation. This is due to the fact that the hierarchical basis functions that are associated with grid points that belong to both grids agree. Thus, `extract` amounts to mere copying of certain components of the coefficient vectors as in Fig. 6.9.

The local solve is the multigrid method based on semicoarsening in the fixed direction j on the anisotropic full grid of resolution $\boldsymbol{\nu}$. The semicoarsening merely acts on the anisotropic full grid and uses the nodal basis. As the residual is a functional, after the V-cycle has been conducted, we employ the transpose of the hierarchical transform. This poses no difficulties, since we have a *full grid* at our disposal: for the corrections on the anisotropic grids we have to implement a standard V-cycle based on semicoarsening. It targets a linear system of equations arising from a nodal discretizations of the bilinear form $a(\cdot,\cdot)$ on **anisotropic full** grids.

Meanwhile it has become clear how to device an efficient multigrid solver for such discrete problems [4, 44, 45, 47, 49]. The main insight is that plain point smoothers will fail. The culprit is the large kernel $\mathrm{Ker}(\mathbf{d})$ of the $\mathbf{d}$ operator in the case of l-forms ($1 \leq l \leq d-1$), because the performance of standard multilevel schemes for linear discrete variational problems essentially depends on the ellipticity of the bilinear form. But $a(\cdot,\cdot)$ behaves utterly differently on $Ker(\mathbf{d})$ and its orthogonal complement! If restricted to $\mathrm{Ker}(\mathbf{d})$, it agrees with the L^2-scalar product. Hence, in the subspace $\mathrm{Ker}(\mathbf{d})$

no amplification of highly oscillatory functions occurs, which renders point smoothing ineffective. On the other part, on the L^2-orthogonal complement $\mathrm{Ker}(\mathbf{d})^\perp$ the $(\mathbf{d}\cdot, \mathbf{d}\cdot)$ part prevails. This can be taken into account in terms of the $L^2(\Omega)$-orthogonal Helmholtz decomposition

$$\mathbf{H}(\mathbf{d}, \Omega) = \mathrm{Ker}(\mathbf{d}) \oplus \mathrm{Ker}(\mathbf{d})^\perp .$$

This decomposition is useful for actual computations, because discrete potentials furnish a representation of $\mathrm{Ker}(\mathbf{d})$ by means of localized basis functions. This paves the way for enhancing the standard smoother by additional smoothing steps in potential space. The resulting scheme is called hybrid smoothing.

Let us illustrate the considerations in the case of 1-forms in three dimensions. In this case $\mathrm{Ker}(\mathbf{curl})$ is provided by the space of gradients. Hence, the vector-fields in its complement are divergence free. Using norm equivalences on convex domains [2], we can loosely state

$$A_1 \approx \mathrm{Id} + \Delta \text{ on } \mathrm{Ker}(\mathbf{curl})^\perp .$$

On the other hand, using the representation $\mathrm{Ker}(\mathbf{curl}) = \mathbf{grad}\, H_0^1(\Omega)$ directly, we can formulate the equivalence

$$a(\cdot,\cdot)_{|\mathrm{Ker}(\mathbf{curl})} \Longleftrightarrow (\mathbf{grad}\,\cdot, \mathbf{grad}\,\cdot)_{L^2(\Omega)} ,$$

with the right hand side living in $H_0^1(\Omega)$. In short, we can write

$$\mathbf{grad}^T \circ A_1 \circ \mathbf{grad} = \Delta \text{ in } H_0^1(\Omega) .$$

This offers a clue how smoothing in potential space can be realized: after lifting into potential space, we can resort to fast standard multigrid for the Laplacian.

In the general case of l-forms, the hybrid smoother is a combination between the classical smoothing (Gauss-Seidel) for l-forms and the system matrix A_l, and point smoothing for $(l-1)$-forms and the corresponding stiffness matrix. In figure 7.5 we give a flow-chart for one step of hybrid smoothing step, where D is the matrix of the exterior derivative.

$$\begin{tikzcd}
 & & & \vec{r} = D^t\vec{\delta} \arrow[r, "\mathrm{GS}(S_{l-1})"] & \vec{c} \arrow[dr, "D"] & & \\
(\vec{\zeta},\vec{\sigma}) \arrow[r, "\mathrm{GS}(A_l)"] & \vec{\zeta} \arrow[r] \arrow[rrrr, bend right] & \vec{\delta} = \vec{\sigma} - A_l\vec{\zeta} \arrow[ur, "D^t"] & & & \vec{\tau} = D\vec{c} \arrow[r] & \vec{\zeta} = \vec{\zeta} + \vec{\tau}
\end{tikzcd}$$

Figure 7.5: Hybrid Smoothing Step

The algorithm for the hybrid smoothing of l-forms $(1 \leq l \leq d-1)$ is described in Fig. 7.6, where S_{l-1} is the stiffness matrix for $(l-1)$-forms and $A_l = S_l + M_l$ is the system matrix for l-forms.

```
Algorithm HS($\vec{\zeta} \in \mathcal{A}$, $\vec{\sigma} \in \mathcal{A}$)

  {
    Gauß-Seidel sweep on $A_l\vec{\zeta} = \vec{\sigma}$
    $\vec{\delta} = \vec{\sigma} - A_l\vec{\zeta}$
    $\vec{r} = D^T\vec{\delta}$
    Gauß-Seidel sweep on $S_{l-1}\vec{c} = \vec{r}$
    $\vec{\tau} = D\vec{c}$
    $\vec{\zeta} = \vec{\zeta} + \vec{\tau}$
    return $\vec{\zeta}$
  }
```

