Adaptive Scattered Data Fitting with Tensor Product Spline-Wavelets

Dissertation

zur Erlangung des Doktorgrades
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von
Daniel Castaño Díez
aus Vitoria-Gasteiz

Bonn, 2005

Angefertigt mit Genehmigung
der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Referentin: Professor Dr. Angela Kunoth
2. Referent: Professor Dr. Karl Scherer
Tag der Promotion: 3. Februar 2005

Diese Arbeit ist mit Unterstützung
der Deutschen Forschungsgemeinschaft
(SFB 350, SFB 256, DFG-Projekt 1028/7-1)
und der Baskischen Regierung
(BFI Programm)
entstanden

Diese Dissertation ist auf dem Hochschulschriftenserver der ULB Bonn
http://hss.ulb.uni-bonn.de/diss_online elektronisch publiziert

Abstract

The core of the work we present here is an algorithm that constructs a least squares approximation to a given set of unorganized points. The approximation is expressed as a linear combination of particular B-spline wavelets.

It implies a multiresolution setting which constructs a hierarchy of approximations to the data with increasing level of detail, proceeding from coarsest to finest scales. It allows for an efficient selection of the degrees of freedom of the problem and avoids the introduction of an artificial uniform grid. In fact, an analysis of the data can be done at each of the scales of the hierarchy, which can be used to select adaptively a set of wavelets that can represent economically the characteristics of the cloud of points in the next level of detail. The data adaption of our method is twofold, as it takes into account both horizontal distribution and vertical irregularities of data. This strategy can lead to a striking reduction of the problem complexity. Furthermore, among the possible ways to achieve a multiscale formulation, the wavelet approach shows additional advantages, based on good conditioning properties and level-wise orthogonality. We exploit these features to enhance the efficiency of iterative solution methods for the system of normal equations of the problem. The combination of multiresolution adaptivity with the numerical properties of the wavelet basis gives rise to an algorithm well suited to cope with problems requiring fast solution methods.

We illustrate this by means of numerical experiments that compare the performance of the method on various data sets working with different multi-resolution bases.

Afterwards, we use the equivalence relation between wavelets and Besov spaces to formulate the problem of data fitting with regularization. We find that the multiscale formulation allows for a flexible and efficient treatment of some aspects of this problem. Moreover, we study the problem known as robust fitting, in which the data is assumed to be corrupted by wrong measurements or *outliers*. We compare classical methods based on re-weighting of residuals to our setting in which the wavelet representation of the data computed by our algorithm is used to locate the outliers.

As a final application that couples two of the main applications of wavelets (data analysis and operator equations), we propose the use of this least squares data fitting method to evaluate the non-linear term in the wavelet-Galerkin formulation of non-linear PDE problems.

At the end of this thesis we discuss efficient implementation issues, with a special interest in the interplay between solution methods and data structures.

Contents

1 Introduction **1**
- 1.1 Data Fitting . . . 1
- 1.2 Wavelets . . . 2
- 1.3 Structure of this Work . . . 5

2 Basic Algorithm **7**
- 2.1 Choice of Λ . . . 7
- 2.2 Normal Equations . . . 8
- 2.3 Description of the Algorithm . . . 9
  - 2.3.1 Horizontal Adaptivity . . . 9
  - 2.3.2 Vertical Adaptivity . . . 11
- 2.4 Stability of the Algorithm . . . 17
  - 2.4.1 A Stable Variation . . . 17
  - 2.4.2 Randering of a Matrix . . . 20
  - 2.4.3 Stability Theorems . . . 22

3 Numerical Performance **27**
- 3.1 Conditioning . . . 27
  - 3.1.1 The Hierarchical Basis . . . 29
  - 3.1.2 The Wavelet Linear B-Spline Basis . . . 32
- 3.2 Nesting Strategy . . . 34
- 3.3 Case Study . . . 37
- 3.4 Irregular Distribution of Measurements . . . 42
- 3.5 Construction of Tensor Products . . . 47

4 Regularization **51**
- 4.1 Smoothing with Wavelets . . . 51
- 4.2 Cross Validation . . . 52
- 4.3 Multilevel GCV . . . 59
  - 4.3.1 Case Study 1: Complexity Reduction . . . 61
  - 4.3.2 Case Study 2: Scale–Localization of Noise . . . 63
- 4.4 Other Regularization Strategies . . . 65
  - 4.4.1 Cutting Off the Iterative Procedure . . . 65

4.4.2 L-Curve Method 68

5 Robust Regression **71**
5.1 Classical Robust Regularization 71
5.2 Wavelets and M-Estimators 72
5.3 Residual-free Methods 74
5.4 A Least Squares Specific Methodology 78
5.5 Basics of the Approach 81
5.6 Global Refitting Criterion 81
5.6.1 Removal of Points in Normal Equations 84
5.7 Local Criterion 86
5.8 Local Corrected Criterion 89
5.9 High Energetic Environments 91
5.9.1 Presence of Noise 91
5.9.2 Large Number of Outliers 93
5.10 Energy Criterion 96
5.11 Several Dimensions 98

6 Application to PDEs **103**
6.1 Wavelets and Differential Operator Equations 103
6.1.1 Approximation of Functions in Wavelet Bases 104
6.1.2 Adaptive Wavelet Schemes 107
6.1.3 Nonlinear Variational Problems 111
6.1.4 LS Approach 112
6.2 The Burgers' Equation 124
6.2.1 Convergence of the Restricted Viscosity Method 125
6.2.2 Smoothing of Gibbs Phenomenon 128
6.2.3 Adaptive Solution 134

7 Implementation and Data Structures **137**
7.1 Observation Matrix A 138
7.1.1 Complexity 139
7.1.2 Evaluation of Matrix Elements 139
7.2 Coefficient Matrix $A^T A$ 139
7.2.1 Construction of Λ_{x_i} 140
7.2.2 Accumulation of Point Contributions 141
7.2.3 Performance Comparison between A and $A^T A$ Methods 143

8 Conclusions and Outlook **145**

A Documentation **149**
A.1 Preface 149
A.2 Adaptive Least Squares Fitting with Wavelets 149
A.3 Program features 152

A.3.1 Call of the Program 152
A.3.2 Creation and Use of the Tree Structure. 154
A.4 Implementation Details 155
A.4.1 Class Hierarchy 155
A.4.2 Creation of the Tree 156
A.4.3 Implementation of Refinement Schemes 159

to my parents
a Cesáreo y Adoración

Chapter 1

Introduction

1.1 Data Fitting

The problem of data fitting is formulated in the following way.

Problem 1.1.1. *Given a cloud of unstructured, not coinciding points denoted by* $P = \{(x_i, z_i)\}_{i=1,\ldots,N}$, $x_i \in [0,1]^d$, $d \in \{1,2,3\}$, $z_i \in \mathbb{R}$ *for all* $i = 1,\ldots,N$ *and defined by the corresponding horizontal values* $X = \{x_i\}_{i=1,\ldots,N}$ *and vertical values* $Z = \{z_i\}_{i=1,\ldots,N}$, *one seeks a function* $f : [0,1]^d \to \mathbb{R}$ *that represents the information contained in* P. *In our approach, we construct an* f *of the form*

$$f = \sum_{\lambda \in \Lambda} d_\lambda \, \psi_\lambda \tag{1.1.1}$$

where the $\{\psi_\lambda\}_{\lambda\in\Lambda}$ *are B-spline wavelets and* Λ *stands for an appropriate set of indices, and force it to approximate* P *in a Least Squares sense, by computing the coefficients* $\{d_\lambda\}_{\lambda\in\Lambda}$ *so that*

$$\sum_{i=1}^{N} (z_i - f(x_i))^2 \tag{1.1.2}$$

attains its minimum.

The construction of surfaces approximating or interpolating both gridded and scattered data sets finds a wide range of applications in science and engineering as explained in the surveys [77], [110], [122], [133]. Different methods have been used. From the vast amount of literature, we briefly comment on the main approaches.

Methods based on the Shannon Theorem and the Irregular Sampling Theorem have a solid theoretical underpinning for the study of band limited functions (see references in [18]) and have an extensive use in signal analysis. Practical algorithms based on this theory work with bases of trigonometric functions and are not well suited for problems requiring some extent of local adaptivity, see [70], [71], [129].

More popular are spline based methods. The application of splines in surface and curve reconstruction has been studied early, [2], [57], [93]. Some extensions

of this technique envisage the enhancement of the efficiency by some kind of data adaptivity. For instance [62] and [134] use different kinds of free-knot approaches that try to place the splines in convenient positions. A closer connection to our work are spline methods that operate on a multiscale orientation on structured grids. [76] presents a first coarse-to-fine strategy with hierarchical splines, suited for gridded, parameterized data. In [111], scattered functional data is increasingly approximated by a hierarchy of grids, an idea extended in [113] by the use of nonuniform grids. A higher adaptivity to data by means of local refinement is attained in [108] for functional, scattered data and in [83] for topology-constrained 3D data. This direction is also adopted in [126], where the local adaptive refinement of the grid is used to generate an appropriate triangulation.

This last idea is used aside from the spline formulation in [130] or in [146], where a binary tree structure allows for a coarse-to-fine, data-driven growth of the triangulation. The opposite approach of starting on a fine level of the triangulation and progressively simplify the mesh is also used, for instance, in [84] with gridded data, or in [12] with an unstructured grid. This kind of procedure is frequently used in terrain modeling related approaches, see [80].

The method of Radial Basis Functions provides good approximating properties, [148], a rich theoretical characterization ([16], [127]) and a mesh free approach, which appears quite natural when working with unorganized data, [24]. However, the data reduction strategies (like thinning [73] or adaptive thinning [66]) that are necessary to construct a multiscale formulation of the problem, as in [74] or [99], or simply to make it tractable, usually enforce the construction of some auxiliary triangulation.

1.2 Wavelets

The basic elements of our approach are wavelets. The central idea of the wavelet theory is the representation and analysis of data according to simultaneous location in space *and* in the frequency domain.

This idea can be traced back to the beginnig of the last century, when Haar presented the first known wavelet system [86]. We summarize now its basis elements using current terminology: One defines a *father function*

$$\phi(x) := \begin{cases} 1 & x \in [0,1) \\ 0 & \text{otherwise} \end{cases} \tag{1.2.1}$$

whose translations and scalings

$$\phi_{j,k}(x) := 2^{j/2}\phi(2^j x - k),\ j \geq 0,\ k = 0, \ldots, 2^j - 1, \tag{1.2.2}$$

called *scaling functions*, induce a *multiresolution analysis* of $L_2(0,1)$. This means that the scale of linear spaces $\{V_j\}_{j\geq 0}$, each defined by

$$V_j := \text{span}\{\phi_{j,k} :\ k = 0, \ldots, 2^j - 1\}, \tag{1.2.3}$$

are nested, i.e.

$$V_0 \subset V_1 \subset \ldots \subset V_j \subset \ldots \subset L_2(0,1) \tag{1.2.4}$$

and their union is dense in $L_2(0,1)$. Now, one defines the so called *mother function* as the particular linear combination of scaling functions

$$\psi(x) := \frac{1}{\sqrt{2}}\phi_{1,0}(x) - \frac{1}{\sqrt{2}}\phi_{1,1}(x) \tag{1.2.5}$$

which yields $\int_0^1 \psi(x)\phi(x)dx = 1$. The translations and scalings

$$\psi_{j,k}(x) := 2^{j/2}\psi(2^j x - k),\ j \geq 0,\ k = 0, \ldots, 2^j - 1, \tag{1.2.6}$$

of the mother function are called *wavelets*. Rescaling (1.2.5), one easily sees that the spans of all the wavelets on each dyadic level,

$$W_j := \text{span}\{\psi_{j,k} : k = 0, \ldots, 2^j - 1\} \tag{1.2.7}$$

constitute orthogonal complements to the V_j defined in (1.2.3) fulfilling $V_{j+1} = V_j \oplus W_j$. We have an orthormal basis for $L_2(0,1)$: Choosing a *coarsest level* $j_0 \geq 0$, every $f \in L_2(0,1)$ has an unique wavelet expansion of the form

$$f(x) = \sum_{k=0,\ldots,2^{j_0}-1} c_{j_0,k}\phi_{j_0,k}(x) + \sum_{j \geq j_0}^{\infty} \sum_{k=0,\ldots,2^j-1} d_{j,k}\psi_{j,k}(x), \tag{1.2.8}$$

where the expansion coefficients fulfill

$$c_{j_0,k} = \int_0^1 f(x)\phi_{j_0,k}(x)dx \text{ and } d_{j,k} = \int_0^1 f(x)\psi_{j,k}(x)dx. \tag{1.2.9}$$

Although the main ideas are already present in this simple system, its poor regularity has limited its application. The real breaktrough of wavelets starts in the late 80's, as the works of Mallat ([116]), Meyer ([119],[120]) and Daubechies ([53],[54]) showed the way to construct wavelet families satisfying more demanding requirements on properties as symmetry, orthogonality, compact support, smoothness and *vanishing moments* (a wavelet ψ is said to have n vanishing moments if $\int_0^1 \psi(x)x^m dx$ is identically zero for $m = 0, \ldots, n-1$ but not for $m = n$). In particular, [54] presents a family of orthonormal wavelets in $L_2(\mathbb{R})$ with compact support and *arbitrary* regularity. In any case, regularity is here acomplished at the expense of introducing larger supports, that is, enworsening the spatial localization of wavelets. A further step was the introduction of the concept of *biorthogonality* in [38], which sacrifices orthogonality in order to construct wavelet families in $L_2(\mathbb{R})$ with a better trade off between the other desirable properties. Instead of an orthogonal family $\{\psi_{j,k}\}_{j \geq 0, k \in \mathbb{Z}}$ that provides for every $f \in L_2(\mathbb{R})$ a unique representation

$$f(x) = \sum_{j \geq 0}^{\infty} \sum_{k \in \mathbb{Z}} (f, \psi_{j,k})\psi_{j,k}(x) \tag{1.2.10}$$

with the property

$$\|f\|_{L_2} = \left(\sum_{j\geq 0} \sum_{k\in\mathbb{Z}} (f, \psi_{j,k})^2 \right)^{1/2} \tag{1.2.11}$$

where $(\cdot,\cdot)$ is the standard scalar product in $L_2(\mathbb{R})$, one has a family $\{\psi_{j,k}\}_{j\geq 0, k\in\mathbb{Z}}$ and its *dual* $\{\tilde{\psi}_{j,k}\}_{j\geq 0, k\in\mathbb{Z}}$, which meet the biorthogonality relation

$$(\psi_{j,k}, \tilde{\psi}_{j',k'}) = \delta_{j,j'}\delta_{k,k'},\ j, j' \geq 0,\ k, k' \in \mathbb{Z} \tag{1.2.12}$$

and every $f \in L_2(\mathbb{R})$ has unique expansions

$$f(x) = \sum_{j\geq 0} \sum_{k\in\mathbb{Z}} (f, \psi_{j,k})\tilde{\psi}_{j,k}(x) = \sum_{j\geq 0} \sum_{k\in\mathbb{Z}} (f, \tilde{\psi}_{j,k})\psi_{j,k}(x). \tag{1.2.13}$$

Although (1.2.11) is no longer valid, a tight relation between the function and its representation in wavelet coordinates is still ensured, as the coefficients of these expansions fulfill

$$\|f\|_{L_2} \sim \left(\sum_{j\geq 0} \sum_{k\in\mathbb{Z}} (f, \psi_{j,k})^2 \right)^{1/2} \sim \left(\sum_{j\geq 0} \sum_{k\in\mathbb{Z}} (f, \tilde{\psi}_{j,k})^2 \right)^{1/2}. \tag{1.2.14}$$

This concept has been exploited by techniques as the *lifting scheme* [137] as a particular case of the *stable completion* method (see e.g. [23], [45]), to gain flexibility in the construction of wavelets with desirable atributes. This has made possible the introduction of wavelets on the interval ([48]) and hence on general domains (see e.g. [22],[39],[50]).

From the vast amount of wavelet constructions now availabe, we have chosen one particularly adapted to our data fitting problem. We want to recall shortly its basic properties. First, we need to fix some notation. We index wavelets by some parameter λ which comprises information such as the *resolution level* or *scale* $|\lambda| := j$, the *spatial location* k, and possibly the type of wavelet in the bivariate or trivariate case. In view of the finite domain Ω, there is a coarsest level $j_0 := 1$. The infinite set of all possible indices will be denoted by $\mathbb{I}$. We will use the convention that the linear combination

$$f(x) = \sum_{\lambda\in\Lambda} d_\lambda \psi_\lambda(x) \tag{1.2.15}$$

for any set of indices $\Lambda \subseteq \mathbb{I}$ includes scaling functions at the coarsest level. That is, we will not usually employ representations of the type (1.2.8) wich made explicit the presence of a scaling function part and a wavelet part. We will employ (tensor products of) the piecewise linear boundary adapted B–spline (pre)wavelets $\{\psi_\lambda\}_{\lambda\in\mathbb{I}}$ constructed in [136] which are a special case of the (pre)wavelets on the interval constructed in [31]: In the univariate case, the scaling functions are B-splines defined on the interval $[0, 1]$. The wavelets on higher levels are linear combinations of these B-splines constructed in order to meet the following properties:

(R) they constitute a Riesz basis for $L_2(\Omega)$, i.e., their finite linear combinations are dense in $L_2(\Omega)$ and one can find two finite positive constants A, B such that the norm equivalence

$$A \sum_{\lambda \in I\!I} |d_\lambda|^2 \leq \|f\|_{L_2}^2 \leq B \sum_{\lambda \in I\!I} |d_\lambda|^2 \tag{1.2.16}$$

holds for every $f \in L_2$ of the form $f(x) = \sum_{\lambda \in I\!I} d_\lambda \psi_\lambda(x)$;

(O) they are semi–orthogonal, i.e., for $|\lambda| \neq |\mu|$ one always has

$$\int_\Omega \psi_\lambda(x)\, \psi_\mu(x)\, dx = 0; \tag{1.2.17}$$

(L) they are compactly supported and satisfy for each λ

$$\operatorname{diam}(\operatorname{supp} \psi_\lambda) \ \sim \ 2^{-|\lambda|}. \tag{1.2.18}$$

The particular construction in [136] generates among all linear boundary adapted B–spline (pre)wavelets that still satisfy (O) those wavelets with smallest support. We have normalized the wavelets such that their $L_2(\Omega)$–norm is one. Plots of these wavelets are displayed in Figures 3.1.5 and 3.1.6.

In view of the fact that finitely many data are to be approximated, one expects that only finitely many indices from $I\!I$ are essential for the representation of f. We will therefore usually work with an index set $\Lambda \subset I\!I$, $\#\Lambda < \infty$.

1.3 Structure of this Work

This monograph is divided as follows. In Chapter 2 we describe our data fitting algorithm. Chapter 3 discusses the properties of the multiscale bases and their relation to the numerical performance of our method, testing it on different model sets of scattered data. Chapter 4 is devoted to the use of smoothness constraints in the construction of surfaces or curves, emphasizing the benefits of the wavelet formulation. In Chapter 5 we present the application of our method to cope with data corrupted by wrong measurements. An application of the data fitting methodology to the efficient numerical computation of solutions for non-linear PDE problems is then demonstrated in Chapter 6. We end with a discussion about diverse issues concerning the computational implementation of the method in Chapter 7.

Chapter 2

Basic Algorithm

2.1 Choice of Λ

Determining for Problem 1.1.1 an appropriate index set Λ for a given P is a matter of central importance. On the one hand Λ should be large enough to be able to represent the features of P. On the other hand too large a Λ is not desirable as we risk to *overfit* the cloud of points, and to process *redundant information* leading to increasing the computation time unnecessarily.

To explain the phenomenon of overfitting, consider the following example. When constructing an approximation to the points on the left of the Figure 2.1.1, we certainly prefer a function like in the central plot to one like in the right plot. The undesirable oscillations in this plot are due to the fact that in the Λ used some wavelets have supports containing very few data points and can oscillate uncontrolled in the space between them.

As for processing redundant information, consider the set of points P given on the left of Figure 2.1.2. Although we have a large number of measurements (10000 points), the vertical values show a very simple structure. The function on the right of the figure offers a good reconstruction, but the number of degrees of freedom we used (8614 wavelets) is clearly unnecessarily high in view of the reconstruction in the central plot, which with only 32 degrees of freedom can successfully reproduce

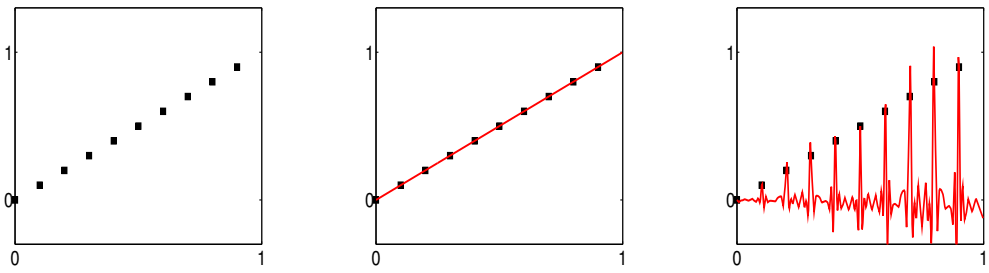


Figure 2.1.1: Overfitting.

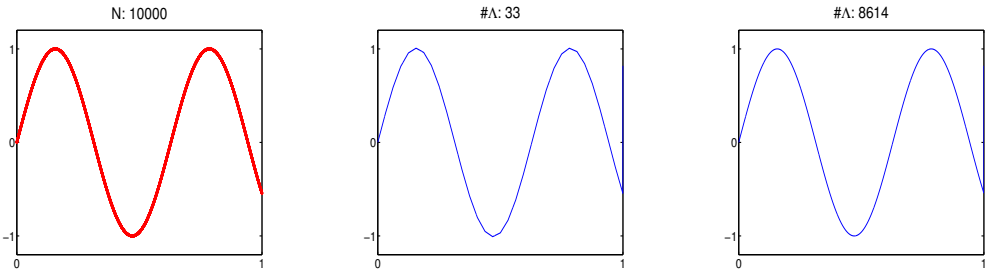


Figure 2.1.2: Redundant information.

the features of P in the 'eye' norm.

2.2 Normal Equations

We recall the computations we need to perform to get the approximating function f. Once a set Λ of indices of basis functions is fixed, taking the $\frac{\partial}{\partial d_\lambda}$ derivatives in equation (1.1.2) for each $\lambda \in \Lambda$ yields the *normal equations* of the problem, namely

$$A^T A\, d = A^T z \Longleftrightarrow: M d = b \tag{2.2.1}$$

where d and z are the column vectors for coefficients and vertical values, and the entries of the *observation matrix* A, the *cross-product matrix* (or *coefficient matrix*) $M = A^T A$ and b are defined by

$$A_{i,\lambda} := \psi_\lambda(x_i), \tag{2.2.2}$$

$$M_{\lambda,\lambda'} := \sum_{i=1}^{N} \psi_\lambda(x_i)\psi_{\lambda'}(x_i), \tag{2.2.3}$$

$$b_\lambda := \sum_{i=1}^{N} z_i\, \psi_\lambda(x_i). \tag{2.2.4}$$

When we want to make explicit the concrete configuration Λ and the set of points X, we will write the observation matrix of Λ with respect to X as $A_{\Lambda,X}$.

With direct solvers like the QR method, it is not easy to take advantage of the sparsity structure of the matrix. We will therefore compute the solution of (2.2.1) by a Conjugate Gradient procedure that works with the matrix $A^T A$ once it has been explicitly formed, or by some other related procedure that allows to work directly with the matrix A, like the CGLS (Conjugate Gradient for Least Squares) or LSQR (a method based on the Lanczos bidiagonalization), see [124].

2.3 Description of the Algorithm

The place and scale localization properties of wavelets (property (L) mentioned in Section 1) allow for a constructive algorithm that locates the relevant degrees of freedom of the problem according to the two goals detailed in Subsection 2.1. The idea is to recursively construct a series of $\Lambda_j, j = 1, \ldots, J$, so that each Λ_j contains adequate wavelets that represent compactly and without spurious oscillations the features of P up to scale j. Λ_j is then used to extract the information from P that we need to predict a set Λ_{j+1} adequate for the features on the next natural scale $j+1$.

2.3.1 Horizontal Adaptivity

The overfitting effects can be partially avoided by an appropriate construction of the set Λ so that wavelets that could introduce artifacts are not included. We use a coarse-to-fine selection algorithm that picks the wavelets in whose support there is indeed information to be represented. We start by constructing the set Λ_{j_0} with all the generator functions and wavelets of some prescribed coarsest level j_0 and then consider the children of the wavelets. Recall that in one spacial dimension a wavelet of index $\lambda = (j, k)$ is said to be a child of a wavelet of index $\lambda' = (j', k')$ if $j = j'+1$ and $k = 2k'$ or $k = 2k'+1$, with a similar definition in the multivariate case.

We construct the set Λ_{j_0+1} keeping only those children in whose support one can find more than a prescribed number q of points. That is, if we denote by $\delta(\Lambda)$ the children of the last-level-wavelets of some set Λ, and by $T_q(\Lambda)$ the subset of a set Λ whose components contain more than q points in their support, the method can be stated as follows.

Algorithm 2.3.1. Horizontal Thresholding

1. Fix j_0 and q.
2. Create Λ_{j_0}.
3. For each $j = j_0, \ldots$.
   (a) Create $\delta(\Lambda_j)$.
   (b) Retain $T_q(\delta(\Lambda_j))$.
   (c) If $T_q(\delta(\Lambda_j)) = \emptyset$ stop, else let $\Lambda_{j+1} := \Lambda_j \cup T_q(\delta(\Lambda_j))$.

Locating points in the support of ψ_λ does not force to run a loop over the whole set of points P, but only over those points that were already in the support of its parent. This allows for an efficient implementation.