Figure 7.6: Hybrid smoother

Beside a smoother, the V-cycle comprises transfer operators between different levels of semicoarsened anisotropic full grids. The elements of the restriction matrix are the degrees of freedom on faces belonging to the fine mesh, computed for basis functions defined on the coarse mesh. In the one dimensional case at level n, we have for the box functions

$$B_{nk}(t) = \frac{1}{2}B_{n+1\,2k-1}(t) + \frac{1}{2}B_{n+12k}(t)\,.$$

Hence, the degrees of freedom on faces belonging to the fine mesh are

$$\int\limits_{2^{-n}(k-1)}^{2^{-n}(k-2^{-1})} B_{nk}(t)dt = \frac{1}{2}\,, \qquad \int\limits_{2^{-n}(k-2^{-1})}^{2^{-n}k} B_{nk}(t)dt = \frac{1}{2}\,,$$

whereas for the hat functions we have

$$\varphi_{nk}(t) = \frac{1}{2}\varphi_{n+1\,2k-1}(t) + \varphi_{n+1\,2k}(t) + \frac{1}{2}\varphi_{n+1\,2k+1}(t)\,.$$

Remark first that the transfers do not mix components. Then, we distinguish two cases for the I-component of the form. If the direction of restriction j belongs to I, then the faces corresponding to the significant degrees of freedom are parallel to the direction j. In this case, we employ the stencil $[\frac{1}{2}, \frac{1}{2}]$, given by the box functions. If j is not in I, then the faces are orthogonal to the direction j, and the hat functions are now relevant.

Denote the index corresponding to the restriction in the direction j by $\boldsymbol{\nu}'$, that is $\boldsymbol{\nu}'_i = \boldsymbol{\nu}_i$ for $i \neq j$ and $\boldsymbol{\nu}'_j = \boldsymbol{\nu}_j - 1$. Note that in the case of the nodal representation on anisotropic full grids, the data structure is simple. It involves merely arrays. For analogy with the sparse grid case, suppose that we access the value corresponding to a point (fixed by $\mathbf{k}$, with $0 \leq k_i \leq 2^{n_i}$ for $0 \leq i \leq d$) and an index I over the function `access(`I`,k)`. We refer to Fig. 7.8 for the restriction and to Fig 7.7 for the prolongation algorithm.

We are now able to describe in Fig. 7.9 the algorithm for the V-cycle by semicoarsening in direction j for l-forms on anisotropic full grids in the nodal representation starting from level $\boldsymbol{\nu}$.

7.2 Stencils on Anisotropic Full Grids

A core component of the multigrid method presented in the previous section is point smoothers for the nodal discretization of $a(\cdot, \cdot)$ on anisotropic full grids. To that end we need information about the mass and the stiffness matrices for different resolutions $\mathbf{n}$ and dimensions d.

Consider the nodal representation of an l-form on an anisotropic full grid $\Omega_{\mathbf{n}}$. As the supports of the basis functions are small, the mass and the stiffness matrices are sparse. Thus, since the bilinear form has constant coefficients, the matrices can be expressed through difference stencils (for the case of 0-forms see e.g. [5, 77]). For fixed dimension d and resolution $\mathbf{n}$, we can analytically compute the *constant* stencil entries and store them as constants. But the preceding algorithm requires the evaluation of stencils for a whole range of different resolutions $\mathbf{n}$ and dimensions d. Hence, we decided to compute the stencil entries numerically, having d and $\mathbf{n}$ as input data.

First, let us clarify the idea of a stencil in the case of a general l-form. In fact, we associate *a stencil to each component* of the differential form.

```
**prolongate**(const $\vec{\tau} \in \mathcal{A}_{\boldsymbol{\nu}'}$, $\vec{\delta} \in \mathcal{A}_{\boldsymbol{\nu}}$, $j \in \{1, \ldots, d\}$)

  {
    for each $\mathbf{k}$ with ($0 \leq k_i < 2^{\nu_i}$, $i = 1, \ldots, d$) and for each
    index $I$

        if ($j \in I$)
            if ($k_j \% 2 == 0$)
              {
                point $\mathbf{p} = \mathbf{k}$
                $p_j = k_j/2$
                $\vec{\delta}$.access($\mathbf{k}$,$I$) = $\vec{\tau}$.access($\mathbf{p}$,$I$)
              }
            else
              {
                point $\mathbf{q} = \mathbf{p} = \mathbf{k}$
                $p_j = (k_j - 1)/2$, $q_j = (k_j + 1)/2$
                $\vec{\delta}$.access($\mathbf{k}$,$I$) = 0.5*$\vec{\tau}$.access($\mathbf{p}$,$I$)
                $\vec{\delta}$.access($\mathbf{k}$,$I$) += 0.5*$\vec{\tau}$.access($\mathbf{q}$,$I$)
              }
        else
            if ($k_j \% 2 == 0$)
              {
                point $\mathbf{p} = \mathbf{k}$
                $p_j = k_j/2$
                $\vec{\delta}$.access($\mathbf{k}$,$I$) = 0.5*$\vec{\tau}$.access($\mathbf{p}$,$I$)
              }
            else
              {
                point $\mathbf{p} = \mathbf{k}$
                $p_j = (k_j + 1)/2$
                $\vec{\delta}$.access($\mathbf{k}$,$I$) = 0.5*$\vec{\tau}$.access($\mathbf{p}$,$I$)
              }
  }
```

Figure 7.7: Prolongation operator

```
restrict(const $\vec{\delta} \in \mathcal{A}_{\boldsymbol{\nu}}$, $\vec{\tau} \in \mathcal{A}_{\boldsymbol{\nu}'}$, $j \in \{1, \ldots, d\}$)

  {
    for each $\mathbf{k}$ with ($0 \leq k_i < 2^{\nu'_i}$, $i = 1, \ldots, d$) and
    for each index $I$

        if ($j \in I$)
          {
            point $\mathbf{r} = \mathbf{q} = \mathbf{p} = \mathbf{k}$
            $p_j = 2k_j$, $q_j = 2k_j - 1$, $r_j = 2k_j + 1$
            $\vec{\tau}$.access($\mathbf{k}$,$I$) = $0.5$*$\vec{\delta}$.access($\mathbf{q}$,$I$)
            $\vec{\tau}$.access($\mathbf{k}$,$I$) += $0.5$*$\vec{\delta}$.access($\mathbf{r}$,$I$)
            $\vec{\tau}$.access($\mathbf{k}$,$I$) += $\vec{\delta}$.access($\mathbf{p}$,$I$)
          }
        else
          {
            point $\mathbf{q} = \mathbf{p} = \mathbf{k}$
            $p_j = 2k_j$, $q_j = 2k_j - 1$
            $\vec{\tau}$.access($\mathbf{k}$,$I$) = $0.5$*$\vec{\delta}$.access($\mathbf{q}$,$I$)
            $\vec{\tau}$.access($\mathbf{k}$,$I$) += $0.5$*$\vec{\delta}$.access($\mathbf{p}$,$I$)
          }
  }
```

Figure 7.8: Restriction operator

```
Algorithm MGV($\nu_j \geq 1$, $\vec{\zeta} \in \mathcal{A}_{\boldsymbol{\nu}}$, $\vec{\sigma} \in \mathcal{A}_{\boldsymbol{\nu}}$)

  {
    HS($\vec{\zeta}, \vec{\sigma}$)
    $\vec{\delta} = \vec{\sigma} - A_l\vec{\zeta}$
    restrict($\vec{\delta}$, $\vec{\tau}$, $j$)
    if ($\nu_j == 1$) solve $A_l\vec{\gamma} = \vec{\tau}$
    else MGV($\nu_j - 1$, $\vec{\gamma}$, $\vec{\tau}$)
    prolongate($\vec{\gamma}$, $\vec{\epsilon}$, $j$)
    $\vec{\zeta} = \vec{\zeta} + \vec{\epsilon}$
    HS($\vec{\zeta}, \vec{\sigma}$)
  }
```

Figure 7.9: Semicoarsening with hybrid smoother

Consider the basis form associated to the face

$$F_{\mathbf{k}I} = [x_{\mathbf{nk}}; \vec{e}_{i_1}, \vec{e}_{i_2}, \dots, \vec{e}_{i_l}] \,,$$

with $0 \le k_i < 2^{n_i}$, $i = 1, \dots, d$

$$\eta^I_{\mathbf{k}}(\mathbf{x}) = \Phi^I_{\mathbf{k}}(\mathbf{x}) dx_I = \prod_{i \in I} B_{k_i}(x_i) \prod_{j \notin I} \varphi_{k_j}(x_j) dx_I \,.$$

Here, for simplicity, we dropped out the indices for the fixed level $\mathbf{n}$. Then, consider all the d-cubes in the anisotropic full mesh which have $F_{\mathbf{k}I}$ as face. We call each l-face belonging to those cubes a neighbor of $F_{\mathbf{k}I}$. Because of the small support of $\Phi^I_{\mathbf{k}}$, each component of the stencil is exactly determined by the interaction between $\eta^I_{\mathbf{k}}$ and the basis forms living on neighbors of the face $F_{\mathbf{k}I}$. Hence, each component of the stencil can be represented on a small anisotropic full grid of resolution $\boldsymbol{\delta}$, with $\delta_j = 1$ for $j \notin I$ and $\delta_i = 0$ for $i \in I$.

In the particular case of the mass matrix, the definition of the $L^2(\Omega)$ scalar product gives

$$\left(\Phi^I_{\mathbf{k}}(\mathbf{x}) dx_I, \Phi^J_{\mathbf{r}}(\mathbf{x}) dx_J\right)_{L^2(\Omega)} = 0 \text{ for } I \neq J \,.$$

For the faces parallel to $F_{\mathbf{k}I}$, the normed box functions yield

$$\left(\Phi^I_{\mathbf{k}}(\mathbf{x}) dx_I, \Phi^I_{\mathbf{r}}(\mathbf{x}) dx_I\right)_{L^2(\Omega)} = \prod_{i \in I} 2^{n_i} \delta_{k_i r_i} \prod_{j \notin I} 2^{-n_j} \theta_{k_j r_j} \,,$$

where the stencils δ and θ are

$$\delta_{k_i r_i} = \begin{cases} 1, & \text{if } k_i = r_i \\ 0, & \text{otherwise} \end{cases} \,, \text{ and } \theta_{k_j r_j} = \begin{cases} 1/6, & \text{if } k_j = r_j \pm 1 \\ 2/3, & \text{if } k_j = r_j \\ 0, & \text{otherwise.} \end{cases}$$

In Fig. 7.10, we present the stencil for the mass matrix in the case of 1-forms in two dimensions.