We see an application of this technique in Figure 2.3.1. The data points are distributed in two areas with different sampling densities. Moreover, the vertical

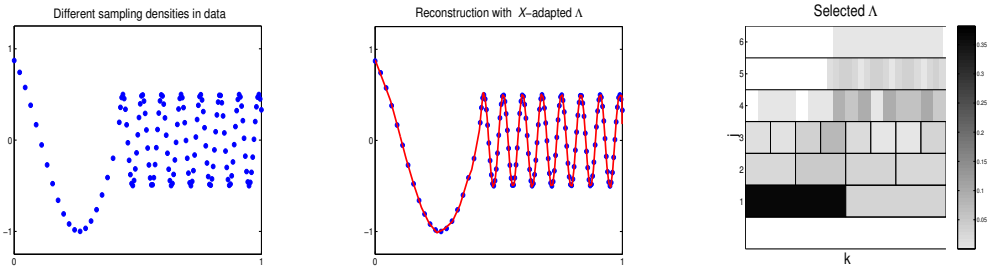


Figure 2.3.1: Choice of Λ adapting to horizontal distribution of data.

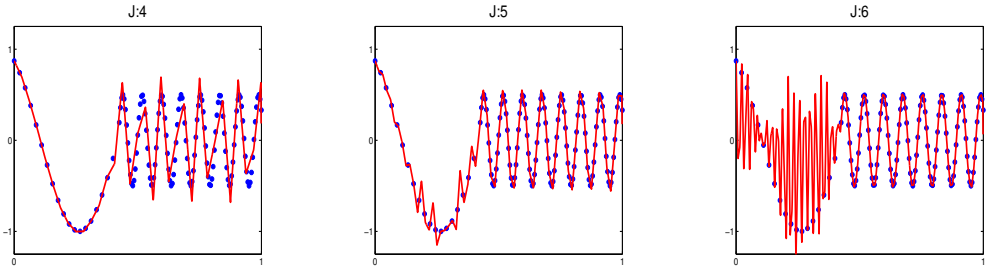


Figure 2.3.2: Difficulties of the full grid approximation.

structure of the data requires a different maximal resolution on each of the two domains. If we let Algorithm 2.3.1 work on this data set, the wavelet indexes given on the right of the figure are automatically detected. Here the vertical axis displays the refinement level j (the algorithm terminates for this example with $j=6$) while the horizontal axis shows the location of wavelet indices. The level of darkness of the boxes in the figure corresponds to their size as indicated in the column bar next to the figure. The reconstruction corresponding to these coefficients is given in the center of the figure.

The attempt of producing a *full grid* approximation, that is, using all the wavelets up to some level J, does not give a satisfying solution. As we see in Figure 2.3.2, $J=4$ does not give enough resolution in the area of big oscillations while $J=6$ produces overfitting in the lower sampling density area. Finally the agreement between these two ends that we find with $J=5$ is not really acceptable.

An example with real word data is provided in Figure 2.3.3. Its represents a set of meteorological data collected by an aerial at the Meteorological Institute of the University of Bonn. The radial sampling geometry is imposed by the measurement conditions: the aerial rotates and makes equispaced measurements in each radial direction in intervals of one degree. In cartesian coordinates, the density of measurements is higher in the center of the domain and decreases gradually in the radial direction. As we can see on the left of Figure 2.3.4, the supports of the wavelets allowed in the configuration constructed by Algorithm 2.3.1 mimic this spatial dis-

tribution. keeping wavelets of high resolution level in the center, whereas in the boundaries only wavelets of low resolution are allowed.

2.3.2 Vertical Adaptivity

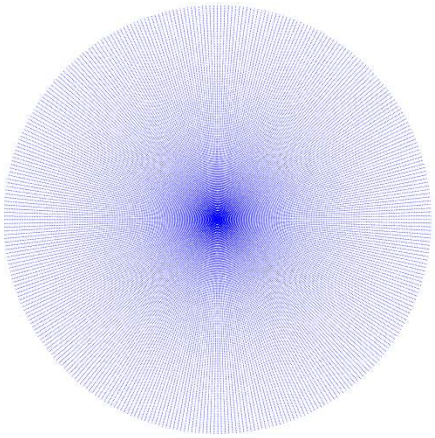
The coarse-to-fine growing algorithm described in Subsection 2.3.1 is able to construct a set Λ that avoids including wavelets in places of the domain where not enough information is available. But it cannot detect if the information is redundant. This can be attained by extending this method by an additional control feature in the step of creating the set Λ_{j+1} according to the information we can extract from Λ_j and P. The idea is to construct an approximation up to level j+1 having already solved for $\{d_\lambda^j\}_{\lambda\in\Lambda_j}$ in

$$\sum_{i=1}^{N}\left(z_i-\sum_{\lambda\in\Lambda_j} d_\lambda^j\ \psi_\lambda(x_i)\right)^2 \to \min!$$

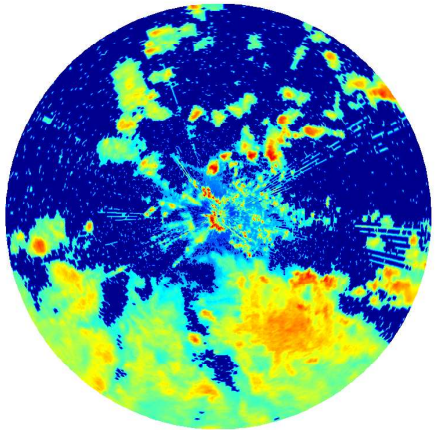
for each previous j. The magnitude of the coefficients can be interpreted as a local smoothness estimator because of the norm equivalence property (R), so we can interpret the presence of big wavelet coefficients as an indicator of potential need for further local refinement. This suggests a level-by-level thresholding procedure that eliminates those degrees of freedom that are detected not to make a significant contribution. This idea can be implemented by a slight modification of the Horizontal Thresholding Algorithm.

Algorithm 2.3.2. Vertical Thresholding

1. Fix j_0, q and $\epsilon > 0$.
2. Create Λ_{j_0}.
3. For each $j = j_0, \ldots$.
   (a) Create $\delta(\Lambda_j)$.
   (b) Select $T_q(\delta(\Lambda_j))$ and construct $\tilde{\Lambda}_{j+1} = \Lambda_j \cup T_q(\delta(\Lambda_j))$.
   (c) Compute $\{d_\lambda^{j+1}\}_{\lambda\in\tilde{\Lambda}_{j+1}}$ that solves $\sum_{i=1}^{N}\left(z_i-\sum_{\lambda\in\tilde{\Lambda}_{j+1}} d_\lambda^{j+1}\ \psi_\lambda(x_i)\right)^2 \to min!$.
   (d) Select $\Lambda_j^\epsilon = \{\lambda \in T_q(\delta(\Lambda_j)) : |d_\lambda^{j+1}| \geq \epsilon\}$.
   (e) If $\Lambda_j^\epsilon = \emptyset$ stop, else let $\Lambda_{j+1} = \Lambda_j \cup \Lambda_j^\epsilon$.

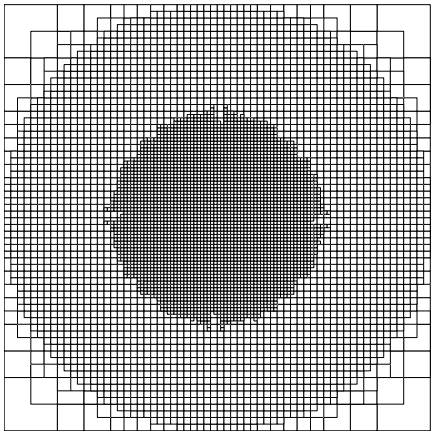


Sampling geometry

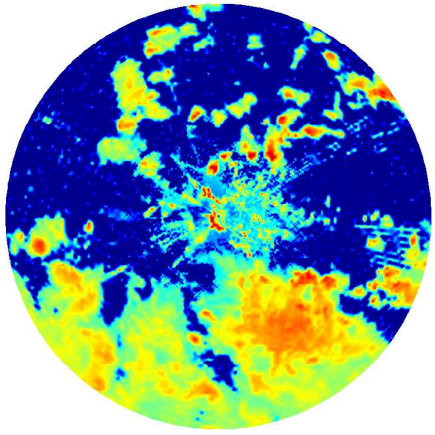


z values

Figure 2.3.3: Radar data set with ~ 50000 measurements. Courtesy of the Meteorological Institute of the University of Bonn.



Wavelet configuration



Reconstruction

Figure 2.3.4: Wavelet reconstruction of radar data from Figure 2.3.3 using a horizontal thresholding with parameter $q = 40$.

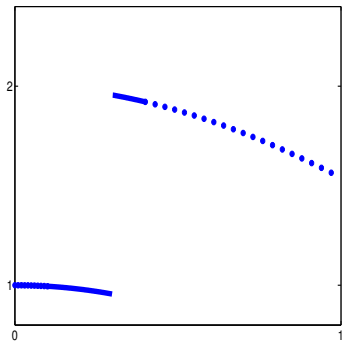


Figure 2.3.5: Test data for Algorithm 2.3.2.

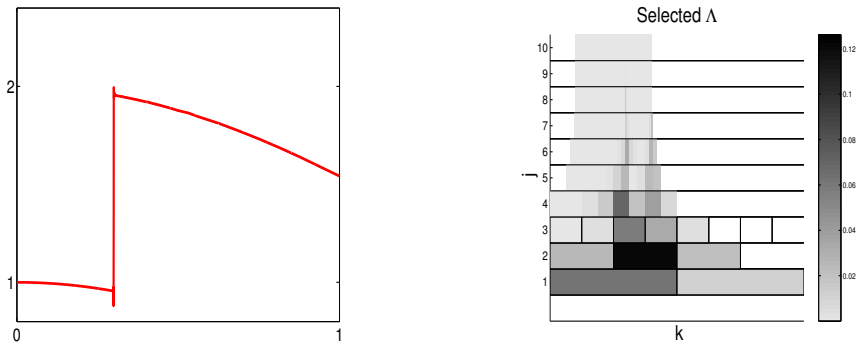


Figure 2.3.6: X-adapted fitting of data in Figure 2.3.5 by the Horizontal Thresholding Algorithm.

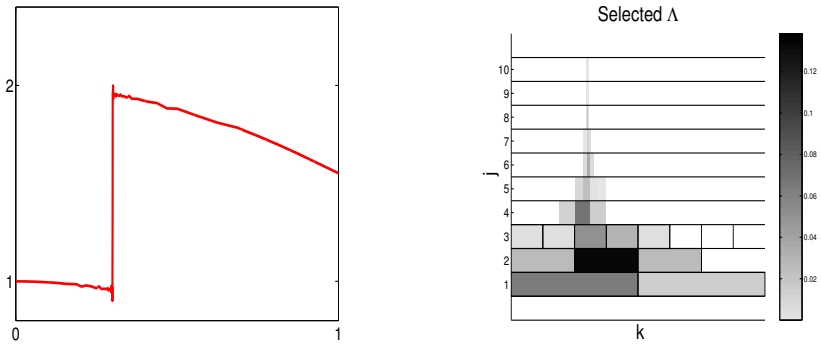


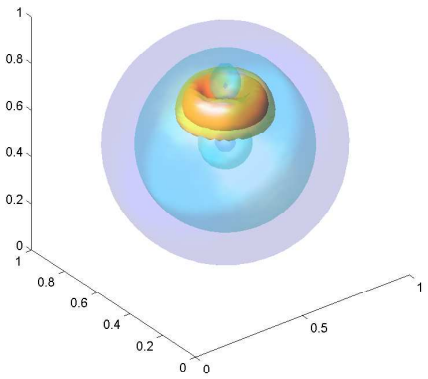
Figure 2.3.7: Z-adapted fitting of data in Figure 2.3.5 by the Vertical Thresholding Algorithm.

Singularity resolution. As an example, let us consider the data in Figure 2.3.5. As to the sampling properties, it shows areas of different densities, including one that concentrates 3000 points. The vertical structure of the data is smooth up to a jump. Constructing an X-adapted approximation using Algorithm 2.3.1 gives the results in Figure 2.3.6. It succeeds to avoid overfitting, but uses 666 wavelets, computed using a coefficient matrix M with 35779 nonzero elements. The Z-adapted fitting of Algorithm 2.3.2 performs a job of similar quality, see Figure 2.3.7, but only 42 wavelets are selected, generating 601 nonzero matrix elements in M and requiring much less computation time.

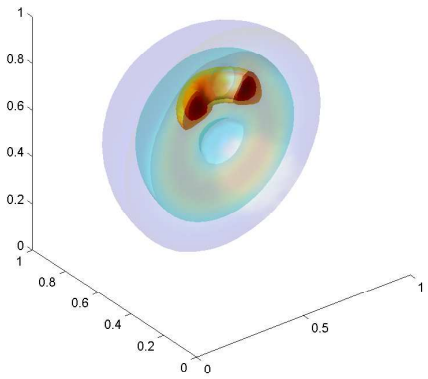
Multivariate data. The algorithm is easily extensible to multivariate data (details of the construction are given in Section 3). In fact, adaptivity is bound to pay off especially well in the case of high dimensional data: If we work with the full grid, the addition of a new dyadic level to a configuration Λ_j adds $\sim 2^{d(j+1)}$ degrees of freedom (d being the dimension of the problem, i.e., the observations lie on the domain $[0,1]^d$). Adding only those wavelets whose support contains a point singularity, the number of new degrees of freedom scales like $k2^d$, k depending on the wavelet family. Thus, the ratio

$$\frac{\#\text{ d.o.f.s by adaptive addition}}{\#\text{ d.o.f.s by addition of full level}} \sim \frac{k2^d}{2^{d(j+1)}} = k2^{-dj}$$

makes clear how the reduction of complexity in high dimensional data sets is going to be more visible and computationally significant. Consider for instance the $3d$ data in Figure 2.3.8. The reconstruction with the full grid up to level $j = 3$ gives a good reconstruction in the smooth areas, but the high gradients appearing in the proximity of the ring are not correctly solved, see Figure 2.3.10. The thresholding algorithm chooses new wavelets of level $j = 4$ only in this area (see center of the Figure), providing the necessary degrees of freedom for a more accurate reconstruction, on the right of the Figure. At this level, only 2107 new degrees of freedom of the 35937 possible ones are chosen by the method.

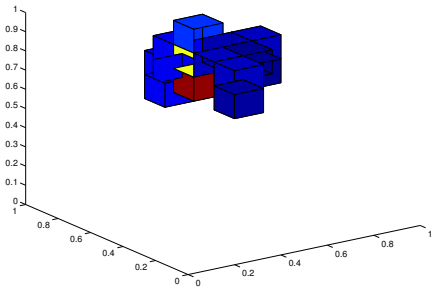


Original data.

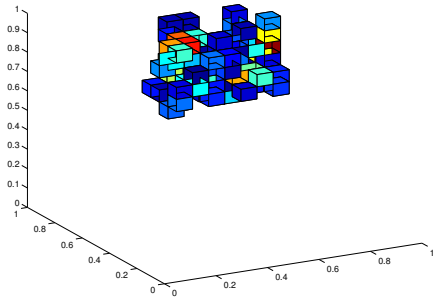


Section across $y = 0.5$.

Figure 2.3.8: The nucleon data set displays the results of a simulation of the two-body distribution probability of a nucleon in the atomic nucleus 16O if a second nucleon is known to be positioned in the neighborhood. Courtesy of VolVis and SFB 382 of the German Research Council (DFG).

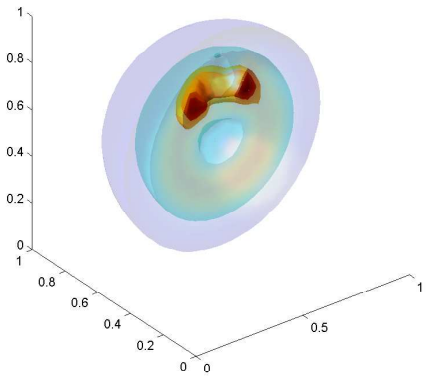


Coefficients to be refined at level $j = 3$

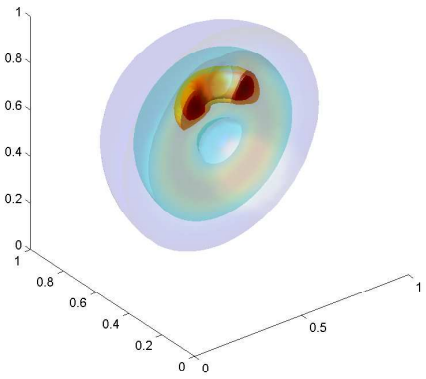


Coefficients at level $j = 4$

Figure 2.3.9: Evolution of the tree created by the Vertical Thresholding Algorithm on the nucleon data from Figure 2.3.8. For clarity, only coefficients of wavelets of tensorial type $\mathbf{e} = (1, 1, 1)$ are depicted (see Section 3.5 for an explanation of this indexing).



Reconstruction at $j = 3$



Adaptive reconstruction at $j = 4$

Figure 2.3.10: Evolution of the reconstruction of the nucleon data at different dyadic levels.

2.4 Stability of the Algorithm

2.4.1 A Stable Variation

As explained in Section 2.2, the tree-growth approach requires the solution of the normal equations

$$M_j d_j = b_j \tag{2.4.1}$$

on index sets Λ_j-s determined by the method. Here arises the question whether the proposed algorithms generate a succession of invertible $\{M_j\}_{j\geq j_0}$. As a matter of fact, the q-method described in Algorithm (2.3.1) does not assure that each M_j is non-singular. But a little variation of the method can indeed give us a constructive algorithm that produces invertible matrices at each level j.

In the following exposition we will use the term "wavelet" referring to a member of the family of the linear B-Spline Wavelet in one dimension, but the results and corresponding proofs are easily generalizable to both higher smoothness degrees and multivariate wavelets.

First of all we will introduce some definitions and nomenclature that will simplify and motivate the further developments.

Definition 2.4.1. [**Internodal spaces**] *For a wavelet ψ_λ, $\lambda = (j,k)$, we denote the internodal spaces of this wavelet ψ_λ by*

$$I_\lambda^i := \left[\frac{2(k-1)+i-1}{2^{j+1}}, \frac{2(k-1)+i}{2^{j+1}}\right] \tag{2.4.2}$$

for $i = 1,\ldots,6$. We introduce as well the notation $I_\lambda^{i,i+1}$ for $I_\lambda^i \cup I_\lambda^{i+1}$ for i=1, . . . , 5.

See an illustration of this definition in Figure 2.4.1. The internodal spaces are just the segments on which the wavelet is a linear function. We can then write any arbitrary ψ_λ as

$$\psi_\lambda(x) = \begin{cases} a_\lambda^i x + b_\lambda^i, & x \in I_\lambda^i,\ i = 1,\ldots,6 \\ 0 & otherwise. \end{cases} \tag{2.4.3}$$

The numerical values of these coefficients can be computed exactly ([136]). However we will only need the following properties

$$a_\lambda^3 > a_\lambda^5 > a_\lambda^1 > 0 \tag{2.4.4}$$

and

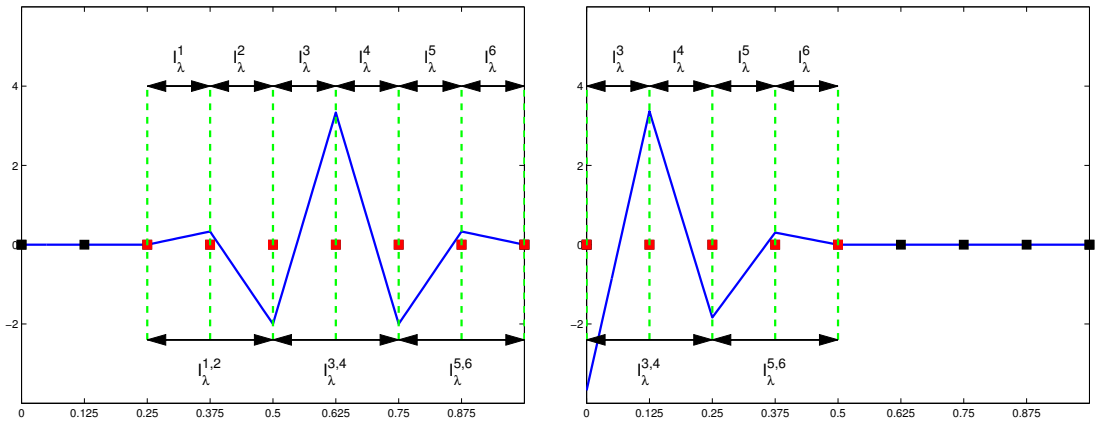
$$a_\lambda^4 = -a_\lambda^3,\ a_\lambda^6 = -a_\lambda^1,\ a_\lambda^2 = -a_\lambda^5. \tag{2.4.5}$$

Note that for two interior wavelets λ and λ' on the same dyadic level we have

$$a_\lambda^i = a_{\lambda'}^i,\ i = 1,\ldots,6 \tag{2.4.6}$$

by translation.

We introduce now a key concept, which will turn up to give an exact measure of the relevancy of a wavelet to correctly describe a set of points.



In an interior wavelet In a boundary wavelet

Figure 2.4.1: Internodal spaces.

Definition 2.4.2. [X**-balanced wavelet**] *Given a cloud of points P with horizontal values X, we say that the wavelet ψ_λ is X-balanced or shortly balanced if there exist three points $x_a, x_b, x_c \in X$ so that $x_a \in I_\lambda^3, x_b \in I_\lambda^4$ and $x_c \in I_\lambda^{3,4}$.*

We examine now the following variant of the Algorithm 2.3.1 which integrates the concept of X-balanced wavelets.

Algorithm 2.4.3. Stable Growth with Horizontal Thresholding.

1. Fix j_0 and create the full grid Λ_{j_0} so that every ψ_λ, $\lambda \in \Lambda_{j_0}$ is X-balanced.
2. For each $j = j_0, \ldots$.
   (a) Create $\delta(\Lambda_j)$.
   (b) Construct $T_{balanced}(\delta(\Lambda_j)) := \{\lambda \in \delta(\Lambda_j), \lambda \text{ is } X\text{-balanced}\}$.
   (c) If $T_{balanced}(\delta(\Lambda_j)) = \emptyset$ stop, else let $\Lambda_{j+1} := \Lambda_j \cup T_{balanced}(\delta(\Lambda_j))$.

We will now devote the rest of this section to prove that this algorithm is *stable* in the sense that all the matrices involved in it are indeed invertible.

Remark 2.4.4. *We wish to point out an important feature of this characterization: we have practically reduced the problem of the invertibility of the matrix, which is global to the whole set of wavelets at hand at each moment, to a precise local check of the information contain of each wavelet.*

Other characterizations are possible, and the one that we propose in this algorithm is not optimal. For instance, it is not necessary for every wavelet of a full grid to be balanced in order to warrant the invertibility of the matrix. One can easily use Theorem 2.4.5 below to see that it reaches if

- just one wavelet on the top level is X-balanced, and
- all the others have two points in their support, one in the third and one in the fourth internodal space.

These conditions are weaker than the one proposed in Algorithm 2.4.3, but their verification requires a joint check of all the wavelets of the new level. The strategy described in Algorithm 2.4.3 is better to the computational implementation, which benefits from the possibility of making a definitive decision about accepting or rejecting an individual wavelet without having to locate on-line the data structures of the other wavelets of the same level.

First we introduce an useful notation for the matrices that we are going to use. As before, given a set of horizontal points X and a set of wavelets indexed by Λ, we denote by A_X^Λ the respective observation matrix, that is the matrix

$$A_X^\Lambda := \begin{pmatrix} \psi_{\lambda_1}(x_{\theta(1)}) & \cdots & \psi_{\lambda_{\#\Lambda}}(x_{\theta(1)}) \\ \vdots & \ddots & \vdots \\ \psi_{\lambda_1}(x_{\theta(N)}) & \cdots & \psi_{\lambda_{\#\Lambda}}(x_{\theta(N)}) \end{pmatrix}. \tag{2.4.7}$$

for some reordering $\theta : \{1, \ldots, N\} \to \{1, \ldots, N\}$ of the original point indices. We will define no particular reordering, as it is of no consequence for our purposes. We will be interested in the rank of the matrix, which is independent of the row or column ordering.

We will use the same super and subscript notation to refer to the coefficient matrices, that is, we will write $M_X^\Lambda := (A_X^\Lambda)^T A_X^\Lambda$.

A trivial, but important observation, is that if M_X^Λ is invertible, $M_{X\cup X'}^\Lambda$ is also invertible for any additional set of points X'. This means: if we fix the wavelet configuration, the addition of new observations cannot make the matrix singular.

We will use this fact in the inverse direction: if we are given a set of points X, and we can extract a subset $\tilde{X} \subset X$ of them so that the coefficient matrix $M_{\tilde{X}}^\Lambda$ is invertible, than we know that the coefficient matrix M_X^Λ is invertible as well.

We start now with the analysis of the matrices arising in Algorithm 2.4.3. We first need to recall a classic result ([132]) from the theory of interpolation with splines.

Theorem 2.4.5. [Schoenberg-Whitney conditions] *Let $\{t_i\}_{i=1,\ldots,n+m}$ be an increasing sequence of knots and $\{x_i\}_{i=1,\ldots,n}$ a strictly increasing sequence of points. Let us denote by N_j the j-th B-Spline of order m constructed on the knots $\{t_i\}$, and by A^{BS} the collocation matrix of the problem, that is*

$$A_{i,j}^{BS} := N_j(x_i), \text{ for } i, j = 1, \ldots, n. \tag{2.4.8}$$

Then, A^{BS} is invertible if and only if

$$t_j < x_j < t_{j+m},\ j = 1, \ldots, n. \tag{2.4.9}$$

This result can be almost immediately transferred to the first step of Algorithm 2.4.3. We check that the coefficient matrix at level j_0 is invertible.

Lemma 2.4.6. *The matrix M_{j_0} produced in the first step of Algorithm 2.4.3 is invertible.*

Proof:
This can be seen using the Schoenberg-Whitney theorem to check the invertibility of a collocation matrix in a nodal basis and then transferring this invertibility to the collocation matrix in the wavelet basis using the wavelet transform, which is known to be non-singular.