In the case of the stencil for the stiffness matrix, we use the same idea as for the evaluation of $S_l \vec{\omega}$. For the I-component of the stencil, consider the corresponding small anisotropic full grid of resolution $\boldsymbol{\delta}$. On this grid, construct an Whitney l-form η^I having all degrees of freedom 0, excepting the interior one (corresponding to I), which is 1. Employ the discrete exterior differential and the mass matrix multiplication:

$$\vec{\xi^I} = D^T M_{l+1} D \vec{\eta}^I \,.$$

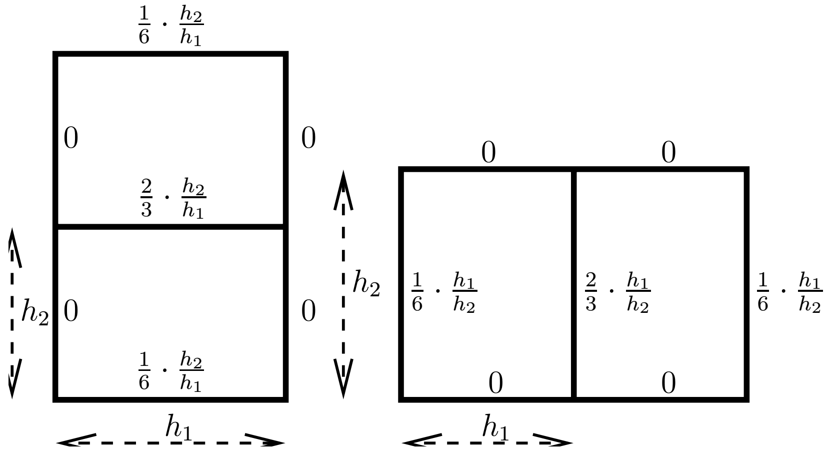

$\frac{1}{6} \cdot \frac{h_2}{h_1}$
0
0
$\frac{2}{3} \cdot \frac{h_2}{h_1}$
h_2
0
0
$\frac{1}{6} \cdot \frac{h_2}{h_1}$
h_1
0
0
h_2
$\frac{1}{6} \cdot \frac{h_1}{h_2}$
$\frac{2}{3} \cdot \frac{h_1}{h_2}$
$\frac{1}{6} \cdot \frac{h_1}{h_2}$
0
0
h_1

Figure 7.10: Example of stencil for mass matrix: $d = 2$, $l = 1$

The implementation of the discrete exterior differential operator is simple in the case of nodal representation on anisotropic full grids, see [13]. As the derivative of the hat function is the Haar-wavelet, the incidence relations (see Fig. 7.11) between l- and $(l+1)$-faces determine the algorithm. Then the pseudocode algorithm for the computation of $\vec{\eta} = D\vec{\omega}$ in the nodal representation on an anisotropic full grid of fixed resolution $\mathbf{n}$ is presented in Fig. 7.12.

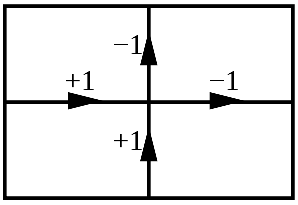


Figure 7.11: Incidence stencil for $l = 0$

7.3 Complexity and Numerical Results

7.3.1 Complexity

Regarding the complexity of the multigrid solver, we face the same situation as for the Q-cycle for Lagrangian finite elements. The computational cost is dominated by the multiplication with the system matrix in the *update* step.

```
for each point $\mathbf{k}$ and each index $I$
    for each $t \notin I$ do {
        sgn = card$\{i \in I;\ i < t\}$;   $s := (-1)^{\mathtt{sgn}}$; $J =$
        $I \cup \{t\}$
        $\vec{\eta}$.access($J$,$\mathbf{k}$) += $s$ * $\vec{\omega}$.access($I$,$\mathbf{k}$)
        point $\mathbf{k}' = \mathbf{k}$; $k'_t = k_t + 1$
        $\vec{\eta}$.access($J$,$\mathbf{k}'$) -= $s$ * $\vec{\omega}$.access($I$,$\mathbf{k}$)
    }
```

Figure 7.12: Exterior derivative operator

For fixed dimension d and order l of the differential form, denote by $N_n^{d,l}$ the number of degrees of freedom for a sparse grid discretization at resolution n. As the multiplication with the system matrix needs $\mathcal{O}(N_n^{d,l})$ operations, the complexity of a multigrid step is then

$$\mathcal{O}\left(\binom{n+2d-2}{d-1} N_n^{d,l}\right).$$

Denote the total number of the sparse grid points by $b_{d,n}$ and the number of the interior points by $a_{d,n}$. Bungartz showed in [23] that

$$a_{d,n} = 2^n \left[\frac{n^{d-1}}{(d-1)!} + \mathcal{O}\left(n^{d-2}\right)\right]$$

and

$$\frac{a_{d,n}}{b_{d,n}} \to 1\,, \text{ for } n \to \infty \text{ and fixed } d.$$

As $N_n^{d,l} \le \binom{d}{l} b_{n,d}$, we conclude that for d and l fixed, the computational complexity of the multigrid step is given by

$$\binom{d}{l} 2^n \left[\frac{n^{2d-2}}{((d-1)!)^2} + \mathcal{O}\left(n^{2d-3}\right)\right].$$

Denoting $h_n := 2^{-n}$, this is

$$\mathcal{O}\left(h_n^{-1} |\log h_n|^{d-1} \cdot |\log h_n|^{d-1}\right),$$

whereas the number of degrees of freedom is

$$\mathcal{O}\left(h_n^{-1}|\log h_n|^{d-1}\right) .$$

As on the case of the Lagrangian finite elements on sparse grids, the problem of finding the multigrid algorithm with optimal computational complexity is still open [60, 64, 67].

7.3.2 Numerical results

In the numerical experiments, Q-cycles with one pre- and one post-(hybrid) smoothing step were applied to the variational problem (5.1) with zero right hand side for $d = 2, 3, 4$ and $0 \leq l \leq d-1$ with exact solution 0. The initial guess had all the interior degrees of freedom set to 1 and the rate of convergence was determined from the reduction of the discrete counterpart of the $\mathbf{H}(\mathbf{d}, \Omega)$-seminorm of the error in the final of 10 multigrid iteration sweeps. The measured convergence rates are listed in Tables 7.1 to 7.3.

n	l=0	l=1	l=2	l=3
4	1.1e-05	4.4e-05	0.00011	0.00014
5	0.00028	0.00026	0.00012	0.00016
6	0.00059	0.00056	0.00017	2.2e-05
7	0.0021	0.0033	0.00056	0.0003
8	0.0052	0.0063	0.0011	0.00049

n	l=0	l=1	l=2	l=3
4	0.0066	0.0098	0.011	0.026
5	0.017	0.0068	0.012	0.026
6	0.023	0.013	0.013	0.027
7	0.052	0.019	0.014	0.028
8	0.089	0.028	0.014	0.028

Table 7.1: Convergence rates (d=4) for Q-cycle (top) and S-cycle (bottom)

In general we observe very fast convergence. The rates on coarser meshes are so close to zero that the measurements are rather sensitive to the choice of the initial vector and round-off. This accounts for the somewhat erratic behavior of the recorded rates. Nevertheless, even for very fine resolutions the rates do not deteriorate significantly. This suggests that the multigrid methods may, indeed, be asymptotically optimal, that is, its rate of convergence is bounded away from 1 even as $n \to \infty$.

n	l=0	l=1
2	0.004	0.0049
3	0.0043	0.0037
4	0.0028	0.0044
5	0.005	0.0047
6	0.002	0.0064
7	0.015	0.014
8	0.0068	0.006
9	0.017	0.017
10	0.0064	0.0072
11	0.018	0.018
12	0.0073	0.0087
13	0.018	0.018
14	0.0079	0.0095
15	0.018	0.018
16	0.0084	0.0097
17	0.018	0.018

n	l=0	l=1
2	0.00048	0.0023
3	0.0013	0.0039
4	0.0041	0.0039
5	0.00041	0.0038
6	0.003	0.004
7	0.0037	0.0041
8	0.008	0.0066
9	0.011	0.01
10	0.012	0.012
11	0.015	0.014
12	0.017	0.016
13	0.017	0.016
14	0.018	0.017
15	0.019	0.017
16	0.019	0.017
17	0.019	0.017

Table 7.2: Convergence rates (d=2) for the Q-cycle (left) and S-cycle (right)

n	l=0	l=1	l=2
3	0.00011	0.00034	0.00036
4	0.00018	0.00051	0.00053
5	0.00024	0.00077	0.00087
6	0.0011	0.0011	0.00032
7	0.0031	0.003	0.00071
8	0.0056	0.0055	0.0014
9	0.011	0.011	0.0017
10	0.026	0.025	0.0018
11	0.033	0.032	0.0019
12	0.035	0.035	0.0019

n	l=0	l=1	l=2
3	0.0024	0.0025	0.013
4	0.008	0.0099	0.013
5	0.0033	0.0059	0.013
6	0.011	0.013	0.014
7	0.019	0.014	0.014
8	0.028	0.0096	0.015
9	0.05	0.014	0.015
10	0.074	0.051	0.015
11	0.088	0.038	0.015
12	0.098	0.023	0.015

Table 7.3: Convergence rates (d=3) for the Q-cycle (left) and S-cycle (right)

Chapter 8

Conclusion

> The problem of creating something which is new, but consistent with everything which has been seen before, is one of extreme difficulty.
>
> **R.P. Feynman**

This work is a breakthrough of the sparse grid techniques in a special field. So far, sparse grids schemes have been based on linear and higher order Lagrange polynomials for H^1 conforming finite elements. The current work generalizes the discretization on sparse grids to discrete differential forms. The extension to general l-forms in d dimensions includes the well known Whitney elements.

The construction is based on one-dimensional differential forms, related wavelet representations and their tensor products. In addition to the construction of spaces, interpolation estimates are given. They display the typpical efficiency of approximations based on sparse grids.

There are aspects of discrete differential forms not paralleled in the theory of Lagrangian finite elements, as the commuting diagram property or the existence of discrete potentials. Another pronounced difference resides in the interpolation procedure, which needs imperatively an approximation step. On the contrary, the classical interpolation by Lagrangian elements on sparse grids is done using directly function values. A computationally cheap and sufficiently accurate approximate interpolation operator is presented.

As core results, discrete inf-sup conditions are shown theoreticaly and experimentaly for mixed second order problems. The existence of stable sparse potentials is a sufficient condition. The proof is made in various in particular cases, completely covering the three dimensional case. Numerical results evidenced stability for $d = 4$, too. The proof of the existence of stable discrete potentials in the most general formulation could not be achieved by the generality of the exterior derivative operator. The results show that the discrete differential forms give rise to viable numerical schemes for the discretization of both $H(\mathbf{d}, \Omega)$-elliptic variational problems and second order mixed problems.

Particular attention was given to the explanation of the involved algorithms, filling a gap in the literature. Details on the multilevel transforms, approximate interpolation operators, mass and stiffness matrix multiplications are given. The construction of general stencils on anisotropic full grids completes the detailed description of the multigrid solver based on semicoarsening.