In fact, as we have assured that all the wavelets in Λ_{j_0} are balanced, we can extract from X a subset of points $X^0 = \{x_i^0\}_{i=1,\ldots,2^{j_0+1}}$ that together with the knots $\{t_i\}$

$$(t_1, \ldots, t_{2^{j_0}+1}) := 2^{-j_0}(0, 0, 1, 2, \ldots, 2^{j_0} - 1, 2^{j_0}, 2^{j_0}) \tag{2.4.10}$$

met the interlacing condition (2.4.9) for order $m = 2$. This is so, because the interval $[0, 1]$ can be split as central internodal spaces of the wavelets of the level j_0, and in each of these intervals there must be at least one point, as the wavelets are balanced. If we denote by $\{\phi_i\}$ the family of splines constructed from the knot sequence (2.4.10), As a result of Theorem 2.4.5 the collocation matrix $A_{X^0}^{BS}$ of this family with respect to the points in X^0 is non-singular. Now, $\{\phi_i\}$ are the scaling functions of the (preorthogonal)-linear wavelets, that is, the wavelets indexed in Λ_{j_0} are linear combinations of elements of $\{\phi_i\}$, so that

$$A_{X^0}^{\Lambda} = T A_{X^0}^{BS} \tag{2.4.11}$$

where the wavelet transformation matrix T is per construction non-singular (see [136]). Thus $A_{X^0}^{\Lambda}$ is non-singular. Consequently $M_{X^0}^{\Lambda_{j_0}}$ is non-singular and so is $M_{j_0} = M_X^{\Lambda_{j_0}}$. ∎

Now we deal with the possibility that the subsequent steps of the algorithm do not produce a full grid. Our strategy will be to check under which conditions on λ' and X we can add a new wavelet indexed by λ' to a previous set of wavelets known to generate an invertible matrix, in a way that the resulting matrix is still invertible.

2.4.2 Randering of a Matrix

First we will interpret the addition of a wavelet to a set of wavelets as the *randering* of the original matrix. We introduce first the following notation: given an arbitrary set of points X and a set of indices Λ we will denote by

- $\Psi_{\tilde{\lambda}}(X)$ the row vector in $\mathbb{R}^N$ containing the evaluation of $\psi_{\lambda'}$ on each of the points in X with components ordered as in the rows of A_X^{Λ}, and by

- $\Psi_\Lambda(\tilde{x})$ the column vector in $\mathbb{R}^{\#\Lambda}$ containing the evaluation of each $\psi_\lambda, \lambda \in \Lambda_{j_0}$ at the point $\tilde{x}$ with components ordered as in the columns of A_X^Λ,

Definition 2.4.7. [**Randered matrix**] *Assume we are given a set of not coinciding points X and a set of wavelet indices Λ so that the corresponding observation matrix A_X^Λ is a square and non-singular matrix. For $\tilde{x} \notin X$ and $\tilde{\lambda} \notin \Lambda$, we call the matrix*

$$A_{X\cup\tilde{x}}^{\Lambda\cup\lambda'} = \left(\begin{array}{c|c} A_X^\Lambda & \Psi_{\tilde{\lambda}}(X) \\ \hline \Psi_\Lambda(\tilde{x})^T & \psi_{\tilde{\lambda}}(\tilde{x}). \end{array}\right).$$

the randering of A_X^Λ with respect to $\tilde{\lambda}$ and $\tilde{x}$.

We need two additional steps to insert this concept into our setting. First we recall a result from Numerical Linear Algebra that gives general conditions for the randering of a matrix to have full rank (Lemma 2.4.8). Then we will state a geometrical interpretation of this result in our particular case (Proposition 2.4.10).

Lemma 2.4.8. *If a matrix $A \in \mathbb{R}^{n\times n}$ has full rank, the randered matrix*

$$\tilde{A} = \left(\begin{array}{c|c} A & p \\ \hline q^T & \omega \end{array}\right).$$

has also full rank if and only if the matrix A, the vectors $p, q \in \mathbb{R}^n$ and the scalar ω satisfy

$$\omega \neq q^T A^{-1} p. \tag{2.4.12}$$

Definition 2.4.9. *Given a set of wavelet indices Λ, a set of points $X = \{x_i\}_{i=1,\ldots,\#\Lambda}$ so that A_X^Λ is invertible, and a set of points $Y = \{y_i\}_{i=1,\ldots,\#\Lambda}$, we denote by $s_{[\Lambda;X;Y]}$ the function that fulfills*

$$s_{[\Lambda;X;Y]} \in Span\{\psi_\lambda : \lambda \in \Lambda\} \tag{2.4.13}$$

and

$$s_{[\Lambda;X;Y]}(x_i) = y_i,\ i = 1, \ldots, \#\Lambda. \tag{2.4.14}$$

Note that this function exists and is uniquely defined.

Proposition 2.4.10. *With the notation used in Definitions 2.4.7 and 2.4.9, we have*

$$s_{[\Lambda;X,\Psi_{\lambda'}(X)]}(x') = \Psi_\Lambda(x')^T (A_X^\Lambda)^{-1} \Psi_{\lambda'}(X). \tag{2.4.15}$$

Proof:
As $s_{[\Lambda;X;\Psi_{\lambda'}(X)]} \in Span\{\psi_\lambda : \lambda \in \Lambda\}$, we can write

$$s_{[\Lambda;X;\Psi_{\lambda'}(X)]}(x) = \sum_{\lambda\in\Lambda} d_\lambda^s \psi_\lambda(x) \tag{2.4.16}$$

for some vector $d^s = \{d_\lambda\}_{\lambda\in\Lambda}$. This vector must fulfill (2.4.14), that is

$$\sum_{\lambda\in\Lambda} d^s_\lambda \psi_\lambda(x_i) = \psi_{\lambda'}(x_i),\ i = 1,\dots,\#\Lambda \tag{2.4.17}$$

or in matrix-vector form

$$A^\Lambda_X d^s = \Psi_{\lambda'}(X). \tag{2.4.18}$$

Now, evaluating (2.4.16) at the point x', we obtain

$$s_{[\Lambda;X;\Psi_{\lambda'}(X)]}(x') = \sum_{\lambda\in\Lambda} d^s_\lambda \psi_\lambda(x') = \Psi_\Lambda(x')^T d^s. \tag{2.4.19}$$

As A^Λ_X is assumed to be invertible, (2.4.18) and (2.4.19) give (2.4.15). ∎

2.4.3 Stability Theorems

We can now prove our main results.

Theorem 2.4.11. [Vertical Extension] *We are given a set of points X, and a set of wavelet indices Λ_{j_0} so that*

- *the wavelets complete a full grid at level j_0, and*
- *each ψ_λ, $\lambda \in \Lambda_{j_0}$, is X-balanced.*

Then, if $\psi_{\lambda'}$ with $\lambda' = (j_0{+}1, k)$ for some k is X-balanced, the matrix $M_X^{\Lambda_{j_0}\cup\lambda'}$ is invertible.

Proof:
First, as the wavelets in Λ_{j_0} are X-balanced, we can extract from X a succession of points $\widetilde{X_0} := \{x^0_k\}_{k=0,\dots,2^{j_0+1}-1,k\neq k'}$ fulfilling the inclusions:

$$x^0_k \in I^{3,4}_{j_0+1,k},\ k = 0,\dots,2^{j_0+1}{-}1,\ k \neq k'. \tag{2.4.20}$$

(See Figure 2.4.2 for a visualization in a concrete example of the concepts used in this proof). In the central segment of the wavelet of interest $\psi_{\lambda'}$ we will chose two points, each in one different central internodal space: $x^0_{k',3} \in I^3_\lambda$ and $x^0_{k',4} \in I^4_{\lambda'}$. Defining $X_0 := \widetilde{X_0} \cup \{x^0_{k',3}, x^0_{k',4}\}$, we immediately see as a direct result of Lemma 2.4.6 that the matrix $A^{\Lambda_{j_0}}_{X_0}$ is invertible, as all the wavelets in Λ_{j_0} are X_0-balanced. Further, as $\psi_{\lambda'}$ is balanced we can find a third point $x'_{\lambda'}$ lying in one of the two central internodal spaces of $\psi_{\lambda'}$.

Now, the interpolating function $s_{[\Lambda_{j_0};X_0;\Psi_{\lambda'}(X_0)]}$ is an element of the span of $\{\psi_\lambda, \lambda \in \Lambda_{j_0}\}$. But this span equals the space of the functions that are piecewise linear on the intervals $I^{3,4}_\lambda$ and continuous at the points $\frac{k}{2^{j_0+1}}$. Additionally,

$s_{[\Lambda_{j_0};X_0;\Psi_{\lambda'}(X_0)]}$ cuts the wavelet $\psi_{\lambda'}$ at the points $x^0_{k',3}$ and $x^0_{k',4}$ per construction, which determines $s_{[\Lambda_{j_0};X_0;\Psi_{\lambda'}(X_0)]}$ on the internodal space $I^{3,4}_{\lambda'}$.

But the points $\psi_{\lambda'}(x^0_{k',3}), \psi_{\lambda'}(x^0_{k',4})$ and $\psi_{\lambda'}(x_{\lambda'})$ are not collinear, which implies that $s_{[\Lambda_{j_0};X_0;\Psi_{\lambda'}(X_0)]}(x')$ cannot be identical to $\psi_{\lambda'}(x')$. Using Proposition 2.4.10 this is equivalent to

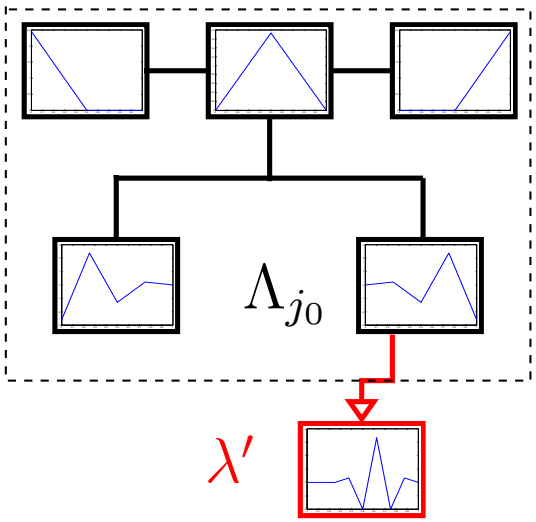
$$\Psi_{\Lambda_{j_0}}(x')^T \left(A^{\Lambda_{j_0}}_{X_0}\right)^{-1} \Psi_{\lambda'}(X_0) \neq \psi_{\lambda'}(x') \tag{2.4.21}$$

which, in view of Lemma 2.4.8 leads us to conclude that the matrix

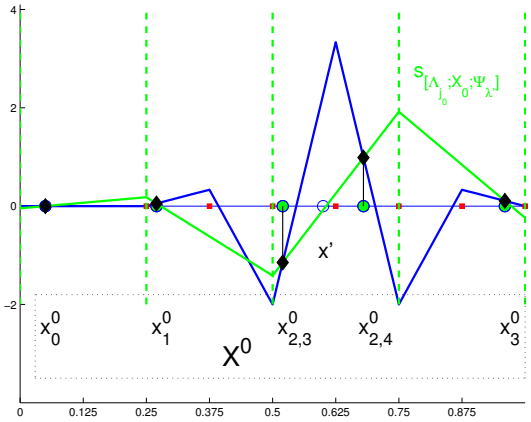
$$A^{\Lambda_{j_0}\cup\lambda'}_{X_0\cup x'} = \left(\begin{array}{c|c} A^{\Lambda_{j_0}}_{X_0} & \Psi_{\lambda'}(X_0) \\ \hline \Psi_{\Lambda_{j_0}}(x')^T & \lambda'(x'). \end{array}\right).$$

is invertible and so is $A^{\Lambda_{j_0}\cup\lambda'}_{X}$.

■



The level $j_0 = 1$ is enlarged by the wavelet $\psi_{2,2}$.



An interpolant constructed with elements linear on intervals of length 2^{-j_0} cannot reproduce a oscillation of frequency 2^{j_0+1}.

Figure 2.4.2: Illustration of Theorem 2.4.11 proving the stability of the vertical extension.

Theorem 2.4.12. [Horizontal Extension] *If the matrix* $A^{\Lambda_{j_0}\cup\lambda'}_{X_0\cup x'}$ *for* $\Lambda_{j_0}, \lambda', X_0$ *and* x' *are like in Theorem 2.4.11, and* $\psi_{\lambda''}, \lambda' = (j_0{+}1, k''), k'' = k'{-}1$ *is an interior,* X*-balanced wavelet, then the matrix* $A^{\Lambda_{j_0}\cup\lambda'\cup\lambda''}_{X}$ *is invertible.*

Proof:
Define $X' := X_0 \cup x'$ and $\Lambda' := \Lambda_{j_0} \cup \lambda'$. For $\psi_{\lambda''}$, we can choose a point $x'' \in X$,

$x'' \notin X'$. Now, the interpolant $s_{[\Lambda',X';\Psi_{\lambda''}(X')]}$, which is well defined as a result of the former theorem, will be shortly denoted as s. We illustrate the situation in Figure 2.4.3, following the concrete example proposed in Figure 2.4.2. Note the following properties.

1. In the interval $I^5_{\lambda''}$, which is equal to $I^3_{\lambda'}$, s cuts per construction two points of $\psi_{\lambda''}$, namely at $x^0_{k',3}$ and x'. As both s and $\psi_{\lambda''}$ are linear on $I^5_{\lambda''}$, we have

$$s(x) = \psi_{\lambda''}(x),\ \forall x \in I^5_{\lambda''}. \tag{2.4.22}$$

   Furthermore, in the interval $I^6_{\lambda''}$, or equivalently $(I^4_{\lambda'})$, s cuts $\psi_{\lambda''}$ in the point $x^0_{\lambda',4}$ *and*, because of (2.4.22), in the point where $I^5_{\lambda''}$ and $I^6_{\lambda''}$ overlap. Therefore, as s and $\psi_{\lambda''}$ are linear on $I^6_{\lambda''}$ we must have

$$s(x) = \psi_{\lambda''}(x),\ \forall x \in I^6_{\lambda''}. \tag{2.4.23}$$

2. If $s(x'')$ is required to be $\psi_{\lambda''}(x'')$ than we would necessarily have

$$s(x) = \psi_{\lambda''}(x),\ \forall x \in I^4_{\lambda''} \tag{2.4.24}$$

   as s cuts $\psi_{\lambda''}$ in a second point of $I^4_{\lambda''}$, namely in the point where $I^4_{\lambda''}$ and $I^5_{\lambda''}$ overlap. Likely, we can expand the reasoning to conclude

$$s(x) = \psi_{\lambda''}(x),\ \forall x \in I^3_{\lambda''}. \tag{2.4.25}$$

Now, s is an element of the span of $\{\psi_\lambda, \lambda \in \Lambda_0 \cup \lambda'\}$. The elements in Λ_0 are linear in $I^{3,4}_{\lambda''}$ and $I^{5,6}_{\lambda''}$. An element of this span fulfilling (2.4.22),(2.4.23),(2.4.24) and (2.4.25) has to fulfill

$$\gamma\psi_{\lambda'}(x) + Ax + B = \psi_{\lambda''}(x),\ \forall x \in I^{3,4}_{\lambda''} \tag{2.4.26}$$

and

$$\gamma\psi_{\lambda'}(x) + Cx + D = \psi_{\lambda''}(x),\ \forall x \in I^{5,6}_{\lambda''} \tag{2.4.27}$$

for a set of real constants γ, A, B, C, D. In view of (2.4.3) The conditions on the pendants in (2.4.26) lead to the system

$$\begin{array}{l} \gamma a^1_{\lambda'} + A = a^3_{\lambda''} \\ \gamma a^2_{\lambda'} + A = a^4_{\lambda''} \end{array} \tag{2.4.28}$$

and those on (2.4.27) lead to

$$\begin{array}{l} \gamma a^3_{\lambda'} + C = a^5_{\lambda''} \\ \gamma a^4_{\lambda'} + C = a^6_{\lambda''}. \end{array} \tag{2.4.29}$$

If we suppose for simplicity that both $\psi_{\lambda'}$ and $\psi_{\lambda''}$ are interior we can omit the subindices λ' and λ'' as noted in (2.4.6) and use the Cramer's rule to solve the two systems. (2.4.28) gives the value

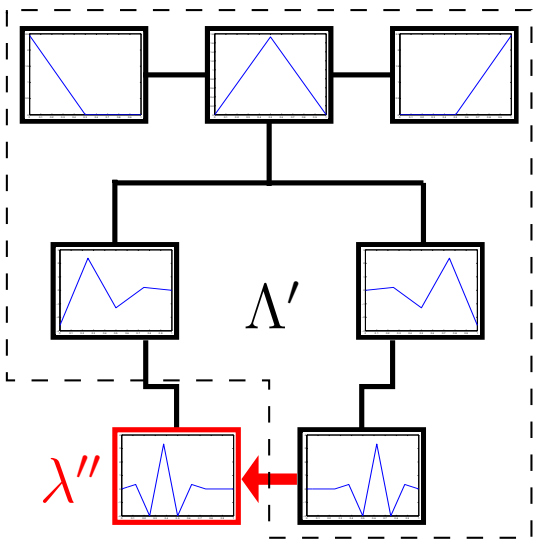
$$\gamma = \frac{a^3 - a^4}{a^1 - a^2}. \tag{2.4.30}$$

Using (2.4.4), this yields $\gamma > 1$, and (2.4.29) gives

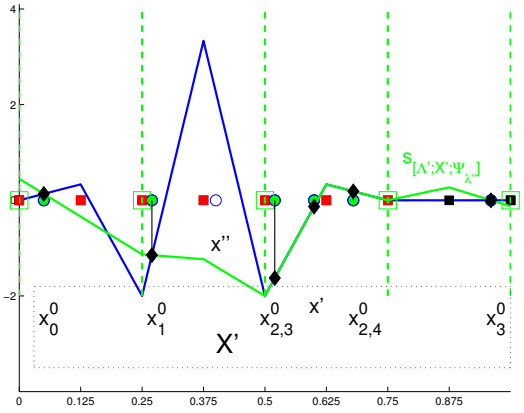
$$\gamma = \frac{a^5 - a^6}{a^3 - a^4} = \frac{a^1 - a^2}{a^3 - a^4}, \tag{2.4.31}$$

where we used (2.4.5). Using (2.4.4), we would have $\gamma < 1$. That is, there is no element of the span $\{\Lambda_0 \cup \lambda'\}$ that can reproduce $\psi_{\lambda''}$ on $I_{\lambda''}^{3,4,5,6}$. This means that $s(x'')$ cannot be $\psi_{\lambda''}(x'')$ and in consequence we can now invoke Proposition 2.4.10 and Lemma 2.4.8 to complete the proof.

■



The configuration of Figure 2.4.3 is enlarged by addition of the wavelet $\psi_{2,1}$, which neighbors a previous one.



The addition of $\psi_{2,2}$ to Λ_{j_0} gives Λ' the possibility of reproducing a localized oscillation of frequency 2^{j_0+1}, but it cannot simultaneously reproduce *two* of them: the one in $I_{2,1}^{5,6}$ is correctly captured, but fails to create $\psi_{2,1}$ on $I_{2,1}^{3,4}$.

Figure 2.4.3: Illustration of Theorem 2.4.12 proving the stability of the horizontal expansion.

Theorems 2.4.11 and 2.4.12 give us the way to construct any set of wavelets with tree structure arising from Algorithm 2.4.3, by vertical or horizontal addition of

individual wavelets. The restrictions in Theorem 2.4.12 imposing that the wavelets are interior were made just in order to ease the readability of the proofs and the general case can be easily inferred. The same can be said about the restriction in Theorem 2.4.11 regarding the extension of a full level, as the generalization to extend arbitrary lacunary levels is obvious. Thus, we have our main result.

Theorem 2.4.13. *The matrices M_j arising at each level j of the Algorithm 3. are invertible.*

Chapter 3

Numerical Performance

One motivation to work with wavelets are their ability to compress information, well stated in Nonlinear Approximation Theory. This lets one hope that the degrees of freedom involved in computations using a wavelets basis are likely to be fewer, and so some advantage in terms of computational time is expectable.

The objective of this section is to point out that in addition to complexity reduction, the numerics involved in the described data fitting algorithm has inherently an advantageous formulation in a wavelet basis. The additional advantages we could expect are due to

- the good condition properties typically attached to wavelet bases and
- their level-to-level orthogonality, property (O).

We first explain why we should expect these two factors to yield numerical advantages, and then we illustrate how they actually work by means of a series comparison examples in which the same cloud of points P is fitted with two different bases. The $\{\psi_\lambda\}_{\lambda\in\Lambda}$ are in one case the linear 2d tensor B-spline wavelet basis and in the other a 2d hierarchical basis produced by tensor products of the 1d hierarchical basis. As we will see, this latter basis provides also a multiscale setup but does not benefit from the two above features.

3.1 Conditioning

In this section we explore some basic numerical properties of our setting. As explained in Section 2.2, we want to use iterative methods such as the Conjugate Gradient Method (CG Method) on the system of normal equations (2.2.1). This motivates the study of the behavior of the *spectral condition numbers* of the systems arising during Algorithm 2.3.2, as they govern the speed of iterative techniques.

This issue relates to the stability properties of the basis involved, a subject that has been deeply investigated in multilevel preconditioning techniques for PDEs, see [47], from which we now recall a few concepts. To fix ideas, we will work with a

Hilbert space H embedded in L_2. We give us a sequence of *nested spaces* $\{V_j\}_{j\geq 0}$ approximating H, that is

$$V_0 \subset V_1 \subset \ldots \subset V_j \subset \ldots \subset H \tag{3.1.1}$$

with

$$\overline{\bigcup_{j\geq 0} V_j} = H \tag{3.1.2}$$

and each V_j being spanned by a set of basis functions $\Psi_j := \{\psi_\lambda\}_{\lambda\in\Lambda_j}$. As before, the indices $\lambda \in \Lambda_j$ typically include information about the scale and position of the corresponding function.

If the sets Ψ_j are nested as well, that is if

$$\Psi_0 \subset \Psi_1 \subset \ldots \subset \Psi_j \subset \ldots \subset \Psi, \tag{3.1.3}$$

then $\Psi := \{\psi_\lambda\}_{\lambda\in\Lambda}$, $\Lambda_j \subset \Lambda$ for each $j \geq 1$, is a *multiresolution basis* of H. We denote by Ψ (resp. Ψ_j) both a set as well as a vector containing the elements of the set. In this chapter we well work with uniform discretizations and Ψ_j corresponds to the full grid approximation at level j. Only in this case comparison between different bases makes sense.

Definition 3.1.1. *Given a basis Ψ for a space H with inner product $(\cdot,\cdot)_H$, define the Gramian matrix G_H^ψ of Ψ with respect to H as the matrix with entries*

$$(G_H^\psi)_{\lambda,\lambda'} := (\psi_\lambda, \psi_{\lambda'})_H,\ \lambda, \lambda' \in \Lambda. \tag{3.1.4}$$

Definition 3.1.2. *A basis Ψ for a space H is said to be stable if there exist positive finite constants c_1, c_2 independent of j such that*

$$c_1\|u\|_{\ell_2(\Lambda)} \leq \|u^T\Psi\|_H \leq c_2\|u\|_{\ell_2(\Lambda)} \tag{3.1.5}$$

holds for every $u \in \ell_2(\Lambda)$.

Definition 3.1.3. *A multiresolution basis Ψ for a space H is said to be uniformly stable, if there exist positive finite constants c_1, c_2 such that*

$$c_1\|u_j\|_{\ell_2(\Lambda_j)} \leq \|u_j^T\Psi_j\|_H \leq c_2\|u_j\|_{\ell_2(\Lambda_j)} \tag{3.1.6}$$

holds for every $u_j \in \ell_2(\Lambda_j)$ and for every $j \geq 0$.

Now, we can formulate a simple but useful consequence of (3.1.5).

Lemma 3.1.4. *Let Ψ_j be a basis for a subspace V_j of a Hilbert space H with Gramian matrix $G_{V_j}^{\Psi_j}$. If Ψ_j is stable according to Definition 3.1.2 with constants c_1 and c_2, then*

$$cond(G_{V_j}^{\Psi_j}) \leq \frac{c_2}{c_1}, \tag{3.1.7}$$

and equality occurs when c_1 and c_2 are tight in (3.1.5).

Proof:
We just note that

$$\frac{\|u_j^T\Psi_j\|_H^2}{\|u_j\|_{\ell_2}^2} = \frac{(u_j^T\Psi_j, u_j^T\Psi_j)_H}{u_j^Tu_j} = \frac{u_j^TG_{V_j}^{\Psi_j}u_j}{u_j^Tu_j}. \tag{3.1.8}$$

Recall that the condition of a matrix is defined as the ratio of its extreme eigenvalues

$$cond(G_{V_j}^{\Psi_j}) = \frac{\lambda_{\max}}{\lambda_{\min}}. \tag{3.1.9}$$

For symmetric positive definite matrices it is well known that the biggest and smallest eigenvalues meet

$$\lambda_{\max} = \max_{u_j} \frac{u_j^TG_{V_j}^{\Psi_j}u_j}{u_j^Tu_j}, \quad \lambda_{\min} = \min_{u_j} \frac{u_j^TG_{V_j}^{\Psi_j}u_j}{u_j^Tu_j}, \tag{3.1.10}$$

and therefore

$$\lambda_{\min} \leq \frac{u_j^TG_{V_j}^{\Psi_j}u_j}{u_j^Tu_j} \leq \lambda_{\max} \tag{3.1.11}$$

with tight bounds. ∎

This lemma states that the sequence of Gramian matrices $\{G_H^{\Psi_j}\}_{j\geq 0}$ has uniformly bounded condition numbers if Ψ is uniformly stable.

Now, if go back to our setting in (2.2.1), we find at each level of the tree growth the coefficient matrix M_j. For dense distributions of points, its elements approximate those of the $G_H^{\Psi_j}$ except for a constant factor. Therefore, the properties of the $\{G_H^{\Psi_j}\}_{j\geq 0}$ can also help us gain some intuition about the properties of $\{M_j\}_{j\geq 0}$.The next two subsections are devoted to the study of some of these properties for the cases of two different choices of Ψ: the Standard Hierarchical Basis (HB) and the Linear B-Spline Wavelet Basis (WB).