Some inherent limitations of the proposed method are given by the fact that only tensor product domains have been successfully investigated. Further, no techniques for boundary value problems with variable coefficients are available yet. As for Lagrangian finite elements on sparse grids, the problem of finding the multigrid algorithm with optimal computational complexity is still open.

In the light of the remaining open questions, the foundation is laid for further investigations, for instance on Whitney-forms on adaptive sparse grid and on applications to problems set in $\boldsymbol{H}(\text{div}; \Omega)$ and $\boldsymbol{H}(\mathbf{curl}; \Omega)$. I strongly believe that this work will foster a new branch of sparse grid research into mixed discretization or even Maxwell's equations.

L'avenir est un lieu commode pour y mettre des songes[1].
Anatole France *Les Opinions de Jérome Coignard*

[1] The future is a comfortable place for placing the dreams.

Bibliography

[1] R. Albanese and G. Rubinacci, *Analysis of three dimensional electromagnetic fileds using edge elements*, J. Comp. Phys., 108 (1993), pp. 236–245.

[2] C. Amrouche, C. Bernardi, M. Dauge, and V. Girault, *Vector potentials in three–dimensional nonsmooth domains*, Math. Meth. Appl. Sci., 21 (1998), pp. 823–864.

[3] D. Arnold, R. Falk, and R. Winther, *Preconditioning in $H(\mathrm{div})$ and applications*, Math. Comp., 66 (1997), pp. 957–984.

[4] ——, *Multigrid in $H(\mathrm{div})$ and $H(\mathbf{curl})$*, Numer. Math., 85 (2000), pp. 175–195.

[5] O. Axelson and V. Barker, *Finite element solution of boundary value problems*, Academic Press, Orlando, 1984.

[6] R. Balder, *Adaptive Verfahren für eliptische und parabolische Differentialgleichungen auf dünnen Gittern*, PhD thesis, Institut für Informatik, TU München, 1994.

[7] R. Balder, U. Rüde, S. Schneider, and C. Zenger, *Sparse grid and extrapolation methods for parabolic problems.*, in International Conference on Computational Methods in Water Resources X, A. P. et al., ed., Kluwer Academic Publishers Group, Dordrecht, 1994, pp. 1383–1392.

[8] R. Balder and C. Zenger, *The solution of multidimensional real Helmholtz equations on sparse grids*, SIAM J. Sci. Comp., 17 (1996), pp. 631–646.

[9] D. Boffi, P. Fernandes, L. Gastaldi, and I. Perugia, *Computational models of electromagnetic resonators: Analysis of edge element approximation*, SIAM J. Numer. Anal., 36 (1999), pp. 1264–1290.

[10] A. Bossavit, *A rationale for edge elements in 3D field computations*, IEEE Trans. Mag., 24 (1988), pp. 74–79.

[11] ——, *A new viewpoint on mixed elements*, Meccanica, 27 (1992), pp. 3–11.

[12] ——, *Électromagnétisme, en vue de la modélisation*, Springer–Verlag, Paris, 1993.

[13] ——, *Computational Electromagnetism. Variational Formulation, Complementarity, Edge Elements*, vol. 2 of Electromagnetism Series, Academic Press, San Diego, CA, 1998.

[14] ——, *On the geometry of electromagnetism IV: "Maxwell's house"*, J. Japan Soc. Appl. Electromagntics & Mech., 6 (1998), pp. 318–326.

[15] ——, *Generalized finite differences in computational electromagnetics*, in Geometric Methods for Computational Electromagnetics, F. Teixeira, ed., vol. 32 of PIER, EMW Publishing, Cambridge, MA, 2001, pp. 45–64.

[16] D. Braess, *Finite Elements: Theory, Fast Solvers and Applications in Solid Mechanics.*, Cambridge University Press, Cambridge, 1997.

[17] S. Brenner and R. Scott, *Mathematical theory of finite element methods*, Texts in Applied Mathematics, Springer–Verlag, New York, 1994.

[18] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer, 1991.

[19] H. Bungartz, *Rekursive Verfahren und hierarchische Datenstrukturen in der numerischen analysis.* Lecture notes, TU München.

[20] ——, *An adaptive Poisson solver using hierarchical bases and sparse grids*, in Iterative Methods in Linear Algebra: Proceedings of the IMACS International Symposium, Brussels, 2.-4. 4. 1991, R. B. P. de Groen, ed., Elsevier, Amsterdam, 1992, pp. 293–310.

[21] ——, *Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung*, PhD thesis, Institut für Informatik, TU München, 1992.

[22] ——, *A multigrid algorithm for higher order finite elements on sparse grids*, ETNA, (1997), pp. 63–77.

[23] ——, *Finite Elements of Higher Order on Sparse Grids*, Shaker Verlag, Aachen, 1998.

[24] H. Cartan, *Formes Différentielles*, Hermann, Paris, 1967.

[25] P. Ciarlet, *The Finite Element Method for Elliptic Problems*, vol. 4 of Studies in Mathematics and its Applications, North-Holland, Amsterdam, 1978.

[26] W. Dahmen, *Wavelet and multiscale methods for operator equations*, Acta Numerica, (1997), pp. 55–228.

[27] W. Dahmen, A. Kunoth, and K. Urban, *Biorthogonal spline-wavelets on the interval - stability and moment conditions*, IGPM-Report 129, RWTH Aachen, 1996.

[28] ——, *A wavelet Galerkin method for the Stokes equation*, Computing, 56 (1996), pp. 259–301.

[29] T. Dornseifer, *Diskretisierung allgemeiner elliptischer Differentialgleichungen in krummlinigen Koordinatensystemen auf dünnen Gittern*, PhD thesis, Institut für Informatik, Technische Universität München, München, Germany, 1997.

[30] T. Dornseifer and C. Pflaum, *Elliptic differential equations on curvilinear bounded domains with sparse grids*, Computing, 56 (1996), pp. 197–213.

[31] V. Girault and P. Raviart, *Finite element methods for Navier–Stokes equations*, Springer, Berlin, 1986.

[32] V. Gradinaru and R. Hiptmair, *Whitney elements on pyramids*, Electron. Trans. Numer. Anal., 8 (1999), pp. 154–168.

[33] ——, *Whitney forms on sparse grids*, Tech. Rep. 153, SFB 382, Universität Tübingen, Tübingen, April 2000.

[34] ——, *Mixed finite elements on sparse grids*, Numer. Math., ?? (2002), p. ?? DOI 10.1007/s002110100382.

[35] ——, *Multigrid for discrete differential forms on sparse grids*, Computing, (2002, to appear). To appear.

[36] M. Griebel, *Parallel multigrid methods on sparse grids*, in Multigrid Methods III, W. Hackbusch and U. Trottenberg, eds., vol. 98 of ISNM, Basel, 1991, Birkhäuser, pp. 211–221.

[37] ——, *Multilevelmethoden als Iterationsverfahren über Erzeugendensystemen,*, B.G. Teubner, Stuttgart, 1994.

[38] ——, *Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences*, Computing, 61 (1998), pp. 151–179.

[39] M. Griebel and P. Oswald, *On additive Schwarz preconditioners for sparse grid discretization*, Numer. Math., 66 (1992), pp. 449–464.

[40] ——, *Tensor product type subspace splittings and multilevel iterative methods for aisotropic problems*, Advances in Computational Mathematics, (1995), pp. 171–206.

[41] M. Griebel, P. Oswald, and T. Schiekofer, *Sparse grids for boundary integral equations*, Tech. Rep. 554, SFB 256, Universität Bonn, 1998.

[42] M. Griebel, M. Schneider, and C. Zenger, *A combination technique for the solution of sparse grid problems*, in Iterative Methods in Linear Algebra, P. de Groen and R. Beauwens, eds., IMACS, Elsevier, North Holland, 1992, pp. 263–281.

[43] M. Griebel and G. Zumbusch, *Parallel multigrid in an adaptive PDE solver based on hashing and space–filling curves*, Parallel Computing, 25 (1999), pp. 827–843.

[44] R. Hiptmair, *Multilevel Preconditioning for Mixed Problems in Three Dimensions*, vol. 8 of Augsburger mathematisch–naturwissenschaftliche Schriften, Wißner, Augsburg, 1996.

[45] ——, *Multigrid method for $H(\mathrm{div})$ in three dimensions*, ETNA, 6 (1997), pp. 133–152.

[46] ——, *Canonical construction of finite elements*, Math. Comp., 68 (1999), pp. 1325–1346.

[47] ——, *Multigrid method for Maxwell's equations*, SIAM J. Numer. Anal., 36 (1999), pp. 204–225.

[48] ——, *Finite elements in computational electromagnetism*, Acta Numerica, (2002). To appear.

[49] R. Hiptmair and R. Hoppe, *Multilevel preconditioning for mixed problems in three dimensions*, Numer. Math., 82 (1999), pp. 253–279.

[50] R. Hiptmair and K. Neymeyr, *Multilevel method for mixed eigenproblems*, Report 159, SFB 382, Universität Tübingen, Tübingen, Germany, 2001. To appear in SIAM J. Sci. Comp.

[51] M. Hopf and T. Ertl, *Parallelizing sparse grid volume visualization with implicit preview and load balancing*, tech. rep., University of Stuttgart., 2001.

[52] T. Iwaniec, *Nonlinear differential forms.*, international summer school 1998, lecture notes, University of Jyväskylä, Department of Mathematics, Jyväskylä, Finland, 1999.

[53] D. Knuth, *The Art of Computer Programming*, vol. 3rd, Addison-Wesley, 2nd ed., 1973.

[54] J.-F. Lee and Z. Sacks, *Whitney elements time domain (WETD) methods*, IEEE Trans. Mag., 31 (1995), pp. 1325–1329.

[55] C. Mattiussi, *An analysis of finite volume, finite element, and finite difference methods using some concepts from algebraic topology*, J. Comp. Phys., 9 (1997), pp. 295–319.

[56] J. Maxwell, *A Treatise on Electricity and Magnetism*, Oxford Univsersity Press reprint, 3rd ed., 1998.

[57] J. Metzger, *Expression templates for efficient C++ operators*, studienarbeit informatik, Math. Inst., Universität Tübingen, Tübingen Germany, January 2001.

[58] P. Monk, *A mixed method for approximating Maxwell's equations*, SIAM J. Numer. Anal., 28 (1991), pp. 1610–1634.

[59] J. Nédélec, *Mixed finite elements in* R^3, Numer. Math., 35 (1980), pp. 315–341.

[60] J. Noordmans and P. W. Hemker, *Convergence results for 3D sparse grid approaches*, Numer. Linear Algebra Appl., 5 (1999), pp. 363–376.

[61] P. Oswald, *Multilevel finite element approximation*, Teubner Skripten zur Numerik, B.G. Teubner, Stuttgart, 1994.

[62] A. Pfaffinger, *Funktionale Beschreibung und Parallelisierung von Algorithmen auf dünnen Gittern*, PhD thesis, Institut für Informatik, TU München, 1997.