3.1.1 The Hierarchical Basis

The *standard hierarchical basis* of linear splines Ψ^{HB} is constructed in the following way. We choose V_0 in (3.1.1) to be the space of end-point interpolating linear splines with knots $\{0, \frac{1}{2}, 1\}$. This is the space of the functions that are continuous and linear in the each of the intervals $[0, \frac{1}{2})$ and $[\frac{1}{2}, 1)$. Consider now a "hat" function

$$\psi(x) := \begin{cases} x & : \quad 0 \leq x \leq \frac{1}{2}, \\ 1-x & : \quad \frac{1}{2} \leq x \leq 1. \end{cases} \tag{3.1.12}$$

Defining the translations and contractions of this functions as

$$\psi_{j,k}(x) = \psi(2^jx - k), \tag{3.1.13}$$

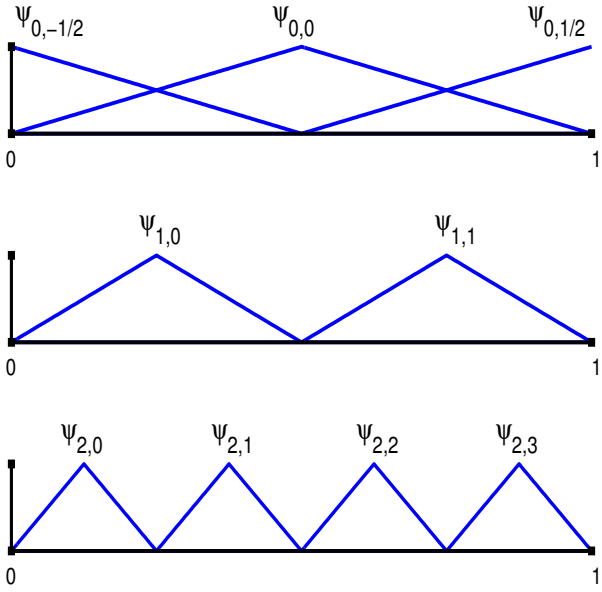


Figure 3.1.1: Standard hierarchical basis.

the basis of B-splines that span V_0 can be denoted by $\Psi_0^{HB} := \left\{\psi_{0,-\frac{1}{2}}|_{[0,1]}, \psi_{0,0}, \psi_{0,\frac{1}{2}}|_{[0,1]}\right\}$.

The whole basis Ψ^{HB} is constructed by translations and contractions of the form

$$\psi_{j,k};\ j > 0,\ k = 0, \ldots, 2^{j-1} \tag{3.1.14}$$

by defining each V_j in the scale (3.1.1) to be spanned by the set

$$\Psi_J^{HB} := \left\{\psi_{j,k} : j \leq J;\ k = 0, \ldots, 2^{j-1} \text{ for } j > 0\ , k \in \left\{-\frac{1}{2}, 0, \frac{1}{2}\right\},\ \text{for } j = 0\right\}. \tag{3.1.15}$$

The first few members of this basis are depicted in Figure 3.1.1. The properties of this basis with respect to H_0^1 are well known in the theory of multilevel preconditioning for PDEs. This basis is uniformly stable (after appropriate rescaling) in $1d$, [150] and unstable in two dimensions [149]. But in our data fitting approach we have to turn our attention to the uniform stability in L_2. And it turns out that this basis is *not* L_2-stable. This result can be considered as a particular case of some deeper theorems [123]. We give now a simple proof that illustrates clearly how the mechanism of instability works.

Theorem 3.1.5. *The standard hierarchical basis is not stable in* $L_2([0,1])$.

Proof:
We are going to prove by providing a counterexample that the lower estimate in equation (3.1.6) cannot be fulfilled for every j by any fixed constant c_1. Consider a

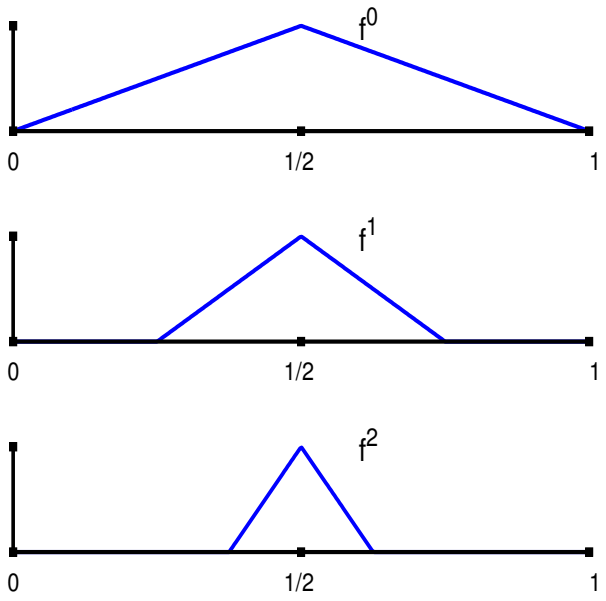


Figure 3.1.2: Sequence of functions with decreasing L_2 energy.

sequence of functions f^j for $j = 0, 1, \ldots$ defined as

$$f^j(x) = \begin{cases} 1 - 2^j + 2^{j+1}x & : \quad \frac{1}{2} - \frac{1}{2^{j+1}} \leq x \leq \frac{1}{2}, \\ 1 + 2^j - 2^{j+1}x & : \quad \frac{1}{2} \leq x \leq \frac{1}{2} + \frac{1}{2^{j+1}}. \end{cases} \tag{3.1.16}$$

These are hat functions centered at the point $x = \frac{1}{2}$, with value $\frac{1}{2}$ there, and with support $\frac{1}{2^j}$, see a few in Figure 3.1.2. One can compute the L_2 norm of each member as

$$\|f^j\|_{L_2}^2 = \frac{1}{3} \cdot \frac{1}{2^j}. \tag{3.1.17}$$

These functions are constructed so that $f^j \in V_j$ for each $j = 0, 1, \ldots$. They have therefore an expansion in terms of the elements of Ψ_j, which we denote by $d_j^T \Psi_j$, where $d_j \in I\!R^{\#\Lambda_j}$. We can take a look at the coefficients of these expansions. f^0 is clearly $\psi_{0,0}$, and hence $d_0 = (0, 1, 0)^T$. f^1 can be constructed by subtracting from f^0 the half of the two hat functions around the point $x = \frac{1}{2}$. Hence $d_1 = (0, 1, 0, -\frac{1}{2}, -\frac{1}{2})^T$. We can likewise construct the following f^j in our sequence by the recursion formula

$$f^j = f^{j-1} - \frac{1}{2}\left(\psi_{j,2^{j-1}} + \psi_{j,2^{j-1}+1}\right). \tag{3.1.18}$$

The result of this mechanism is the following. The coefficient vector of each f^j extends the coefficient vector of f^{j-1} by adding 2^j new components, from which exactly two are not zero and have the value $-\frac{1}{2}$, that is, for each $J \geq 0$

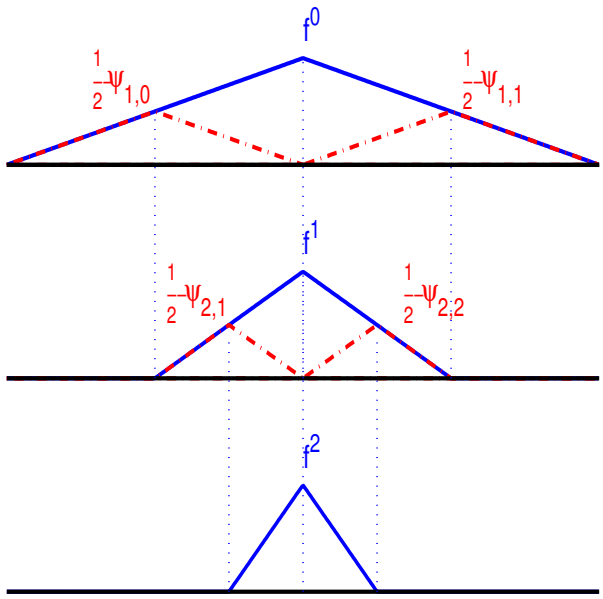


Figure 3.1.3: Recursive construction of the sequence f^j.

$$d_J{}^T = (\overbrace{0,1,0}^{j=0}\ ,\ \overbrace{-\frac{1}{2},-\frac{1}{2}}^{j=1}\ ,\ \overbrace{0,-\frac{1}{2},-\frac{1}{2},0}^{j=2}\ ,\ \dots\ ,\ \overbrace{0,\dots,0,-\frac{1}{2},-\frac{1}{2},0,\dots,0}^{j=J}). \quad (3.1.19)$$

The ℓ_2 norm of each of the coefficient vectors is clearly bigger at each scale j. On the other hand, by (3.1.17) the L_2 norm of the corresponding function decays as $\frac{1}{2^j}$. Therefore the stability condition cannot be met. ∎

Numerical computations show that the condition numbers of the Gramian matrices effectively grow like 2^j. In the two dimensional case, the Gramian matrix with respect to $L_2([0,1]\times[0,1])$ can be written as the tensor product of two Gramian matrices with respect to $L_2([0,1])$ (note that this construction is only valid for L_2 and not for smoother spaces). Consequently, the condition number in two dimensions will grow like 2^{2j}. Note that the condition number is only a very coarse information about the complete spectral properties of a matrix. Let us turn our attention to the whole spectrum of these matrices, as depicted in the Figure 3.1.4 for $j = 1$ to 5. We should note that the explosion of the condition number is due to decreasing minimal eigenvalues, that accumulate around zero.

3.1.2 The Wavelet Linear B-Spline Basis

Wavelets are a special case of a multiresolution basis constructed in the following way. Each subspace V_j in the nested sequence (3.1.1) has an *orthogonal complement*

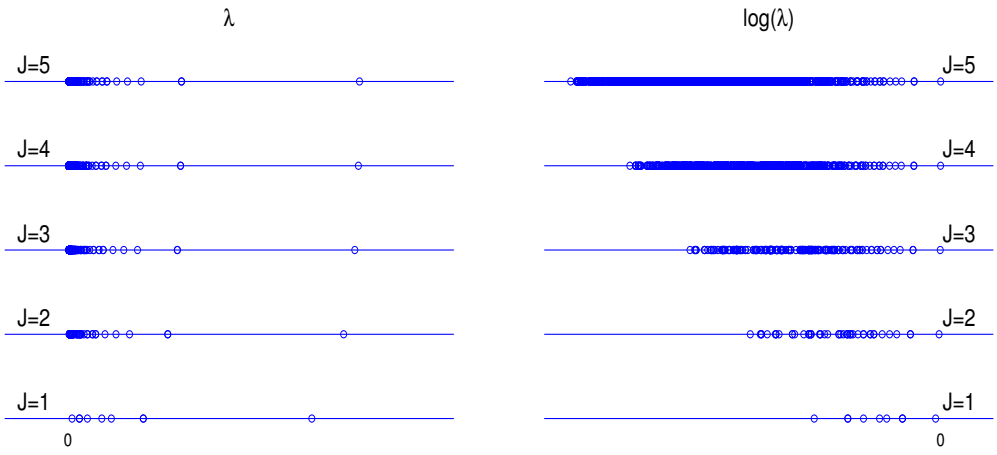


Figure 3.1.4: Eigenvalues of the Gramian matrix $G_{L_2}^{\Psi^{HB}}$ of the standard hierarchical basis in 2d.

in V_{j+1}, which we denote by W_j, such that

$$V_j \oplus W_j = V_{j+1}. \tag{3.1.20}$$

Let S_j be a set spanning W_j. We fix a basis Ψ_0 for V_0 and construct a basis Ψ_j^{WB} of V_j of the form

$$\Psi_j^{WB} = \Psi_0 \bigcup_{j' \geq 0}^{j} S_{j'} \tag{3.1.21}$$

for each $j \geq 1$. The wavelet linear B-spline basis Ψ^{WB} arises when applying this kind of construction to the scale of spaces associated to the hierarchical basis Ψ^{HB} presented in the precedent subsection. Henceforth, Ψ_j^{HB} and Ψ_j^{WB} span the same space for each $j \geq 0$. The first levels are depicted in Figure 3.1.5. We can see in Figure 3.1.6 that wavelets that overlap the boundary are no longer translations of wavelets in the interior of the domain. The requirement of orthogonality between levels enforces the loss of this property.

The different levels are orthogonal w.r.t. L_2, so that the Gramian matrices turn out to be block diagonal. The eigenvalues are those of the individual blocks. In the case of the linear B-splines wavelets in 1d these blocks are tridiagonal matrices, except for a slight perturbation at the boundary, whose eigenvalues are symbolically computable. The Gramian matrix in the two dimensional case is a tensor product of two matrices of the diagonal case. We can see the eigenvalues for levels $j = 1$ to 5 in Figure 3.1.7 and observe how the condition numbers are uniformly bounded.

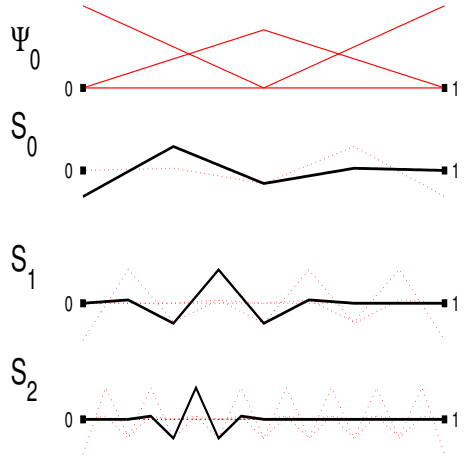


Figure 3.1.5: Wavelets on the interval.

3.2 Nesting Strategy

In Algorithm 2.3.2, one system like (2.2.1) has to be solved at each $j = 1, \ldots, J$. This series of systems are *nested*, that is, the system matrix and the right hand side at a level j contain as restriction the system matrix and the right hand side of the level $j-1$. This naturally rises the question whether and to which extent one can exploit having solved the system on $j-1$ to facilitate solving on level j.

Let us consider the following model problem. For the first level $j = 1$ one has the *inner* system

$$M_1 \, d_1^{(1)} = b_1 \tag{3.2.1}$$

with $d_1^{(1)}, b_1 \in I\!R^{n_1}$, $M_1 \in I\!R^{n_1 \times n_1}$, $n_1 = \#\Lambda_1$. The notation $d_j^{(i)}$ indicates that the vector relates to functions of scale j, having been computed on a system including components up to level i.

Now, the next level $j = 2$ has $n_2 = \#(\Lambda_2 \backslash \Lambda_1)$ new components. The *extended system* arising at this level can be written as an extension of the former,

$$\begin{pmatrix} M_1 & M_{1,2} \\ M_{1,2}^T & M_2 \end{pmatrix} \begin{pmatrix} d_1^{(2)} \\ d_2^{(2)} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \tag{3.2.2}$$

where $d_1^{(2)} \in I\!R^{n_1}, d_2^{(2)}, b_2 \in I\!R^{n_2}, M_{1,2} \in I\!R^{n_1 \times n_2}, M_2 \in I\!R^{n_2 \times n_2}$. The rectangular matrix $M_{1,2}$ represents the coupling between the two levels. We write (3.2.2) in the more compact way

$$M \, d = b, \tag{3.2.3}$$

where $b, d \in I\!R^{\tilde{n}}$, $M_1 \in I\!R^{\tilde{n} \times \tilde{n}}$, $\tilde{n} = n_1 + n_2 = \#\Lambda_2$. When we solve (3.2.3) for d after having solved (3.2.2) for $d_1^{(1)}$, our first attempt could be to simply extend with

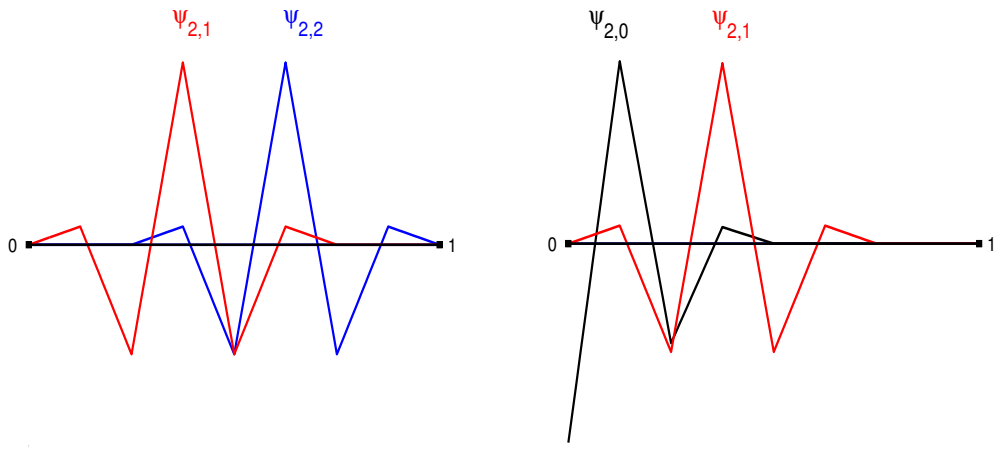


Figure 3.1.6: Interior wavelets and boundary wavelets.

zeros the solution for the first level, and to use the so constructed vector

$$d^0 := \begin{pmatrix} d_1^{(1)} \\ 0 \end{pmatrix} \tag{3.2.4}$$

as starting value for the solution of (3.2.3) in the iterative method.

But this strategy is not guaranteed to be useful. We recall shortly how the Conjugate Gradient Method would proceed to seek for the solution of (3.2.3). Denote by $\{\nu_i\}_{i=1,\dots,\tilde{n}}$ and $\{u_i\}_{i=1,\dots,\tilde{n}}$ the eigenvalues and eigenvectors of M, and by η_i the projection of the initial error on the i-th eigenvector, that is,

$$\eta_i = (d - d^0\,,\, u_i)_{\ell_2(I\!R^{\tilde{n}})},\ i = 1, \dots, \tilde{n}. \tag{3.2.5}$$

The CG creates a series of approximating solutions $\{d^l\}_{l=1,\dots}$ whose distance to the true solution is given by the expression

$$\|d - d^l\|_{\ell_2} = \sum_{i=1}^{\tilde{n}} \eta_i P_l(\nu_i). \tag{3.2.6}$$

P_l is a polynomial of degree l fulfilling $P_l(0) = 1$. This P_l is precisely the one that minimizes the right hand side of (3.2.6) among all the polynomials of degree l meeting $P(0) = 1$. This property is the key of the rapid convergence of CG in comparison to other iterative methods.

Let us explore the structure of the terms η_i in (3.2.6) defined by (3.2.5). Denote by $u|_1$ the vector in $I\!R^{n_1}$ resulting from the extraction of the n_1 first components of $u \in I\!R^{\tilde{n}}$ and by $u|_2$ the vector in $I\!R^{n_2}$ formed by the last n_2 components of u. We can now write (3.2.5) as

$$\begin{aligned} \eta_i &= (d|_1 - d^0|_1\,,\, u_i|_1)_{\ell_2(I\!R^{n_1})} + (d|_2 - d^0|_2\,,\, u_i|_2)_{\ell_2(I\!R^{n_2})} \\ &= (d|_1 - d_1^{(1)}\,,\, u_i|_1)_{\ell_2(I\!R^{n_1})} + (d|_2 - 0\,,\, u_i|_2)_{\ell_2(I\!R^{n_2})} \end{aligned} \tag{3.2.7}$$

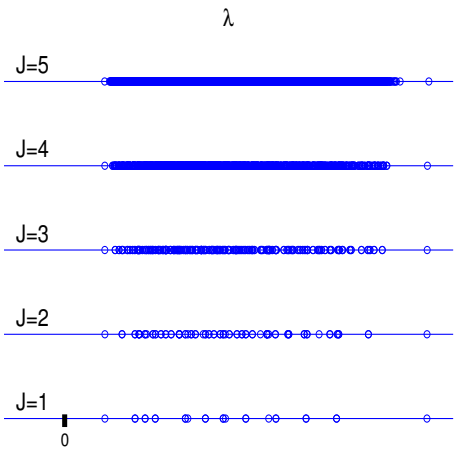


Figure 3.1.7: Eigenvalues of the Gramian matrix $G_{L_2}^{\Psi^{WB}}$ of the linear B-spline wavelet basis in 2d.

This expression helps to clear how the nesting works. If we work with wavelets, the Gramian matrix corresponding to M is a block diagonal matrix. The level-coupling matrix $M_{1,2}$ is zero. The consequence of this fact is twofold.

1. The solution of the inner system is clearly independent of the outer system and hence
$$d|_1 = d_1^{(1)}. \tag{3.2.8}$$

2. The eigenvectors of M are padded versions of the eigenvectors of M_1 and M_2, that is, denoting the eigenvalues and eigenvectors of M_1 by $\{\nu_i^{(1)}\}_{i=1,\ldots,n_1}$ and $\{u_i^{(1)}\}_{i=1,\ldots,n_1}$ those of M_2, and those of M_2 by $\{\nu_i^{(2)}\}_{i=1,\ldots,n_1}$ and $\{u_i^{(2)}\}_{i=1,\ldots,n_2}$, the eigenvectors of M can be written as
$$\begin{aligned} u_i &= \begin{pmatrix} u_i^{(1)} \\ 0 \end{pmatrix}, \; i = 1, \ldots, n_1 \\ u_i &= \begin{pmatrix} 0 \\ u_i^{(2)} \end{pmatrix}, \; i = n_2 + 1, \ldots, \tilde{n} \end{aligned} \tag{3.2.9}$$

We can use the observations leading to (3.2.8) and (3.2.9) to simplify (3.2.7) to

$$\eta_i = (0,\, u_i|_1)_{\ell(\mathbb{R}^{n_1})} + (d|_2\,,\, 0)_{\ell(\mathbb{R}^{n_2})},\; i = 1, \ldots, n_1. \tag{3.2.10}$$

The expression in (3.2.6) can be now written as

$$\|d - d^l\|_{\ell_2} = \sum_{i=n_1+1}^{n_1+n_2} \eta_i P_l(\nu_i), \tag{3.2.11}$$

showing that we only have error contributions stemming from the outer degrees of freedom. The error contributions of the inner degrees of freedom have already been accounted for and will no longer appear.

The former argumentation would be however of no use if we could directly work with the Gramian matrix, as (3.2.3) would be decoupled into two independent systems, and there would be no point in solving both of them together. But in the data fitting application we do not work with the Gramian matrix but with the coefficient matrix in (2.2.1), whose entries only approximate those of the Gramian matrix, and so the $M_{1,2}$ matrix for the wavelet basis is not exactly zero. We actually do have a coupling. Anyway, if this coupling is weak, we can still expect (3.2.8) and (3.2.9) to be approximately fulfilled, and use the former argumentation to expect some kind of improvement the error reduction rate when using this nesting procedure.

In the standard hierarchical basis, this analysis fails. The matrix $M_{1,2}$ is not zero even for the Gramian matrix and we cannot expect to benefit from the nesting strategy up to same extent as when working with the wavelet basis.

3.3 Case Study

In this subsection we illustrate the features described in the above sections with a fitting experiment involving synthetically produced and real data.

In our example we consider the function in the left hand side of Figure 3.3.1, sampled in the points in the right of the figure, and we compute a full grid approximation both for the WB and the HB so that the number of degrees of freedoms is comparable. We are then interested in comparing the performance of a numerical method, as explained in following subsections, working with each of the two bases.

Error Decay . Our algorithm produces a series of Λ_j, one for each resolution scale, and solves equation (2.2.1) at each level. That is, we solve a series of systems

$$M_j d_j = b_j, \quad j = 1, \ldots, J \tag{3.3.1}$$

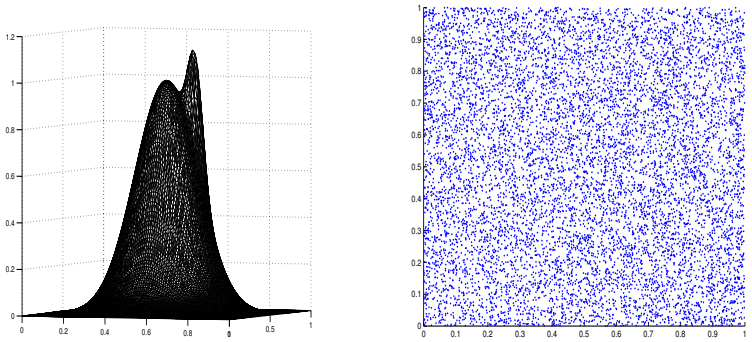


Figure 3.3.1: Testing data.

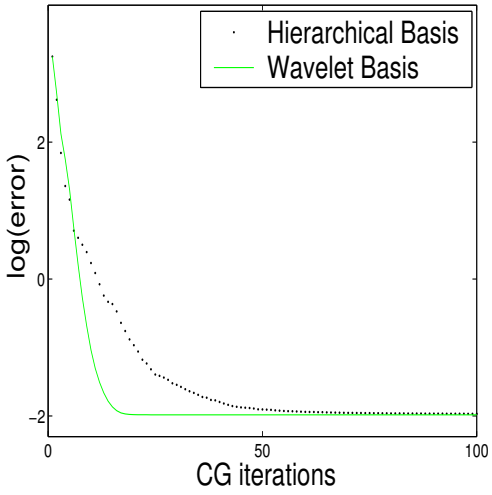


Figure 3.3.2: Error decay rate at the top level J in Λ for WB and HB.

that produces a series of level approximations $\{f_{j_0}, \ldots, f_J\}$. As we employ an iterative method, we generate at each resolution level j a sequence of approximating functions $f_j^{(0)}, f_j^{(1)}, \ldots, f_j^{(k)}, \ldots$, that converge to the f_j formed with the d_j coefficients that exactly solve the corresponding system in (3.3.1).

A crucial question to measure the efficiency of the fitting procedure is the number of iterations we have to perform at each level j to obtain f_j.

In Figure 3.3.2 we see how the error per CG iteration, defined as

$$\left(\sum_{i=1}^{N} \left(z_i - f_J^{(k)}(x_i) \right)^2 \right)^{\frac{1}{2}} \tag{3.3.2}$$

decays with the number k of CG iterations k both in the Wavelet and Hierarchical Bases when computing the solution at the corresponding top level $J = 4$, starting with an initial value of zero.

As expected, we observe that we need less iterations when working with wavelets. Although we are referring here only to the reconstruction of points from Figure 3.3.1, this effect can be generally observed for sets of points of different characteristics. It is worth mentioning that the convergence of the CG procedure or its derivations is very sensitive to scalings, which must be taken into account in any statement about convergence. That is why the present results were made choosing the best decay rates both for the WB and the HB after running a series of tests for each basis.

Nesting .

The Vertical Thresholding Algorithm proposed in Section 2.3.2 creates the sequence $\{\Lambda_j\}_{j=j_0,\ldots}$ level by level by solving the system of equations (3.3.1) for each j.

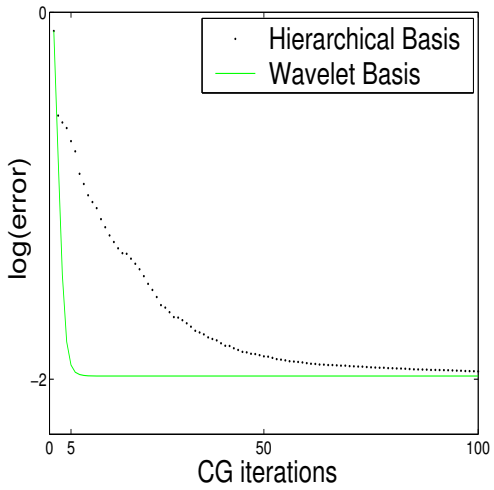


Figure 3.3.3: Comparison of the error decay used nesting on WB and HB.

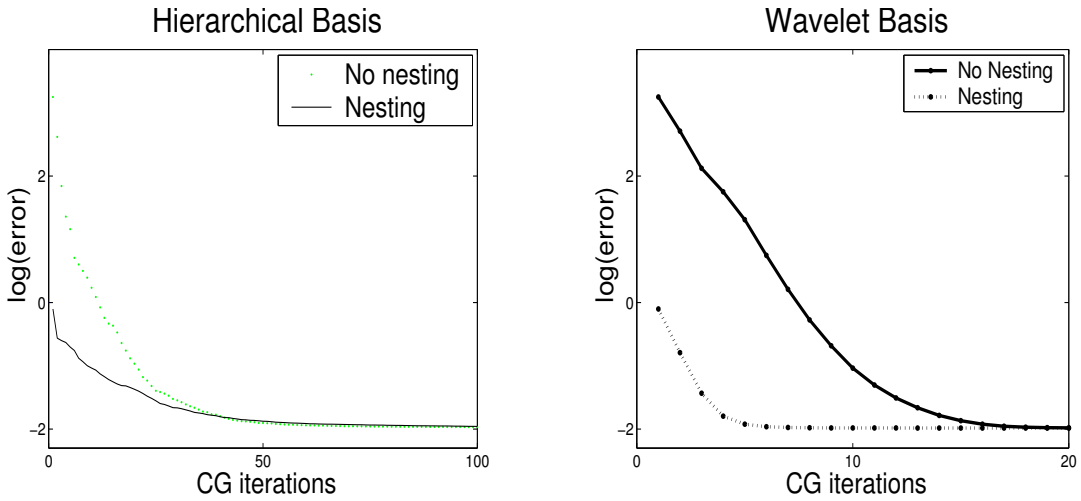


Figure 3.3.4: Error decay by nesting.

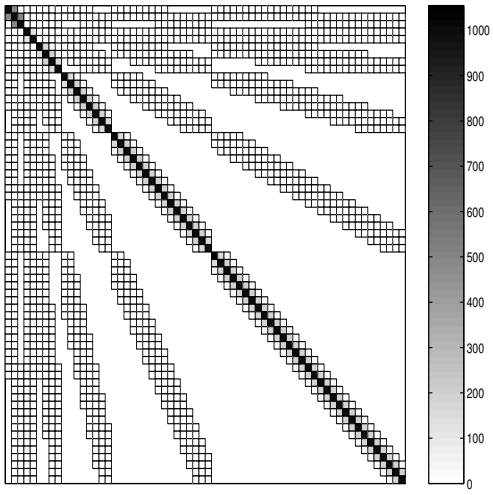


Figure 3.3.5: Typical structure of a coefficient matrix M_j in a wavelet basis. The elements in the fingers that couple wavelets on different levels are not zero like in the Gramian matrix $G_{L_2}^{\Psi^{WB}}$. Nevertheless, they are several orders of magnitude below the elements coupling neighboring wavelets on the same dyadic level. As a result, the matrix is strongly diagonal dominant.

In fact, it needs to solve for the coefficients d_j at each scale to get some information about the properties of the data that allows for a right choice of the set of indexes Λ_{j+1} at the next level. However, we can try to reuse these partial level solutions to accelerate the error decay by proposing the coefficients computed at level j as starting value for the computation of the coefficients at the next level j+1.

If we repeat the experiment of the former section giving this time the solution computed at the previous level instead of zero as starting value for the iterative method, we get the error decay rate given in Figure 3.3.3. We observe that the advantage of wavelets has become more striking.

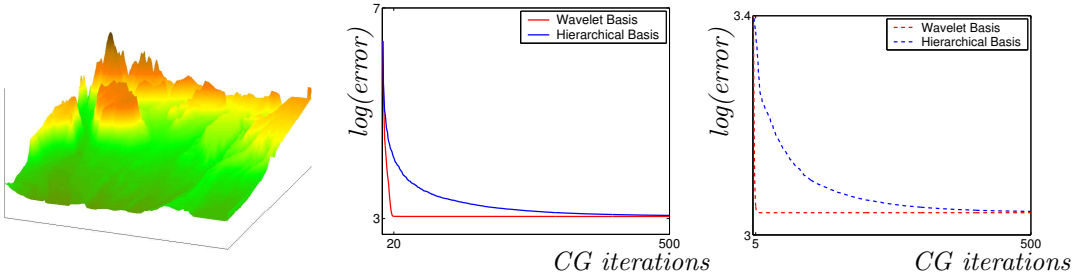
This can be interpreted in terms of the orthogonality of wavelets between levels, which makes them a natural choice for the algorithm we propose, based on level by level growth.

Every time we extend a set Λ_j with new elements corresponding to the next scale to form Λ_{j+1}, the coupling between old and new degrees of freedoms is going to be weak. Looking for instance at the structure of a typical A^TA matrix like in Figure 3.3.5 we observe that off-diagonal blocks, representing interactions between levels, contain small elements in comparison to elements in diagonal blocks, representing interaction between elements on the same level. Because of this weak coupling, the CG acts as if it was looking for fast independent solutions on each scale. This means that the coefficient values computed for the scales up to level j by minimization on the set of indexes Λ_j are not severely affected when minimizing on a larger set Λ_{j+1} that includes new higher frequencies. So, if we feed the CG on Λ_{j+1} with a

starting value $d_{j+1}^{(0)}$ that includes a good guess for the degrees of freedoms in Λ_j, the work corresponding to this subset of Λ_{j+1} has been already done and only minor corrections are needed to cope with the new higher frequencies, that act like a perturbation.

In the Hierarchical Basis all the degrees of freedoms are strongly coupled and stay in the way of each other during the CG process. That can be observed on the left of Figure 3.3.4 comparing the decay error rates obtained with the Hierarchical basis, with and without. We observe that the nesting does not give any help, as attaining the global minimums involves as many iterations in the two cases. On the right of the same figure we see the results for wavelets in the analogous experiment. The nesting does really allow for a reduction of the iterations we need to reach the minimum.

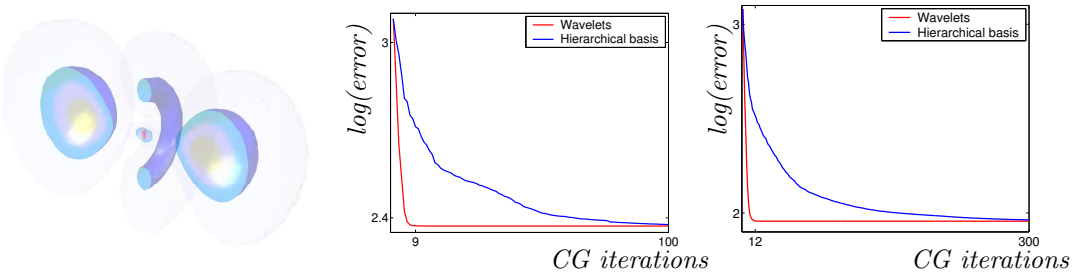
This trend is obviously going to be the more noticeable the higher the maximal resolution level is, as the conditioning of matrices arising from a hierarchical basis ansatz is going to be worse, whereas it remains bounded in the case of wavelets. For instance, we give in Figure 3.3.6 the results of a data fitting experiment with the data depicted on the left of the Figure, stemming from a Digital Elevation Model of the Earth ([85]) and including more than 160000 points. A maximal resolution level $J = 7$ was employed, involving more than 65000 degrees of freedom working with a full grid.



Geophysical data set Without nesting With nesting

Figure 3.3.6: Performance at high resolution data sets.

Also a higher dimensionality of the data enhances this effect: when fitting the $3d$ data on the left plot of Figure 3.3.7, corresponding to the simulation of the spatial probability distribution of the electron in an hydrogen atom residing in a strong magnetic field (courtesy of the project SFB 352 of the German Research Council), the faster convergence of wavelets is already visible at lower resolution levels, as seen in the central and right plots, depicting the error decay with the number of iterations.



Original data $J = 2$ $J = 3$

Figure 3.3.7: Performance on multivariate data.

3.4 Irregular Distribution of Measurements

Our considerations so far have been based on an extrapolation of the good properties of the Gram matrices approximating the experimental coefficient matrices M_j. This extrapolation is obviously not always possible, as the closeness of the entries of both matrices relies on the local distribution of measurements. One can expect that a variation of the density of observations on the domain should have an effect on the spectral properties of the coefficient matrix.

In any case, experiments typically show that the effect of these factors may indeed alter the eigenvalue distributions, but the good properties observed with regular distributions partially remain.

Typical results correspond to those computed with the fitting experiment described in detail in this section, done on the function of the case study from the previous section, plotted on the left of Figure 3.3.1. The function is sampled on three different sampling geometries, each comprising 20000 points, and conceived merely in order to construct a sort of scale of increasing irregularity.

- grid data set: the measurement points are located on a regular grid.
- random data set: the measurement points were randomly generated following a uniform distribution.
- cluster data set: a fraction of the observations is located on a regular grid, while the rest was generated concentrating on four spots (clusters).

Firstly, we explore the behavior of the eigenvalues of the coefficient matrices at increasing different levels: see the results in Figure 3.4.1.

- grid data set: the condition numbers of different levels remain close to the condition number of the Gram matrix, worsening in a fast inappreciable way.
- random data set: At lower levels, the condition number approximates the theoretical value, but for higher levels, where the number of points of the

support of the wavelets gets rarer, the summations of the elements of the coefficient matrix corresponding to high-frequency wavelets differ from the integrals of the Gram matrix.

- cluster data set: The divergence of the spectrum in relation to the those of the Gram matrix starts already at small frequencies: wavelets on these levels have certainly a large support and the number of measurements on it is certainly big, but the irregular distribution across the support prevents the summation from approaching the integral.

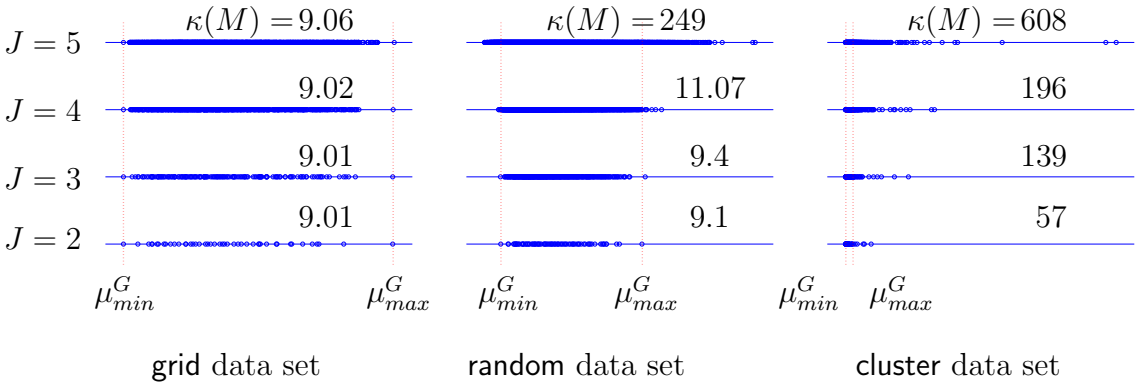
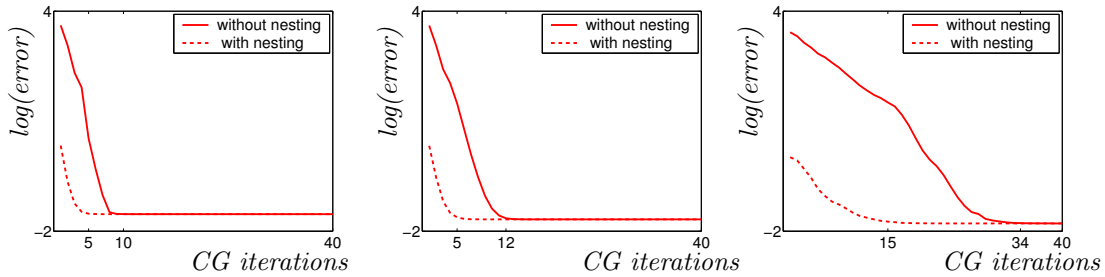


Figure 3.4.1: Spectra at different dyadic levels of the coefficient matrices for different horizontal distributions. Each distribution comprises 20000 points. The dotted lines mark the location of the smallest (μ^G_{mix}) and largest (μ^G_{mix}) eigenvalues of the Gram matrix.

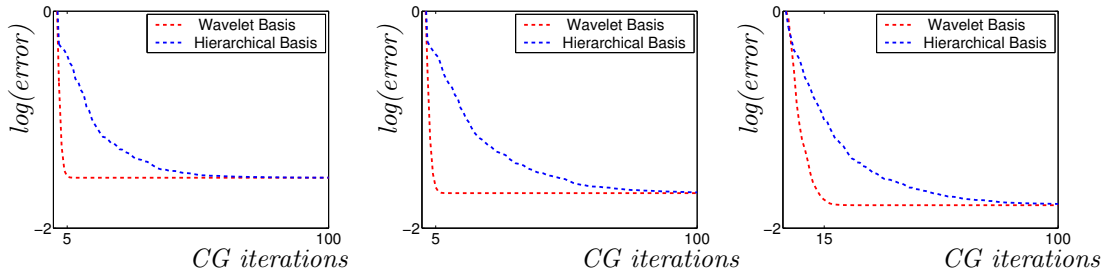
Although the difference in condition numbers could let think that irregular distributions destroy the capacity of wavelets to generate normal equations with good convergence properties, the iteration count shows that it is not the case: In Figure 3.4.2 we see the iteration count at the maximal resolution level $J = 4$. The error decay rate deteriorates in the case of clustered data, but it convergence still occurs in a few iterations (we point out that at level $J = 4$ matrices involve 1089 degrees of freedom). Moreover, the capacity of reducing iterations by a nesting strategy is still available. Still more interesting are the results depicted in Figure 3.4.3, where iteration counts for Wavelet and Hierarchical Basis functions are given. The performance of the Hierarchical Basis does not suffer much under the effect of irregularity, as the shape of the error decay curve remains similar in all three cases. The good performance properties of wavelets relay on their delicate construction, which assures orthogonality between levels. This fine property obviously suffers under discretization. The hierarchical basis displays the opposite side: it acts kind of "stable in its stability". As their functions have a simpler structure and smaller supports, the effect of discretization is weaker.

Nevertheless, in all three cases, the wavelet basis still manages to outperform the hierarchical basis.



grid data set random data set cluster data set

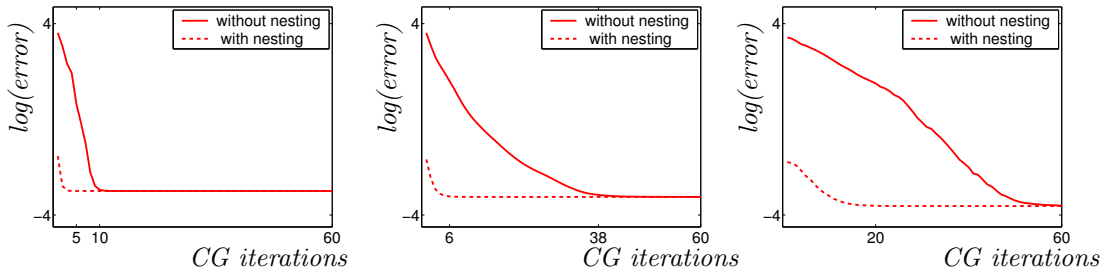
Figure 3.4.2: Error reduction per CG-iteration for the different point distributions referred to in Figure 3.4.1 with and without the nesting strategy. The resolution level is $J = 4$, involving 1089 degrees of freedom.



grid data set random data set cluster data set

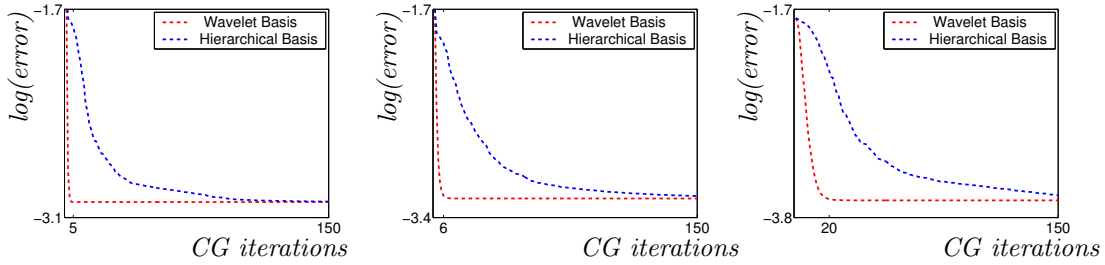
Figure 3.4.3: Error reduction per CG-iteration for the different point distributions referred to in Figure 3.4.1 using a linear wavelet basis and a linear hierarchical basis, in both cases using a nesting strategy. The resolution level is $J = 4$, involving 1089 degrees of freedom.

The results on the next dyadic levels, represented in Figures 3.4.4 and 3.4.5 confirm the trend: in spite of a certain deterioration of convergence properties, the capacity of the wavelets to take profit of the nesting strategy remains, as well as the advantage on the hierarchical basis, in which the effect of the irregular distribution starts to be more noticeable.



grid data set random data set cluster data set

Figure 3.4.4: Error reduction per CG-iteration for the different point distributions referred to in Figure 3.4.1 with and without the nesting strategy. The resolution level is $J = 5$, involving 4212 degrees of freedom.



grid data set random data set cluster data set

Figure 3.4.5: Error reduction per CG-iteration for the different point distributions referred to in Figure 3.4.1 using a linear wavelet basis and a linear hierarchical basis, in both cases using a nesting strategy. The resolution level is $J = 5$, involving 4212 degrees of freedom.

This kind of results is also encountered with sets of irregularly distributed observations typically appearing in actual data analysis scenarios, as those depicted in Figure 3.4.6:

- sea data set: 13361 points. The data represents the depth of a sector of the Caribbean around Puerto Rico. Most observations lie along lines, as it is frequently the case in bathymetrical data sets. Data available at [128].
- factory data set: 19920 points. The cloud of points represents an industrial facility. This set was produced by Prof. Steiger (G.H. Essen) using a *matching* procedure. In this photogrammetric technique, several views of a point are required to determine its tridimensional position. The matching of these several *stereo* images frequently results in an irregular pattern of successfully matched points, as shown in the Figure.
- radar data set: 28800 points, measured as explained in Section 2.3.1.
- glacier data set: 8614 points. This glacier contour data is proposed by Franke among other model sets for fitting of scattered data, [78].

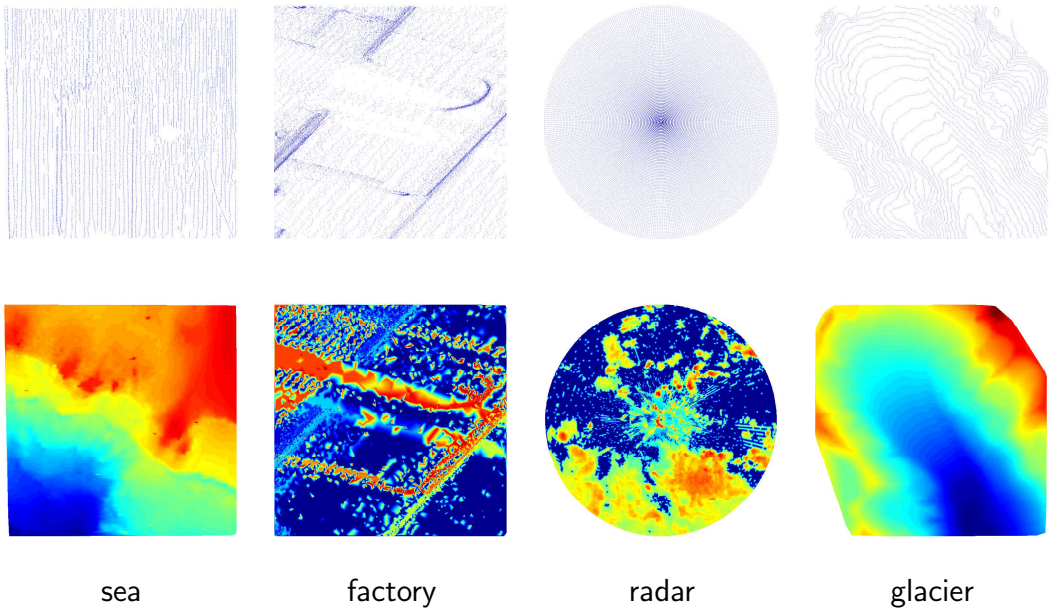


Figure 3.4.6: Data sets with different patterns of irregular data.

Iteration counts are offered on Figure 3.4.7. Again, wavelets appear to offer a more rapid convergence, although (depending on the data distribution) the error decay is less clean: as an effect of the spreading of the spectrum caused by the lack of irregularity, convergence slows down in the proximity of the minimum.

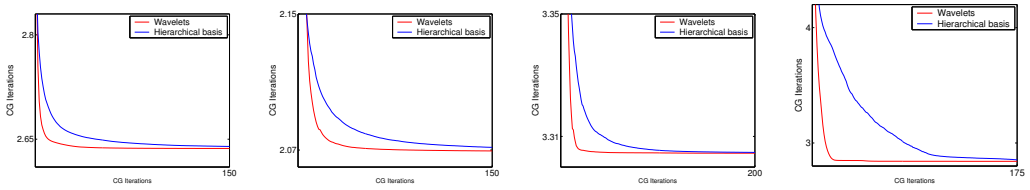


Figure 3.4.7: Error reduction per CG-iteration for the different point distributions referred to in Figure 3.4.6 using a linear wavelet basis and a linear hierarchical basis.

3.5 Construction of Tensor Products

There are several ways to describe bivariate functions with tensor product constructions of monovariate wavelets. The most direct way, as explained above, is to consider the wavelet decomposition of the original space of interest V_J

$$V_J = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \ldots \oplus W_{J-1} \tag{3.5.1}$$

and to work with bases for each of these spaces, that is, the set of scaling functions $\Phi_j := \{\phi_{j_0,k}\}_{k=0,\ldots,2^{j_0}-1+m}$, where m is the order of the spline-wavelets, for the space V_{j_0}, and the set of wavelets $\Psi_j := \{\psi_{j,k}\}_{k=0,\ldots,2^j-1}$ for each W_j. One construction is to develop $V_J \otimes V_J$ using the formula (3.5.1), yielding

$$V_J \otimes V_j = (V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \ldots \oplus W_{J-1}) \otimes (V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \ldots \oplus W_{J-1})\,. \tag{3.5.2}$$

Taking tensor products of the bases on each of the summands in (3.5.2) gives rise to the *anisotropic basis* , which with the shorthand

$$\Phi_{j_x} \otimes \Phi_{j_y} := \left\{\phi_{j_x,k_x}(x) \cdot \phi_{j_y,k_y}(y)\right\}_{k_x=0,\ldots,2^{j_0}-1+m;\; k_y=0,\ldots,2^{j_0}-1+m} \tag{3.5.3}$$

and similarly for the other combinations of Φ_j and Ψ_j, can be written as

$$\Phi_{j_0} \otimes \Phi_{j_0} \bigcup_{\substack{j_x=j_0 \\ j_0 \le j_y \le J-1}} \Phi_{j_x} \otimes \Psi_{j_y} \bigcup_{\substack{j_0 \le j_x \le J-1 \\ j_y=j_0}} \Psi_{j_x} \otimes \Phi_{j_y} \bigcup_{\substack{j_0 \le j_x \le J-1 \\ j_0 \le j_y \le J-1}} \Psi_{j_x} \otimes \Psi_{j_y} \tag{3.5.4}$$

Thus, in this construction, a basis element of $V_J \otimes V_J$ consist of the tensor product of two functions of possibly different scales. This difference can range between 0 and $J - j_0$, and consequently, bivariate functions whose support has a rather elongated shape are included in this basis. This kind of functions might not be desirable when performing local refinements.