[63] C. Pflaum, *Convergence of the combination technique for the finite element solution of poisson's equation*, Tech. Rep. TUM–I9328, Technische Universität München, 1993.

[64] C. Pflaum, *Diskretisierung elliptischer Differentialgleichungen mit dünnen Gittern*, PhD thesis, TU München, 1996.

[65] C. Pflaum, *A multilevel algorithm for the solution of second order elliptic differential equations on sparse grids*, Numer. Math., 79 (1998), pp. 141–155.

[66] ——, *Fast and robust multilevel algorithms*, Habilitationsschrift, Mathematische Institute, Universität Würzburg, Würzburg, Germany, 1999.

[67] B. K. P.W. Hemker and J. Noordmans, *3d multigrid on partially ordered sets of grids*, in Multigrid Methods V, W. Hackbusch and G. Wittum, eds., Lecture Notes in Computational Science and Engineering 3, Berlin, 1998, Springer.

[68] P. A. Raviart and J. M. Thomas, *A Mixed Finite Element Method for Second Order Elliptic Problems*, vol. 606 of Springer Lecture Notes in Mathematics, Springer, Ney York, 1977, pp. 292–315.

[69] J. Roberts and J. Thomas, *Mixed and hybrid methods*, in Finite Element Methods, P. Ciarlet and J. Lions, eds., vol. 2 of Handbook of Numerical Analysis, North-Holland, Amsterdam, 1991, pp. 527–639.

[70] U. Rüde, H. Bungartz, and M. Griebel, *Extrapolation, combination and sparse grid techniques for elliptic boundary value problems*, in Computer Methods in Applied Mechanics and Engineering 116, 1992, pp. 243–252.

[71] T. Schiekofer, *Die Methode der Finiten Differenzen auf Dünnen Gittern zur adaptiven Multilevel-Lösung partieller Differentialgleichungen*, PhD thesis, Universität Bonn, Institut für Angewandte Mathematik, 1998.

[72] S. Schneider, *Adaptive Solution of Elliptic Partial Differential Equations by Hierarchical Tensor Product Finite Elements*, PhD thesis, Fakultät für Informatik, TU München, 2000.

[73] R. Sedgewick, *Algorithmen in C++*, Addison-Wesley, 1992.

[74] SGI, *The standard template library programmer's guide.* http://www.sgi.com/Technology/STL/, 1999.

[75] F. Sprengel, *Periodic interpolation and wavelets on sparse grids*, Numerical Algorithms, 17 (1998), pp. 147–169.

[76] G. Strang, *Wavelets and dilation equations*, SIAM Review, 31 (1989), pp. 614–627.

[77] G. Strang and G. Fix, *An Analysis of the Finite Element Method*, Prentice Hall, Eaglewood Cliffs, 1973.

[78] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesly-Cambridge Press, 1996.

[79] B. Stroustrup, *The C++ Programming Language*, Addison Wesley Longman, Reading, MA, 3rd ed., 1997.

[80] S. Subramaniam, S. Ratnajeevan, and S. Hoole, *Edge elements*, in Finite Elements, Electromagnetics and Design, S. Hoole and S. Ratnajeevan, eds., Elsevier, Amsterdam, 1995, ch. 9, pp. 342–393.

[81] C. Teitzel, R. Grosso, and T. Ertl, *Particle tracing on sparse grids*, in Proc. 9th Eurographics Workshop on Visualisation in Scientific Computing, D. Bartz, ed., 1998, pp. 132–142.

[82] C. Teitzel, M. Hopf, and T. Ertl, *Volume visualisation on sparse grids*, Computing and visualisation in science, (1999), pp. 47–59.

[83] ——, *Scientific visualisation on sparse grids*, in Proc. of Scientific Visualisation - Dagstuhl '97, H. Hagen, G. Nielson, and F. Post, eds., IEEE Computer Society Press, 2000, pp. 284–295.

[84] K. Urban, *Wavelet bases in* $\mathbf{H}(\mathrm{div})$ *and* $\mathbf{H}(\mathbf{curl})$, Math. Comp., 70 (2001), pp. 739–766.

[85] J. Webb, *Edge elements and what they can do for you*, IEEE Trans. Mag., 29 (1993), pp. 1460–1565.

[86] T. Weiland, *Die Diskretisierung der Maxwell–Gleichungen*, Phys. Bl., 42 (1986), pp. 191–201.

[87] ——, *Time domain electromagnetic field computation with finite difference methods*, Int. J. Numer. Modelling, 9 (1996), pp. 295–319.

[88] J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Review, 34 (1992), pp. 581–613.

[89] ——, *An introduction to multilevel methods*, in Wavelets, Multilevel Methods and Elliptic PDEs, M. Ainsworth, K. Levesley, M. Marletta, and W. Light, eds., Numerical Mathematics and Scientific Computation, Clarendon Press, Oxford, 1997, pp. 213–301.

[90] H. Yserentant, *Hierarchical bases*, in ICIAM 91.

[91] ——, *Old and new convergence proofs for multigrid methods*, Acta Numerica, (1993), pp. 285–326.

[92] C. Zenger, *Sparse grids*, Tech. Rep. 18–90, SFB 342, TU München, 1990.

[93] G. Zumbusch, *A sparse grid pde solver*, in Advances in Software Tools for Scientific Computing (Proceedings SciTools '98), H. P. Langtangen, A. M. Bruaset, and E. Quak, eds., no. 10 in Lecture Notes in Computational Science and Engineering, Springer, 2000, ch. 4, pp. 133–178.