In an alternative construction, we can consider the spline space V_J just as the sum of the spline space on the previous dyadic level and its orthogonal complement

spanned by wavelets $V_J = V_{J-1} \oplus W_{J-1}$ and write

$$\begin{aligned} V_J \otimes V_J &= (V_{J-1} \oplus W_{J-1}) \otimes (V_{J-1} \oplus W_{J-1}) && (3.5.5)\\ &= (V_{J-1} \otimes V_{J-1}) \oplus (V_{J-1} \otimes W_{J-1}) \oplus && (3.5.6)\\ &\quad \oplus (W_{J-1} \otimes V_{J-1}) \oplus (W_{J-1} \otimes W_{J-1})\,. && (3.5.7) \end{aligned}$$

Now, iterating on $(V_{J-1} \otimes V_{J-1})$ back to level j_0 we find the following *isotropic decomposition* of $V_J \otimes V_J$

$$\begin{aligned} V_J \otimes V_J &= (V_{j_0} \otimes V_{j_0}) \oplus && (3.5.8)\\ &\quad (V_{j_0} \otimes W_{j_0}) \oplus (W_{j_0} \otimes V_{j_0}) \oplus (W_{j_0} \otimes W_{j_0}) && (3.5.9)\\ &\quad \oplus \ldots \oplus && (3.5.10)\\ &\quad (V_{J-1} \otimes W_{J-1}) \oplus (W_{J-1} \otimes V_{J-1}) \oplus (W_{J-1} \otimes W_{J-1})\,. && (3.5.11) \end{aligned}$$

Recalling (3.5.3), we can write the full isotropic basis as

$$\Phi_{j_0} \otimes \Phi_{j_0} \bigcup_{j_0 \le j \le J-1} \Phi_j \otimes \Psi_j \bigcup_{j_0 \le j \le J-1} \Psi_j \otimes \Phi_j \bigcup_{j_0 \le j \le J-1} \Psi_j \otimes \Psi_j. \tag{3.5.12}$$

As each term is orthogonal to all the others, the Gram matrix of this basis is merely a diagonal concatenation of the Gram matrices of each term, which in turn is the Kronecker product of the Gram matrices of the two factors.

Now, we introduce a notation simplification: in customary compact notation for wavelet tensor product constructions, the multiindex λ that identifies an individual wavelet is to be understood as $\lambda := (j, \mathbf{k}, \beta)$, with $\mathbf{k} = (k_x, k_y)$ and $\mathbf{e} = (e_x, e_y)$, so that

$$\psi_{j,\mathbf{k},\mathbf{e}}(x, y) = \psi^{e_x}_{j,k_x}(x)\psi^{e_y}_{j,k_y}(y). \tag{3.5.13}$$

where the superindex distinguishes between scaling functions and wavelets: $\psi^0_{j,k_x}(x) = \phi_{j,k_x}(x)$ and $\psi^1_{j,k_x}(x) = \psi_{j,k_x}(x)$. With this motivation, we will write $\Psi^0_j = \Phi_j$ and $\Psi^1_j = \Psi_j$. We further simplify the notation defining $G^e_j := G^{\Psi^e_j}$. Now, the Gram matrix of the isotropic basis (3.5.12) of the space $V_j \otimes V_j$ is block diagonal, with the following set of blocks:

$$G^0_{j_0} \otimes G^0_{j_0} \bigcup_{j_0 \le j \le J-1} G^0_j \otimes G^1_j \bigcup_{j_0 \le j \le J-1} G^1_j \otimes G^0_j \bigcup_{j_0 \le j \le J-1} G^1_j \otimes G^1_j. \tag{3.5.14}$$

The spectrum of the Gram matrix in isotropic constructions is therefore

$$\mu^{j_0,0} \otimes \mu^{j_0,0} \bigcup_{j_0 \le j \le J-1} \mu^{j,0} \otimes \mu^{j,1} \bigcup_{j_0 \le j \le J-1} \mu^{j,1} \otimes \mu^{j,0} \bigcup_{j_0 \le j \le J-1} \mu^{j,1} \otimes \mu^{j,1}, \tag{3.5.15}$$

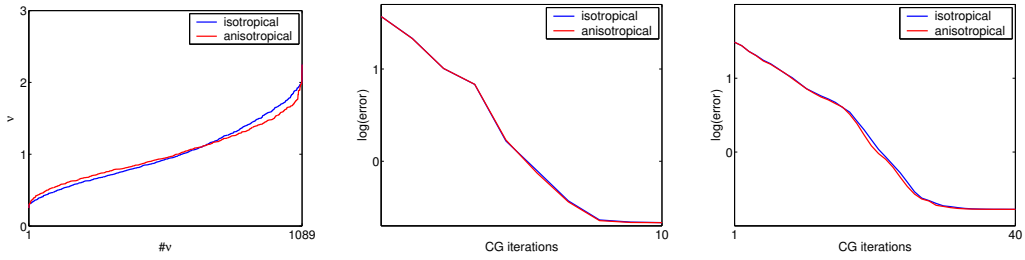
where $\mu^{j,0} = \{\mu^{j,0}_l\}_{l=1,\ldots,2^j+m}$ are the eigenvalues of G^0_j and $\mu^{j,1} = \{\mu^{j,1}_l\}_{l=1,\ldots,2^j}$ those of G^1_j.

The condition number of the Gramian matrices in the bivariate case is the same, as both the minimal and the maximal eigenvalue appear in the block $G^0_{j_0} \otimes G^0_{j_0}$ representing the scaling functions on the coarsest level, common to both isotropic and anisotropic constructions.

Further, there is a large overlapping of eigenvalues, corresponding to the blocks $G^1_j \otimes G^1_j$, also common to both constructions (and which carry the majority of eigenvalues). Moreover, $\mu^{j,0}$ and $\mu^{j,1}$ are similar values, as the G^0_j and G^1_j have similar structures.

The spectra of both matrices are depicted on the right plot of Figure 3.5.1. In practice, this all translates into very similar convergence properties, as we can see in the central and right plots of Figure 3.5.1, where the error decays of both tensor product constructions are hardly distinguishable from each other, both for a gridded and clustered data set. Still in the case of the irregular distributions from Figure 3.4.6 both constructions yield remarkably similar spectral structures, as we can observe in Figure 3.5.2. As explained in Section 3.4, the eigenvalues spread out in presence of scattered data, which can blow up the spectral condition number while the bulk of eigenvalues remains clustered in their original environment. The histograms show how this spreading has very similar features in both bases: both data sets glacier and sea maintain their spectral structures close to the one of the Gramian, and the same holds for factory and sea. The spreading occurs in both tensorial constructions.

Note that this can lead to very different condition numbers of the Gramian matrices in the two bivariate bases for one data set, on account on one isolated eigenvalue, but for convergence purposes it makes no real difference, as the eigenvalue density shows a similar profile in both cases.



Eigenvalues of the Gram Matrix in both basis

Error decay for gridded data

Error decay for clustered data

Figure 3.5.1: Eigenvalues of the Gram matrix with different tensor product constructions.

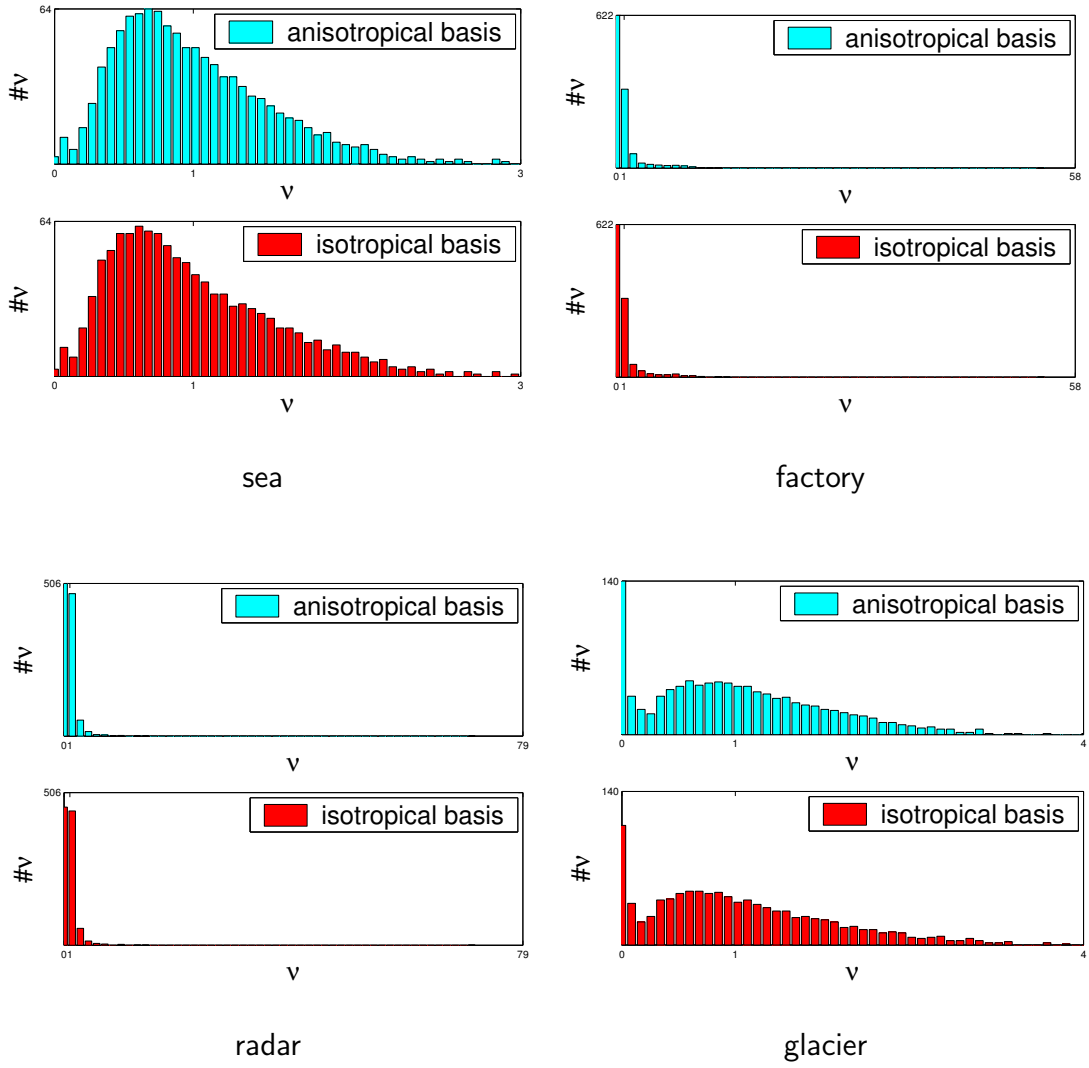


Figure 3.5.2: Eigenvalue histograms of the coefficient matrices for the different point distributions described in Figure 3.4.6, computed with an isotropic and an anisotropic basis of wavelets. For factory and radar, isolated eigenvalues induce the spectrum to spread, but the bulk of the eigenvalues is still concentrated in the same area as in sea and glacier. This spectral behavior is in accordance with the ones of the Gramian matrices for both isotropic and anisotropic constructions.

Chapter 4

Regularization

4.1 Smoothing with Wavelets

A classical way to extend least squares fitting methods to force them to produce smooth surfaces is to add a smoothing term in terms of a Sobolev norm to the functional

$$J(f) = \sum_{i=1}^{N} (z_i - f(x_i))^2 \,, \tag{4.1.1}$$

see e.g. [62] for general splines, [134] for splines with variable knots, [83] for multiscale splines, [94] for splines on triangulations, or [82] in a wavelet reformulation of the spline problem. This yields a cost functional of the form

$$J_\nu(f) = \sum_{i=1}^{N} (z_i - f(x_i))^2 + \nu \|f\|_Y^2 \tag{4.1.2}$$

where $\nu > 0$ balances the fidelity of the data and the smoothness requirement, and Y is usually chosen as the Sobolev space H^1 or H^2.

In this chapter we use the fact that the collection $\{\psi_\lambda : \lambda \in I\!I\}$ constitutes a *Riesz basis* for $L_2(\Omega)$ and, moreover, one has *norm equivalences* for functions in Sobolev spaces $H^\alpha = H^\alpha(\Omega)$ (or even more general in Besov spaces [61]) in the range $\alpha \in [0, \gamma)$ of the form

$$\| \sum_{\lambda=(j,\mathbf{k},\mathbf{e})\in I\!I} d_\lambda \, \psi_\lambda \|_{H^\alpha(\Omega)}^2 \;\sim\; \sum_{j \geq j_0} 2^{2\alpha j} \sum_{\mathbf{k},\mathbf{e}} |d_{j,\mathbf{k},\mathbf{e}}|^2. \tag{4.1.3}$$

The parameter γ depends on the smoothness of the wavelet family which is, for instance, $\gamma = \frac{3}{2}$ for piecewise linear wavelets and accordingly higher for smoother wavelets.

In view of (4.1.3), the wavelet formulation allows to represent functions in Sobolev norms in terms of sequence norms of weighted coefficients from their wavelet expansions. Accordingly, replacing $\|\cdot\|_Y$ by the corresponding sequence norm on the

right hand side of (4.1.3), the minimization of the resulting functional is equivalent to the solution of the normal equations

$$(A^T A + \nu R)d = A^T z. \tag{4.1.4}$$

Here A is shorthand for A_{Λ_j}, j being the actual level of computation, and the diagonal matrix R has the structure

$$R = \begin{pmatrix} 2^{2\alpha j_0} I & & \\ & \ddots & \\ & & 2^{2\alpha j} I \end{pmatrix}. \tag{4.1.5}$$

This illustrates some interesting properties of the representation of the smoothing term in the wavelet context (4.1.3):

(I) The different dyadic scales are decoupled. This is interesting in two ways.

  (i) It keeps the weak decoupling between levels of the matrix $A^T A$, which is explicitly used to speed up the efficient numerical solution of the normal equations as seen in Chapter 3.

  (ii) It gives an insight into the way regularization works. In fact, the effect of the regularization term boils down to a penalization of the higher frequencies. As ν controls the balance between fidelity and regularization, α controls the relative penalization across scales.

(II) One has easy access to the entire scale of fractional Sobolev spaces $\{H^\alpha\}_{\alpha>0}$ which reduces to a simple diagonal scaling in the wavelet framework.

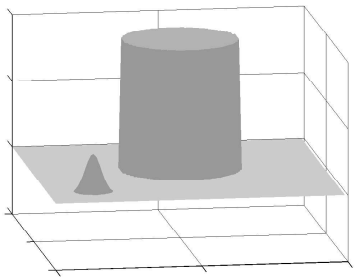
(III) The formulation is independent of the wavelet family. Only the smoothness of the family imposes a limit on the upper bound γ of α for which (4.1.3) holds.

We illustrate an example of the application of property (II) when fitting the data in Figure 4.1.1. We freeze ν and perform the regularized fitting for several fractional Sobolev spaces ranging between H^1 and H^2. One can observe in the results in Figure 4.1.2 a smooth transition between the hardly controlled oscillations at the borders of the plateau in the H^1 case and their over–smoothing in the H^2 case.

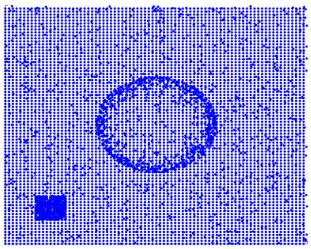
4.2 Cross Validation

The formulation of (4.1.2) opens the question of the choice of the parameter ν controlling the balance between approximation quality and smoothness of the fitting function, and of the space X, that is, of the α in the scale H^α.

Different ways have been proposed and currently used, see for example [68, 88, 141] for surveys of several methods. We will now concentrate in one of the most



Original function



Horizontal distribution of measurements

Figure 4.1.1: Example data set: Irregular sampling of a function showing sharp features.

popular methodologies: the use of some leave-one-out criterion, leading to the *Ordinary Cross Validation (OCV)* method, or to the less expensive *Generalized Cross Validation method (GCV)*, see e.g. [142].

The general concept in this kind of approach is that a good model should predict any new datum. The choice of the regularization parameters should generate a model that succeeds to predict all the data points.

To this end, one denotes for each $\ell = 1, \ldots, N$ by $f_\nu^{[\ell]}$ the approximation to the whole set of data with exception of the point x_ℓ, with respect to a regularizing parameter ν, that is:

$$f_\nu^{[\ell]} \text{ minimizes } \sum_{i \neq \ell}^{N} (z_i - f(x_i))^2 + \nu \|f\|_X^p. \tag{4.2.1}$$

The *Ordinary Cross Validation* merit figure for ν is then defined as

$$OCV(\nu) = \sum_{\ell=1}^{N} \left(z_\ell - f_\nu^{[\ell]}(x_\ell)\right)^2. \tag{4.2.2}$$

The functional illustrates the idea behind these methods, namely to find the regularization parameter leading to the reconstruction that is least affected by any single data point. In any case, the form of (4.2.2) is not suited for computational use. The direct application of this formula would involve the computation of $f_\nu^{[\ell]}$ for every ℓ in the cloud of points and for every ν which we would test during the minimization of (4.2.2). Fortunately the computations can be strongly simplified making use of the so called *leave-one-out Lemma* stated by Craven and Wahba ([41]) in the framework of regression with smoothing splines, which we can formulate for our case. The use of the leave-one-out Lemma leads immediately to the following OCV identity:

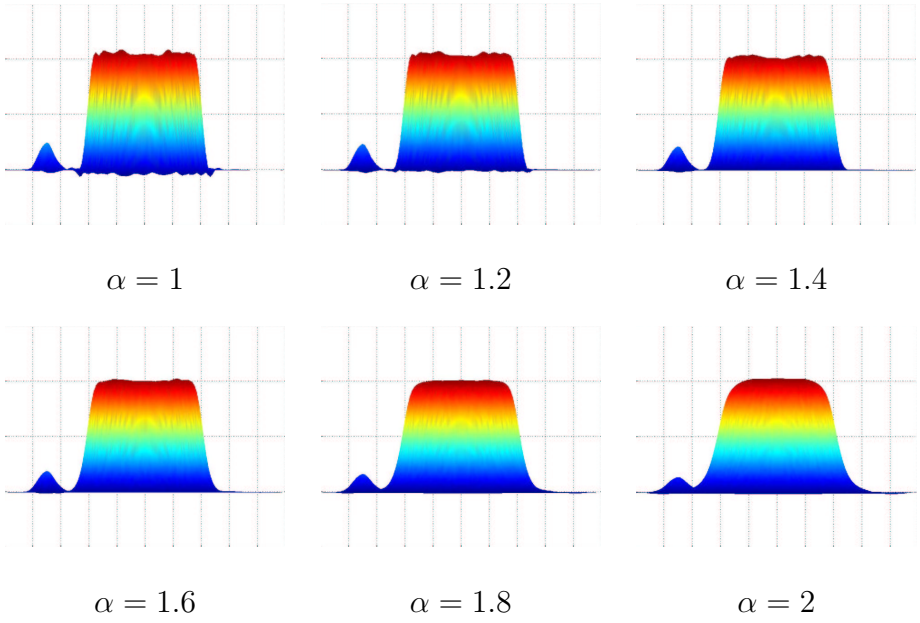


Figure 4.1.2: Smoothing of data in Figure 4.1.1 (cross sections) for $\nu = 100$ and different choices of α for $Y = H^\alpha$ in (4.1.2).

Theorem 4.2.1.

$$\sum_{\ell=1}^{N} \left(z_\ell - f_\nu^{[\ell]}(x_\ell)\right)^2 \equiv OCV(\nu) = \frac{1}{N}\sum_{i=1}^{N} \frac{(z_i - f_\nu(x_i))^2}{(1-h_{i,i})^2} \tag{4.2.3}$$

where f_ν is the function whose wavelet expansion satisfyes (4.1.4). In this expression we get rid of the $f^{[\ell]}$ terms, paying the price of having to compute the diagonal elements of the *influence matrix* defined in (4.2.5) below. The computation cost would still be unaffordable. This motivates the construction of the *Generalized Cross Validation method*, in which one takes the ν that minimizes the GCV functional

$$GCV(\nu) = \frac{1}{N}\frac{\sum_{i=1}^{N}(z_i - f_\nu(x_i))^2}{\sum_{i=1}^{N}(1-h_{i,i})^2} = \frac{\|(I - H(\nu))z\|^2}{(\mathrm{tr}(I - H(\nu)))^2} \tag{4.2.4}$$

where $H \in I\!R^{N\times N}$ with entries $h_{i,j}$ is the *influence matrix* of the problem, defined as

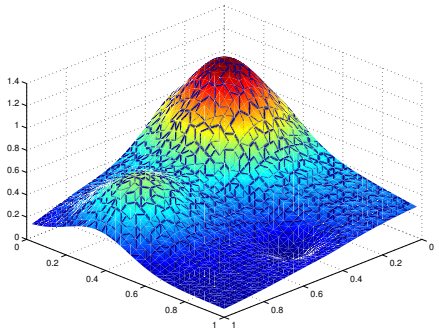
$$H := A\left(A^T A + \nu D\right)^{-1} A^T \tag{4.2.5}$$

and f_ν is the solution obtained by solving (4.1.4) for d with a given ν and letting $f_\nu = \sum_{\lambda\in\Lambda} d_\lambda \psi_\lambda$. One should note the intensive computational effort required by this approach, as a linear system has to be solved each time the GCV of a ν is

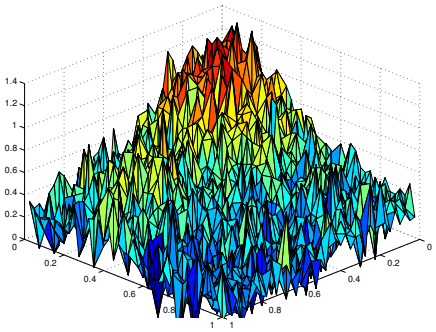
computed. Moreover, a real computation of the trace of the influence matrix for each ν by computing the individual elements would be unfeasible.

Fortunately, the trace can be stochastically estimated in a relatively cheap way as $\mathrm{tr}(H(\nu)) = u^T H(\nu) u$, u being a random vector with entries 1 and -1, see [98].

Statistical meaning of General Cross Validation As explained by Wahba in [143], the GCV is not only easier to compute. It also shows important theoretical properties that one could not prove with ordinary cross validation, although experiments show that the results attained by the two methods are very close. In particular, one can prove the "weak cross validation theorem", see [140], that states the asymptotic optimality of the GCV in presence of data corrupted by white noise. The validity of this theory developed for splines can be translated *verbatim* for the wavelets employed here. As an example, consider the data set of Figure 4.2.1. The left plot is the popular Franke function, see [77], on which we take a discrete regular sampling adding white noise to obtain the right plot.



Original function.



Original function corrupted by white noise

Figure 4.2.1: The Franke function and a discrete sampling corrupted by white noise.

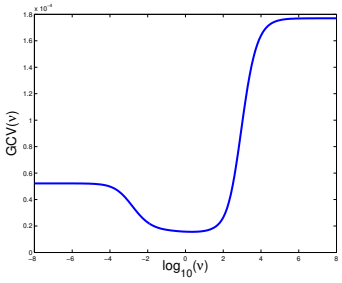
The computation of $GCV(\nu)$ in the case $\alpha = 2$ gives the left plot of Figure 4.2.2, where one can distinguish a clear minimum. Moreover, the shape of this curve is interestingly similar to the one of the curve on the middle, which represents the ℓ_2 error of the approximation produced with a smoothing parameter ν, that is, the quantity $\left(\sum_{i=1}^{N} (f_\nu(x_i) - f(x_i))^2\right)^{\frac{1}{2}}$, where f stands for the original Franke function without noise. This plot would be obviously not available in a real denoising experiment, where the original function is not known and is given here as reference function.

The structures of the two plots is the same, that is, a first flat region for small values ν, where regularization is inactive, a transition area, and a second flat region for large values of ν, where regularization has completely smoothed the resolution.

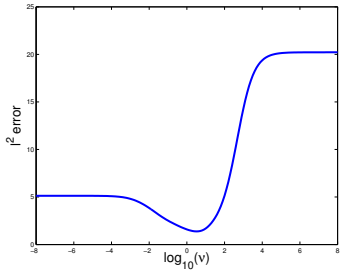
But the coincidence is not only qualitative in these coarse features as the two transition areas are quite similar, with the order of magnitude of the two minima being similar.

Several remarks are in order in view of these plots:

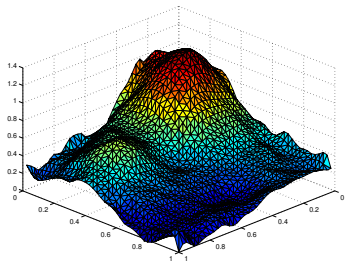
- This concordance visualizes the suitability of the GCV functional as merit figure.
- The minimum in the GCV is flatter than the minimum of the error plot. The consequence is a high parameter sensibility. See Figure 4.2.3 to appreciate how the reconstruction quality decreases away from the detected minimum.



GCV functional for different values of ν



ℓ_2 error of the reconstruction f_ν for different values of ν

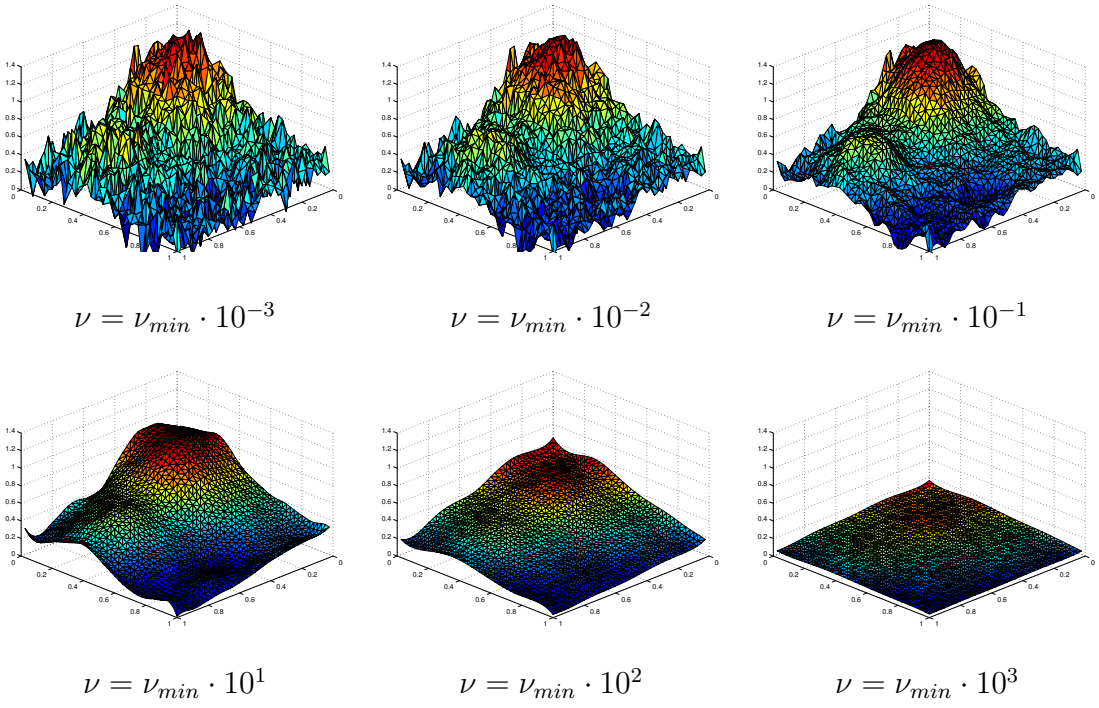


Reconstruction at the ν that minimizes $GCV(\nu)$.

Figure 4.2.2: Denoising corrupted Franke data set from Figure 4.2.1 using the GCV.

Obviously, a further question is the influence of the regularization space. This can be easily analyzed in the wavelet framework and we illustrate it by computation of the GCV using different choices of α. A few reconstructions can be seen in Figure 4.2.4. One can see that the correct choice of the space is a capital issue in order to assure reconstruction quality, as we will discuss thoroughly in Section 4.3. In this constructed case, a rough interpretation of these results is possible: as the noise is located at the highest level of frequencies, spaces H^α with larger α that penalize these frequencies stronger, are going to work better. In any case, the GCV attains to locate a ν providing a reconstruction close to the best possible at each α.

Overfitting Moreover, although the original application of this technique appears in a denoising scenario [143], its interpretation as leave-one-out balancing method provides a heuristics to construct a merit function for the parameter ν which can work in other data fitting situations requiring some amount of regularization. For instance, this occurs for data sets with a highly varying point distribution or with holes in the domain, which typically produce overfitting artifacts. To describe this,



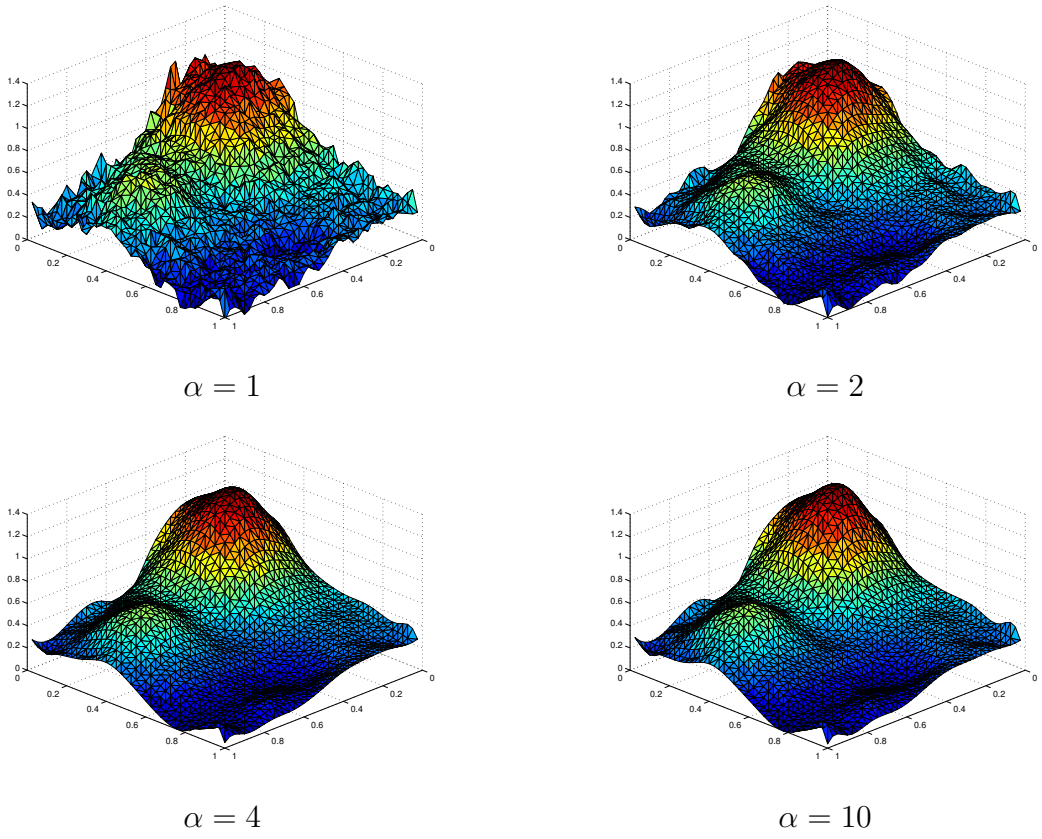
$\nu = \nu_{min} \cdot 10^{-3}$ $\nu = \nu_{min} \cdot 10^{-2}$ $\nu = \nu_{min} \cdot 10^{-1}$

$\nu = \nu_{min} \cdot 10^{1}$ $\nu = \nu_{min} \cdot 10^{2}$ $\nu = \nu_{min} \cdot 10^{3}$

Figure 4.2.3: Reconstruction quality around the GCV minimum for $\alpha = 2$.

consider the data set in Figure 4.2.5. The area in the lower right corner is not densely populated, and a merely least–squares driven reconstruction would produce large oscillations in this area in order to minimize the distance of the surface to the surrounding points, which will act as a leverage.

We see in the reconstruction computed with cubic spline wavelets for three values of ν in Figure 4.2.6 that the GCV criterion provides some kind of order of magnitude information, as the minimal value $\nu = \nu_{\min}$ succeeds to reproduce the features of the original data without artifacts. Away from this minimum, the value $\nu = (50)^{-1}\,\nu_{\min}$ provides too weak a regularization as it does not eliminate the oscillation that one can expect for the poorly populated area. We further illustrate this behavior in Table 4.2.2, where we have listed in the first row the approximation errors $\left(\sum_{i=1}^{N}(z_i - f(x_i))^2\right)^{1/2}$ for different values of ν. In this case of synthetically generated data, we can also compare the approximation error of the reconstructions with the true approximation error. In the second and third row, we have listed the sum of the absolute values of all wavelet coefficients with respect to level 1 and level 2, respectively. We see that the reconstruction for $\nu = (50)^{-1}\,\nu_{\min}$ slightly overestimates the ℓ_2 norm of the wavelet coefficients on level $j = 2$. The opposite situation occurs for $\nu = 50\,\nu_{\min}$ which is too strong a regularization. Wavelets on level $j = 2$ are underestimated, as one can see from the third row in Table 4.2.2, yielding that high-frequency details are not reconstructed properly. In fact, the height of the



$\alpha = 1$ $\alpha = 2$

$\alpha = 4$ $\alpha = 10$

Figure 4.2.4: Denoising of the corrupted Franke data from Figure 4.2.1 by the GCV method for different choices of α.

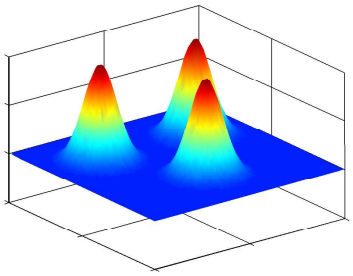
peaks has been reduced (see the cross section of the different reconstructions in Figure 4.2.7) and in flat areas, oscillations of frequency $j = 1$ are not compensated with contributions of higher frequency. Correspondingly, this reconstruction yields a larger approximation error.

We give a further example, this time using linear spline wavelets, in Figure 4.2.8, corresponding to sea floor elevation data from Puerto Rico [128]. This geophysical data set includes a strong irregularity in the horizontal distribution of measurements, as seen in the graphic on the right. In fact, clusters, lines and holes are present.

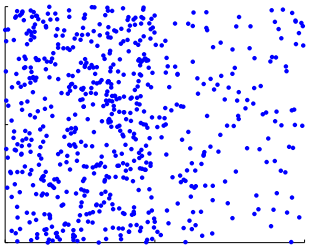
Again, the GCV succeeds to provide a good orientation value for the order of magnitude of the regularizing parameter ν. Compare in Figure 4.2.9 the spiky reconstruction attained with $\nu = (100)^{-1}\nu_{\min}$ on the left, and the over-smoothed surface generated with $\nu = 200\nu_{\min}$ with the reconstruction provided by the value $\nu = \nu_{\min}$. Table 4.2.3 displays the same trends as in the previous example. Again, the over-smoothed surface presents a higher approximation error, and the poorly regularized reconstruction differ from the others at higher frequencies; compare the ℓ_2 norm of the wavelet coefficients at levels $j = 3$ and $j = 4$ for the three regularizations.

α	$\nu_{min} \cdot 10^{-3}$	$\nu_{min} \cdot 10^{-2}$	$\nu_{min} \cdot 10^{-1}$	ν_{min}	$\nu_{min} \cdot 10$	$\nu_{min} \cdot 10^{2}$	$\nu_{min} \cdot 10^{3}$
1	5.069	4.712	3.526	2.140	4.534	13.323	19.156
2	4.425	3.146	2.008	1.378	2.816	9.294	17.794
3	2.787	2.120	1.508	1.170	2.040	6.593	15.979
4	2.142	1.666	1.313	1.087	1.688	5.171	14.341
5	1.770	1.544	1.210	1.070	1.344	3.601	11.404
6	1.670	1.414	1.108	1.080	1.223	2.865	9.378
7	1.540	1.208	1.069	1.097	1.226	2.872	9.398
8	1.342	1.081	1.089	1.102	1.227	2.878	9.4195
9	1.150	1.076	1.101	1.104	1.220	2.885	9.440
10	1.070	1.096	1.105	1.104	1.230	2.892	9.460

Table 4.2.1: ℓ_2 error of the reconstruction for different values of ν surrounding the minimum ν_{min} of the GCV for different choices of α.



Data: Gaussian peaks.



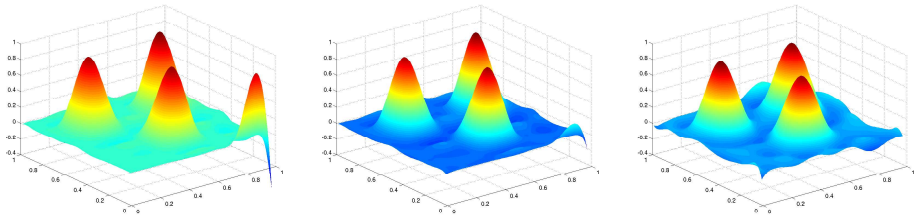
Sampling points.

Figure 4.2.5: Synthetic data with two zones of different density distributions.

4.3 A Multilevel Version of Generalized Cross–Validation

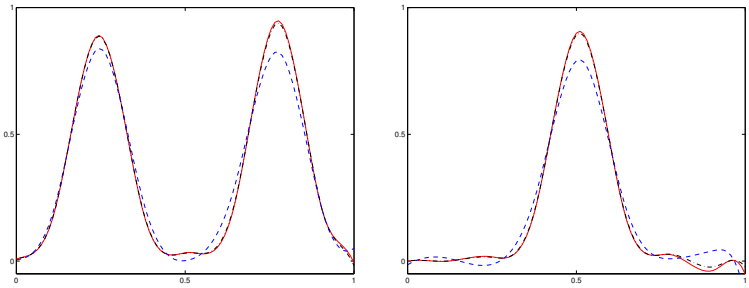
The main idea presented in this Section is the simultaneous use of the GCV with the hierarchical growth of the wavelet tree as the levels become higher. The basis for this is the observation made above that the wavelet representation penalizes the different dyadic levels separately with the same weight $2^{2\alpha j}$. Instead, we propose the following approach. In view of the form of (4.1.5), we prescribe a diagonal penalizing matrix with the same values for all entries of each level. Instead of the particular weights $2^{2\alpha j}$, we propose here to use some quantities $\nu_j > 0$, which are to be computed independently following a GCV criterion. That is, at each level j the normal equations attain the form

$$(A_{\Lambda_j}^T A_{\Lambda_j} + R_j)d = A_{\Lambda_j}^T z, \tag{4.3.1}$$



$\nu = (50)^{-1}\,\nu_{\min}$ $\nu = \nu_{\min}$ $\nu = 50\,\nu_{\min}$

Figure 4.2.6: Regularized reconstruction of data from Figure 4.2.5 with different ν.



cross section at $y = 0.7$ cross section at $y = 0.3$

Figure 4.2.7: Cross sections of the peaks for the three reconstructions given in Figure 4.2.6. Straight and tightly dashed lines corresponding to $\nu = (50)^{-1}\nu_{\min}$ and $\nu = \nu_{\min}$ nearly coincide, whereas the loosely dashed line corresponding to $\nu = 50\,\nu_{\min}$ gives a different shape.

where the diagonal matrix R_j is defined componentwise as $(R_j)_{\lambda,\lambda'} := \delta_{\lambda,\lambda'}\nu_{|\lambda|}$ for some set of scalars $\{\nu_{j'}\}_{j_0 \le j' \le j}$. These scalars are computed inductively at each level following a GCV criterion. At level j_0, we define the influence matrix to be

$$H(\nu_{j_0}) := A_{\Lambda_{j_0}}(A^T_{\Lambda_{j_0}} A_{\Lambda_{j_0}} + R_{j_0})^{-1} A^T_{\Lambda_{j_0}},$$

and ν_{j_0} is obtained by minimization of the corresponding GCV potential (4.2.4). At any subsequent level j we define the influence matrix

$$H(\nu_j; \nu_{j_0}, \ldots, \nu_{j-1}) := A_{\Lambda_j}(A^T_{\Lambda_j} A_{\Lambda_j} + R_j)^{-1} A^T_{\Lambda_j}$$

to have ν_j as only variable and take the $\nu_{j'}$ from previous levels $j_0 \le j' < j$ as parameters. ν_j is then likewise computed based on minimizing (4.2.4).

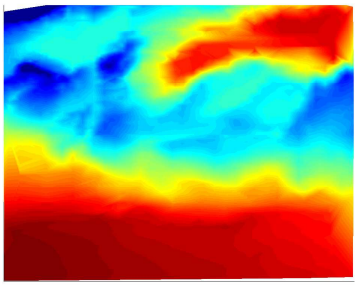
Note that the penalizing term can no more be interpreted as coming from a Sobolev norm of the function in the scale H^α, as we do not prescribe any relation between the penalizing terms $\{\nu_j\}_{j \ge j_0}$.

This approach offers some interesting advantages:

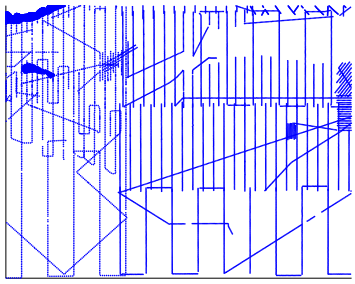
1. The procedure is easily built into the coarse–to–fine growth of the tree.

	Original function	$\nu = 0$	$\nu = (50)^{-1}\,\nu_{\min}$	$\nu = \nu_{\min}$	$\nu = 50\,\nu_{\min}$
approximation error	–	0.705	0.705	0.707	0.969
$\sum_{\mathbf{k},\mathbf{e}} \lvert d_{1,\mathbf{k},\mathbf{e}}\rvert^2$	5.0e-2	5.0e-2	4.4e-2	4.8e-2	5.1e-2
$\sum_{\mathbf{k},\mathbf{e}} \lvert d_{2,\mathbf{k},\mathbf{e}}\rvert^2$	1.1e-2	5.1e-2	1.4e-2	1.1e-2	1.7e-3

Table 4.2.2: Approximation error and ℓ_2 norm of wavelet coefficients at different levels for the reconstructions given in Figure 4.2.6. As additional reference, we provide values corresponding to the original function, whose wavelet coefficients are analytically computable. The columns for different ν's give the numbers for the reconstructions.



Depth map



17089 sampling points.

Figure 4.2.8: Puerto Rico seafloor data.

2. One can attain a higher flexibility for the smoothing effect.

3. Overfitting artefacts are typically localized in scale. This makes a method which is able to disentangle the several scales a natural choice.

4. Computationally, the method is much cheaper, as we will see below at the end of Section 4.3.1

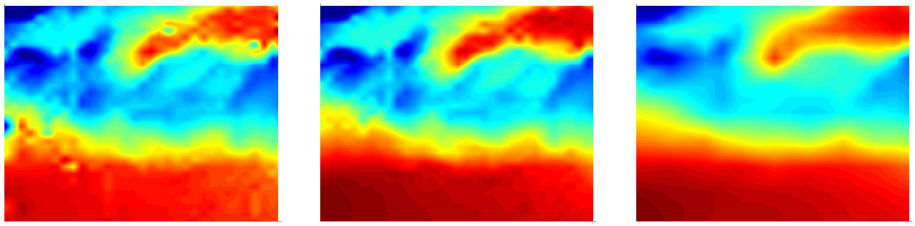
We discuss these points by means of two examples.

4.3.1 Case Study 1: Complexity Reduction

We continue the example with the synthetic data in Figure 4.2.5. We first propose the value $\alpha = 1/2$ and get the reconstruction on the left of Figure 4.3.1 at the GCV

	$\nu = (100)^{-1}\,\nu_{\min}$	$\nu = \nu_{\min}$	$\nu = 200\,\nu_{\min}$
approximation error	7.4e3	7.5e3	3.45e8
$\sum_{\mathbf{k},\mathbf{e}} \lvert d_{1,\mathbf{k},\mathbf{e}}\rvert^2$	5.6e6	5.6e6	5.6e6
$\sum_{\mathbf{k},\mathbf{e}} \lvert d_{2,\mathbf{k},\mathbf{e}}\rvert^2$	1.0e5	1.0e5	7.1e4
$\sum_{\mathbf{k},\mathbf{e}} \lvert d_{3,\mathbf{k},\mathbf{e}}\rvert^2$	4.4e4	4.2e5	7.4e3
$\sum_{\mathbf{k},\mathbf{e}} \lvert d_{4,\mathbf{k},\mathbf{e}}\rvert^2$	1.7e4	1.0e4	2.3e3

Table 4.2.3: Approximation error and ℓ_2 norm of wavelet coefficients at the different levels for the reconstructions given in Figure 4.2.9.



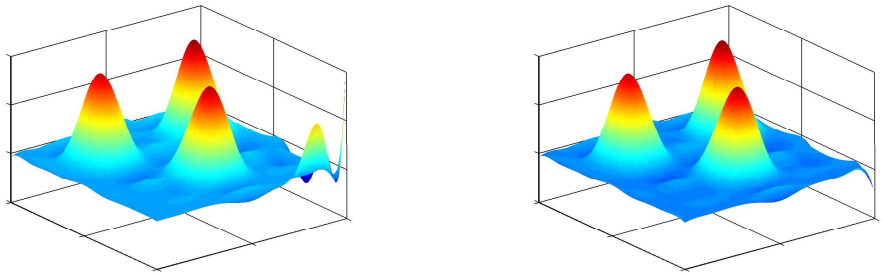
$\nu = \frac{1}{100}\nu_{min}$ $\nu = \nu_{min}$ $\nu = 200\,\nu_{min}$

Figure 4.2.9: Regularized reconstruction of the geophysical data from Figure 4.2.8 with different choices of ν.

minimum. We observe that we do not get rid of the undesired oscillation in the right corner of the figure. A new choice for $\alpha = 5/2$ gives the plot on the right side of Figure 4.3.1. In spite of the fact that for this choice of α the norm equivalence (4.1.3) is no longer valid, the reconstruction is better in this case.

The reason for the fact that two different choices of α yield different results can be easily understood in view of the plot on the left side of Figure 4.3.2. The surface represents the GCV as a function of ν and α. We can now see what happens: the values of the GCV function on the line corresponding to a fixed $\alpha = 1/2$ are clearly higher than the ones on the line corresponding to $\alpha = 5/2$. As there is no clear way to predict which α will give the best result, one should perform a minimization of the GCV *simultaneously* in the *two* parameters α and ν in order to ensure to get the best GCV reconstruction.

However, this procedure increases considerably the computation time, as it re-

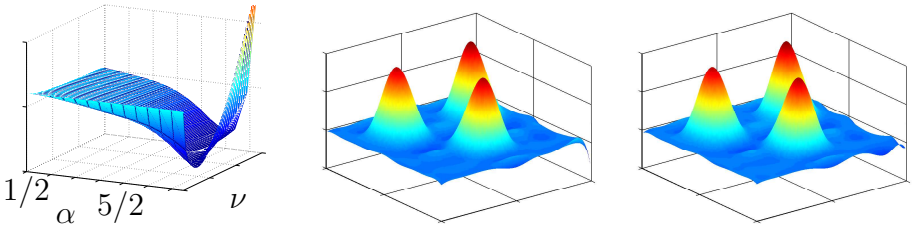


$\alpha = \frac{1}{2}$ $\alpha = \frac{5}{2}$

Figure 4.3.1: Regularized fitting of data from Figure 4.2.5 with ν chosen in the minimum of the GCV for different choices of α.

quires many more GCV function evaluations, each of which involves the solution of two systems with the dimension of the one system in (4.1.4). In contrast, the multilevel GCV only needs to perform two *one*–dimensional GCV minimizations, one for each dyadic level, but attains comparable quality. This method found the parameters $\nu_1 = 0.12$ and $\nu_2 = 3.16$. Such a stronger weighting of $j = 2$ in relation to $j = 1$ cannot be attained by small values of α on the Sobolev scale, independently of whichever ν one may use.

For the present example, the result is given in Figure 4.3.2, middle and right plots. Some numbers on approximation errors and sizes of wavelet coefficients are provided in Table 4.3.1.



Dependence of GCV surface on α and ν

Reconstruction at the minimum of the GCV surface

Reconstruction with multilevel GCV

Figure 4.3.2: Multilevel Reconstruction.

4.3.2 Case Study 2: Scale–Localization of Noise

We apply the algorithm described above to the data set in Figure 4.3.3, which corresponds to a bathymetrical study of part of the sea floor of the Dominican

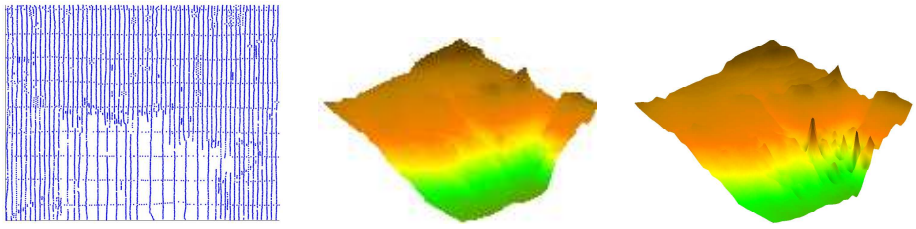
	$\alpha = \frac{1}{2}$	$\alpha = \frac{5}{2}$	$\alpha = 3.1$	multilevel GCV
approximation error	0.914	0.707	0.707	0.709
$\sum_{\mathbf{k},\mathbf{e}} \lvert d_{1,\mathbf{k},\mathbf{e}}\rvert^2$	5.0e-2	4.9e-2	4.8e-2	4.8e-2
$\sum_{\mathbf{k},\mathbf{e}} \lvert d_{2,\mathbf{k},\mathbf{e}}\rvert^2$	5.1e-2	1.4e-2	1.1e-2	1.1e-2
minimal ν	0.05	2.5e-3	7.9e-4	—
penalization at $j = 1$	0.09	0.08	0.06	0.13
penalization at $j = 2$	0.20	2.57	4.29	3.16

Table 4.3.1: Approximation error and ℓ_2 norm of wavelet coefficients at the different levels for the reconstructions given in Figures 4.3.1 and 4.3.2. $\alpha = 3.1$ corresponds to the global minimum of the GCV. The penalization at level j corresponds to $\nu 2^{2\alpha j}$ in the first three columns, and to ν_j in the last column.

Republic. The data also stems from [128]. Measurements of the sea floor depth are irregularly distributed, forming lines, clusters and holes, as seen on the left in Figure 4.3.3. The central plot shows a visualization of the depth of the full set of data points using piecewise linear interpolation.

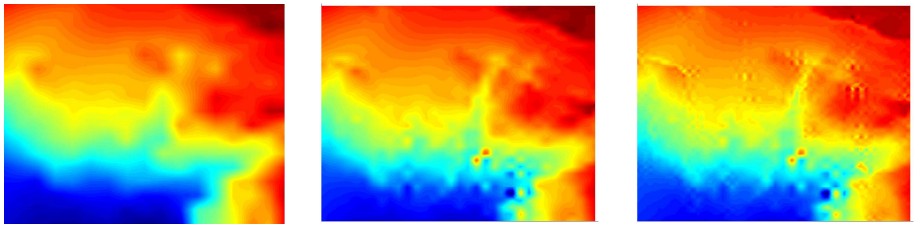
We see on the right of Figure 4.3.4 the effect of an unregularized multiscale reconstruction. In fact, the general shape is correctly caught but it fails to reproduce the high–frequency details of the data in areas of very irregular distribution, making evident the need of some smoothing mechanism.

In order to tune this smoothing we inspect the structure of the artefacts that one wants to prevent. In Figure 4.3.4 we see a view from above of the unregularized reconstruction of the data in Figure 4.3.3 for different dyadic levels. We note that the reconstruction does not present any undesirable structure when the maximal level is $J = 3$. However, at this level not all the features of the data are present in the reconstruction, and further refinement is required. When we add the following level $J = 4$ with wavelets with a correspondingly smaller support, some of them do not contain enough data points and can oscillate without control, giving rise to the oscillations located in a few critical areas, in the plot in the center. Addition of the new dyadic level $J = 5$ creates overall little noisy spots.



Sampling geometry | Piecewise linear interpolation | Unregularized least squares reconstruction

Figure 4.3.3: Bathymetry of part of the Dominican Republic.



$J = 3$ $J = 4$ $J = 5$

Figure 4.3.4: Least squares reconstruction without regularization with increasing maximal resolution level J.

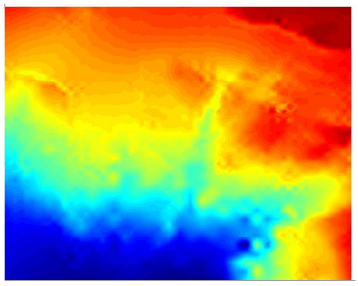
Clearly the problem is located at scales $j = 4$ and $j = 5$. As we see from formula (4.1.5), a Sobolev penalization touches necessarily all the scales, letting only the parameter α free to weight them. Like in the former case the result of the regularization is highly dependent on the right choice of α.

In Figure 4.3.5 we see two reconstructions attained with the GCV method. With $\alpha = 1$, on the left, the GCV minimum does not prevent most artifacts to appear, whereas $\alpha = 2$ provides a much better reconstruction, which is, on the other hand, perhaps too smooth. But there is no obvious way to predict which α is going to give a good reconstruction. In contrast, the multilevel GCV algorithm as explained above circumvents this lack of information and gives an apparently correct reconstruction on the right of Figure 4.3.6.

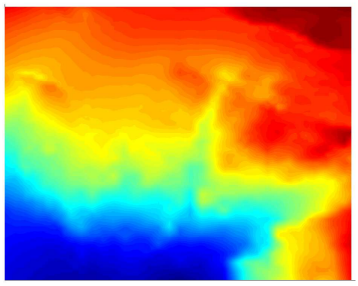
4.4 Other Regularization Strategies

4.4.1 Cutting Off the Iterative Procedure

As explained in [143] in the framework of data fitting with splines or by [87] for more general inverse problems, a regularizing effect can be attained by premature

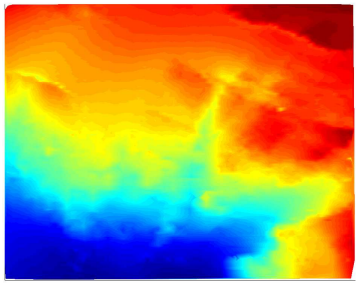


$\alpha = 1$

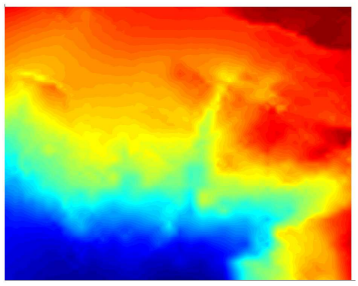


$\alpha = 2$

Figure 4.3.5: GCV-Regularization of data in Figure 4.3.3 for different Sobolev spaces.



linear piecewise interpolation



Reconstruction with multilevel GCV

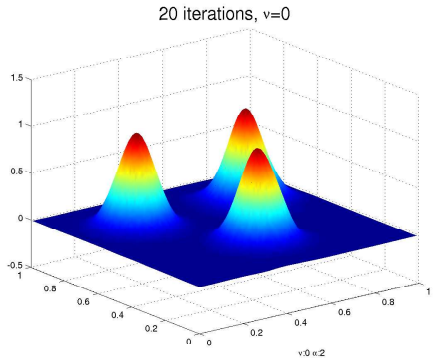
Figure 4.3.6: Results of multilevel GCV regularization of the data from Figure 4.3.3.

cutting off the iterating procedure in the minimizing process. The heuristic reason is that the CG tends to project the solution first onto the coarsest scales. This effect can be observed when analyzing the data of Figure 4.2.5.

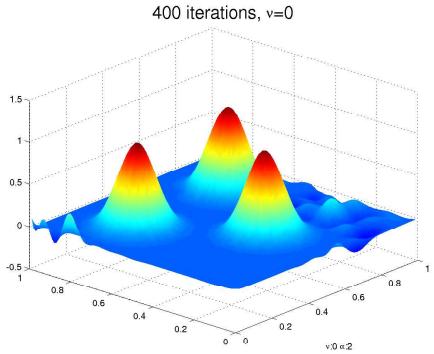
A reasonable smoothness quality can be attained by cutting the minimizing procedures after a few iterations, even in absence of an additional regularizing term in the normal equations (with $\nu = 0$) . The result can be seen in the left hand of Figure 4.4.1. At the right hand, we observe that further iterations damage the reconstruction as high frequencies switch on.

In spite of the simplicity of the idea, the automatic selection of an adequate number of iterations poses a problem by itself, which has been envisaged by different approaches, see [144] for a discussion on the use of GCV techniques for a simultaneous tuning of smoothing parameter ν and number or iterations, or [25] where the L-Curve method (to be explained in the next subsection) is used.

Note the connection of this regularizing approach with methods based on Singular Value Decomposition, that also use a controlled damping of big frequencies and are popular in the image restoration community (see [105] and references herein).



Reconstruction after 20 CG iterations



Reconstruction after 400 CG iterations

Figure 4.4.1: Smoothing by cutting off the iterations in the CG method.

We recall it now briefly. The solution of the problem

$$\min \|Ad - z\|_{\ell_2} \tag{4.4.1}$$

can be expressed in terms of the Singular Value Decomposition of $A \in I\!R^{N\times L}$, that is $A = U\Sigma V^T$, where $U \in I\!R^{N\times N}$ and $V \in I\!R^{L\times L}$ are orthogonal and Σ is a diagonal matrix whose entries are the singular values of A, $\{\sigma_i\}_{i=1,\ldots,r}$, if $r = rank(A)$. We can now write the solution to (4.4.1) in terms of the basis $\{v_i\}_{i=1,\ldots,N}$ given by the columns of V. If we expand the data vector z in terms of the basis $\{u_i\}_{i=1,\ldots,N}$ given by the columns of U as

$$z = \sum_{i=1}^{N} \xi_i u_i, \text{ with } \xi_i = u_i^T z,\ i = 1, \ldots, N, \tag{4.4.2}$$

the solution of (4.4.1) can be written as

$$d = \sum_{i=1}^{r} \frac{1}{\sigma_i} \xi_i v_i. \tag{4.4.3}$$

This representation allows for introducing a regularizing effect by frequency-dependent filtering, that is, one constructs a regularized approximation d_{reg} to the solution of (4.4.1) by multiplying each eigenvalue contribution with a *filter factor* ϕ_i

$$d_{reg} = \sum_{i=1}^{r} \frac{1}{\sigma_i} \phi_i \xi_i v_i. \tag{4.4.4}$$

From the analytic point of view the Tikhonov regularization can be understood as a special form of this technique, as it boils down to the selection of one specific form of filters. As this methods requires the Singular Value Decomposition of the matrix A, its application is conditioned to availability of techniques to compute or at least approximate it easily, as it involves heavy computations.

4.4.2 L-Curve Method

We use now another popular method of automatic selection of the smoothing parameter, namely the L-Curve method, used in inverse problems comparable to the data fitting scenario in fields as geophysics [28] or medical tomography [106].

The method operates by fixing a set $\mathcal{N}$ of regularizing parameters ν, and computing the corresponding approximations $\{d^\nu\}_{\nu\in\mathcal{N}}$. Then one compares the norm of the residuals and the norm of the approximations by plotting $\|d^\nu\|_{\ell_2}$ versus $\|z-f^\nu\|_{\ell_2}$ for the selected range of ν. The motivation is the following: one expects that big values of the smoothing parameters will flatten the reconstruction so as to generate big residuals, whereas too small values of ν allow for solutions with big structure and consequently big norms of the coefficients. The L-Curve tries to visualize the trade-off of this two quantities. In particular, a straightforward order of magnitude analysis of the singular value representation of the solutions and residuals (see [68] or [91] for details) shows that in the case of data corrupted by white noise, for medium-big values of ν the norm of the solution varies little with increasing ν so that the locus of $(\|z - f^\nu\|_{\ell_2},\|d^\nu\|_{\ell_2})$ appears in general very flat. Inversely, in the area in which ν is small, the dominance of the noise to an increase of the approximation norm while the residuals change comparatively little, creating a steep slope in the $(\|z - f^\nu\|_{\ell_2},\|d^\nu\|_{\ell_2})$ curve. These two effects conform an L-shaped curve. The L-Curve criterion is to pick the parameter value lying the corner of this curve, which is usually defined as the value of ν that maximizes the curvature defined as

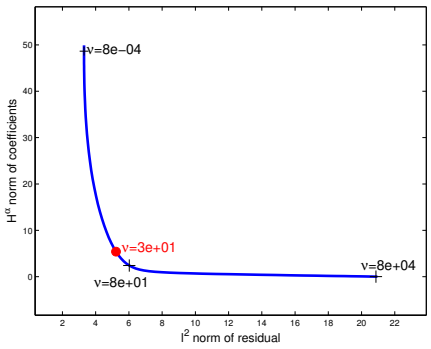
$$\kappa(\nu) = \frac{\xi''(\nu)\eta'(\nu) - \xi'(\nu)\eta''(\nu)}{(\xi'(\nu)^2 + \eta'(\nu)^2)^{\frac{3}{2}}}. \tag{4.4.5}$$

where $\xi(\nu) = \|z - f^\nu\|_{\ell_2}$ and $\eta(\nu) = \|d^\nu\|_{\ell_2}$ Although the mathematical background of the L-Curve does not sound as solid as the one of GCV, comparisons of both methods report comparable results [89].

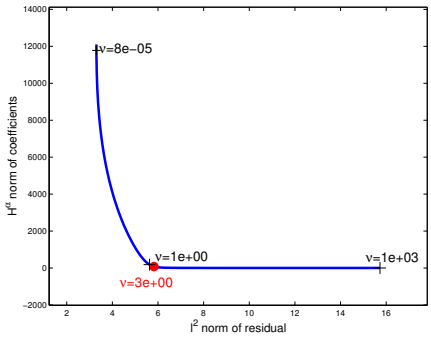
We represent in Figure 4.4.2 the L-diagrams created by the method when analyzing the data from Figure 4.2.1 with $\alpha = 1, 2, 4, 10$, as done before by the means of the GCV. The diagrams are indeed L-shaped and the order of magnitude of the ν lying at the corner provide acceptable reconstructions, being close to the value of the ν predicted by the GCV (marked in red on the same diagram). The corresponding reconstructions are given in Figure 4.4.3 and an analysis of the quality of the minimum for the different choices of α can be seen in Table 4.4.1.

α	$\nu_L \cdot 10^{-3}$	$\nu_L \cdot 10^{-2}$	$\nu_L \cdot 10^{-1}$	ν_L	$\nu_L \cdot 10$	$\nu_L \cdot 10^{2}$	$\nu_L \cdot 10^{3}$
1	4.993	4.352	2.855	2.409	7.396	16.632	19.780
2	4.808	3.841	2.516	1.589	1.688	5.139	14.301
4	2.594	2.085	1.631	1.278	1.092	1.858	5.850
10	1.205	1.070	1.099	1.105	1.107	1.359	3.645

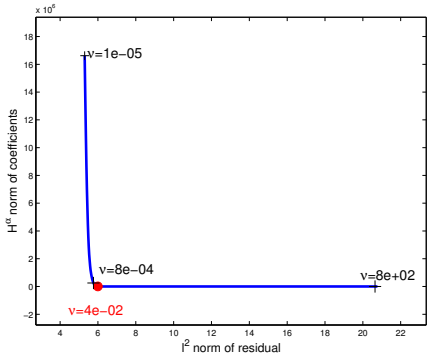
Table 4.4.1: ℓ^2 error of the reconstruction for different values of ν surrounding the maximum curvature point ν_L of the L-curve for different choices of α.



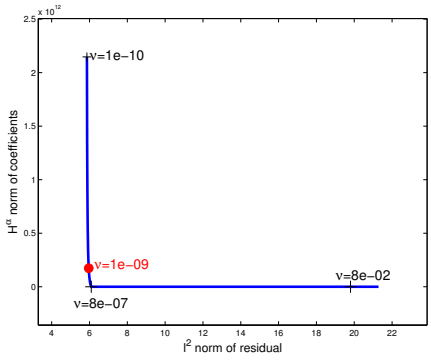
$\alpha = 1$



$\alpha = 2$

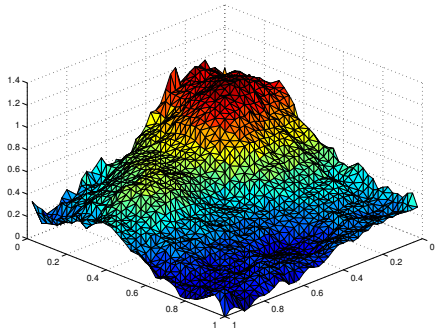


$\alpha = 4$

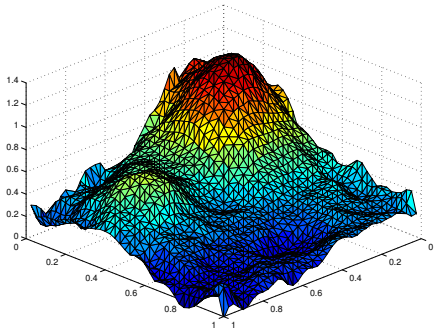


$\alpha = 10$

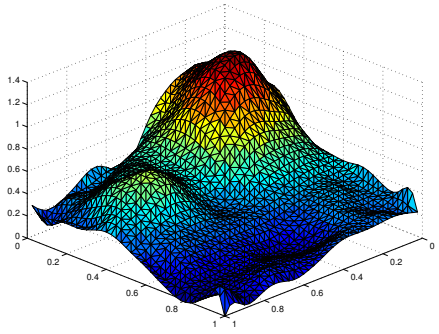
Figure 4.4.2: Coefficient norm / residual plots illustrating the applicability of the L-curve method to the corrupted Franke data as given in Figure 4.2.1. The red marked quantities correspond to the ν that minimizes the GCV.



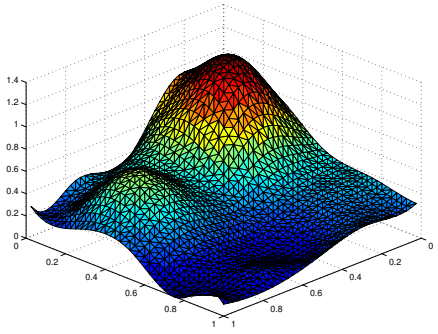
$\alpha = 1$



$\alpha = 2$



$\alpha = 4$



$\alpha = 10$

Figure 4.4.3: Reconstructions of the corrupted Franke data from Figure 4.2.1 by the L-curve method for the choices of α detailed in Figure 4.4.2. Compare these results with those in Figure 4.2.3, attained by the GCV method.

Chapter 5

Robust Regression

We point out possible extensions of our method to cope with data distributions corrupted with bad measurements.

5.1 Classical Robust Regularization

The M-estimator method (see e.g. [97]) proposes a revision of the classical Least Squares method of finding a function $f = f(\theta_1, \ldots, \theta_m)$ that approximates the data points $\{(x_i, z_i)\}_{1,\ldots,N}$ by finding the set of parameters $\theta_1, \ldots, \theta_m$ that minimizes

$$J(f) = \sum_{i=1}^{N} (r_i)^2 \tag{5.1.1}$$

where $r_i := z_i - f(x_i)$ is the *residual* of the i-th measurement. This strategy causes problems when some *outlying* points have vertical values that, possibly as a result of a bad measurement, are very different from the ones of the neighboring data. They typically produce a large residual. The contribution of these points is still amplified by the square function in (5.1.1). This means that the global minimization of (5.1.1) is highly influenced precisely by these erroneous points.

A possible cure is an alternative weighting for the residuals. One replaces $(\cdot)^2$, which enhances large residuals, by some other function ρ that does not increase so rapidly with large values of the argument, and considers the f that minimizes

$$J_\rho(f) = \sum_{i=1}^{N} \rho\left(z_i - f(x_i)\right) \to \min! \tag{5.1.2}$$

for a fixed ρ. The *M-estimator* is now given by the set of parameters $\theta_1, \ldots, \theta_m$ that solves

$$\sum_{i=1}^{N} \psi(r_i) \frac{\partial r_i}{\partial \theta_j} = 0 \ \text{ for } j = 1, \ldots, m, \tag{5.1.3}$$

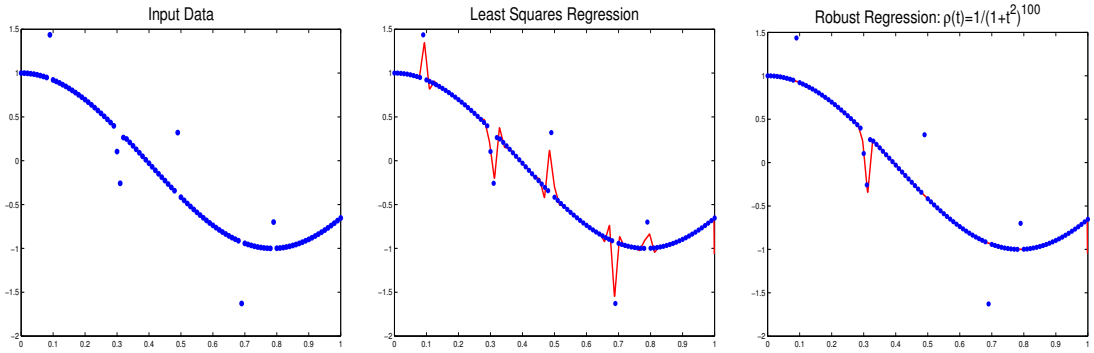


Figure 5.2.1: Successful application of M-Estimator techniques.

where ψ is called the *influence function*, defined as $\psi(t) := \frac{d\rho(t)}{dt}$. We can now define a *weight function* $w(t) := \frac{\psi(t)}{t}$ and write (5.1.3) as

$$\sum_{i=1}^{N} w(r_i) r_i \frac{\partial r_i}{\partial \theta_j} = 0 \text{ for } j = 1, \ldots, m. \tag{5.1.4}$$

The parameters that solve these equations can be identified (under adequate conditions on ρ) as the solution of a Least Squares problem with weights:

$$\sum_{i=1}^{N} w(r_i)(r_i)^2 \rightarrow \min! \tag{5.1.5}$$

and can be computed iterating on k

$$\sum_{i=1}^{N} w(r_i^{(k-1)})(r_i^k)^2 \rightarrow \min! \tag{5.1.6}$$

until consistency is reached.

Popular choices are $\rho(t) = |t|$ (giving $w(t) = |t|^{-1}$) or $\rho(t) = \frac{t^2/2}{1+t^2}$.

5.2 Wavelets and M-Estimators

An example of application can be seen in Figure 5.2.1. The classical Least Squares approximation tries to adapt to the outliers, while the robust one ignores them, because they produce big residuals and contribute only slightly to (5.1.2). Note that the method recognizes the other local features present in the plot, the peak pointing downward located at $x = 0.3$. This actual peak is not deleted.

However, the performance of the method is conditioned to the assumption that outliers indeed produce big residuals. This is not certain to occur, especially when

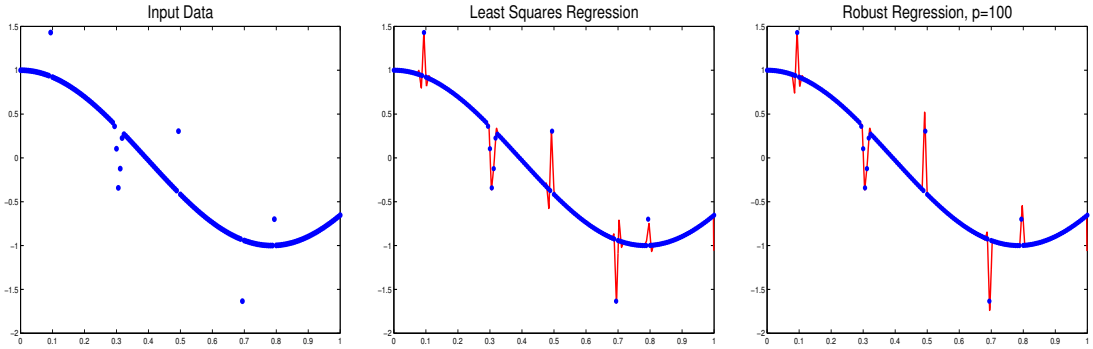


Figure 5.2.2: Problems of M-estimator techniques. The function $w(t) = \frac{1}{(1+t^2)^p}$ is used as weighting.

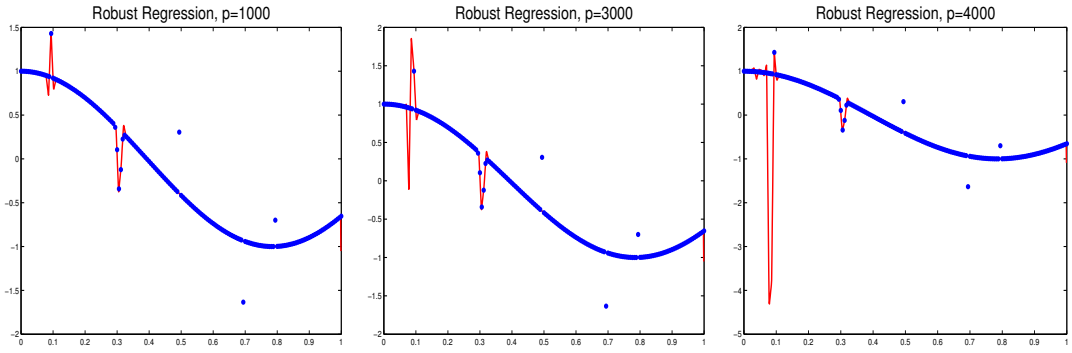


Figure 5.2.3: Increasing robustness varying the parameter p in $w(t) = \frac{1}{(1+t^2)^p}$.

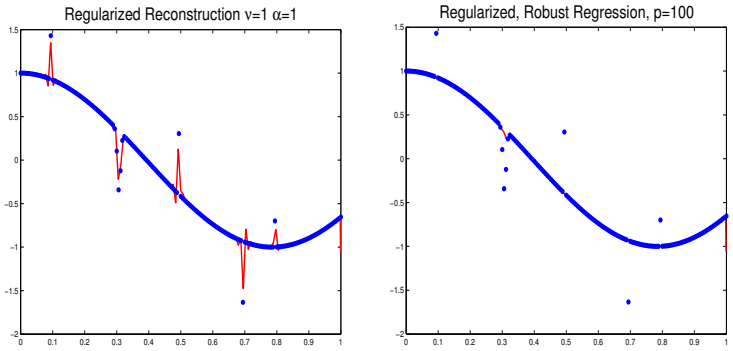


Figure 5.2.4: Interplay of regularization and robustness.

we work with wavelets, or with some other locally-adapting method, as the outliers are treated as local irregularities, and the method can be flexible enough to produce a high local oscillation that reproduce the data. The data in Figure 5.2.2 is a slight modification of the one in Figure 5.2.1. This modification is enough to generate a reconstruction in which outliers are not automatically signalized by their big residuals. A moderate ρ does not detect the outliers.

Changing ρ to produce more and more aggressive robustness by bigger and bigger penalization to the big residuals, produces results like in Figure 5.2.3. Some outliers are detected, but one of them remains undetected. It is in fact undetectable, as its initial residual was zero. Still worse, some regular points are erroneously considered as outliers, as the reconstruction give them a small, non zero residual. An amplification of this effect can be seen on the right of the figure.

Let us end this subsection noting that this effect could be solved by a *regularization* technique. As the problem seems to be that the flexibility of the wavelets do not allow for big residuals to be formed, one can think of adding some regularizing term to (5.1.1), so that the approximating f not only tries to cut the data points but to simultaneously keep a smooth appearance. Experiments show that this procedure is indeed able to mark outliers with big residuals (see the left hand side of Figure 5.2.4) that are detected by the nonlinear functional in (5.1.2). However it causes the additional problem that points forming some local structure tend to be marked with residuals as well as outliers, leading to a deletion of that local structure, as seen on the right of the figure.

5.3 Residual-free Methods

Diverse approaches (see [15], [107]) have proposed robust methods of regression based on wavelets. These methods do not work with L-estimator techniques but with adaptations of the thresholding procedure, and rely on the wavelet transformation of the data.

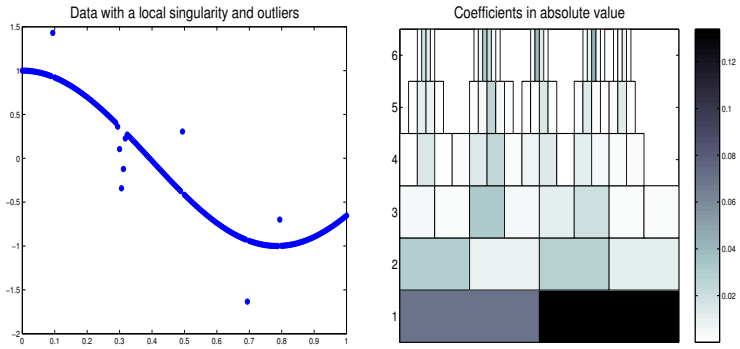


Figure 5.3.1: Corrupted data and wavelet coefficients of the reconstructing function.

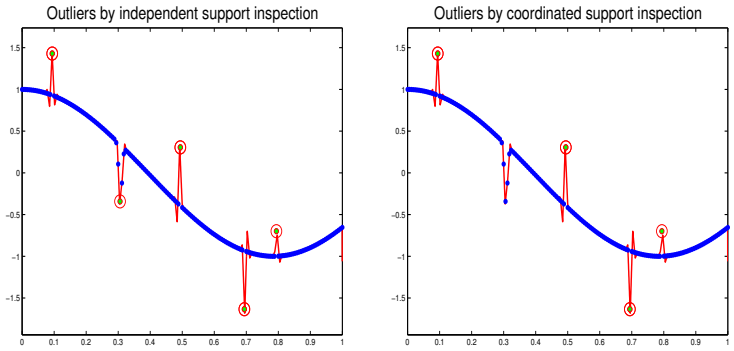


Figure 5.3.2: Outlier detection by wavelet-driven data inspection.

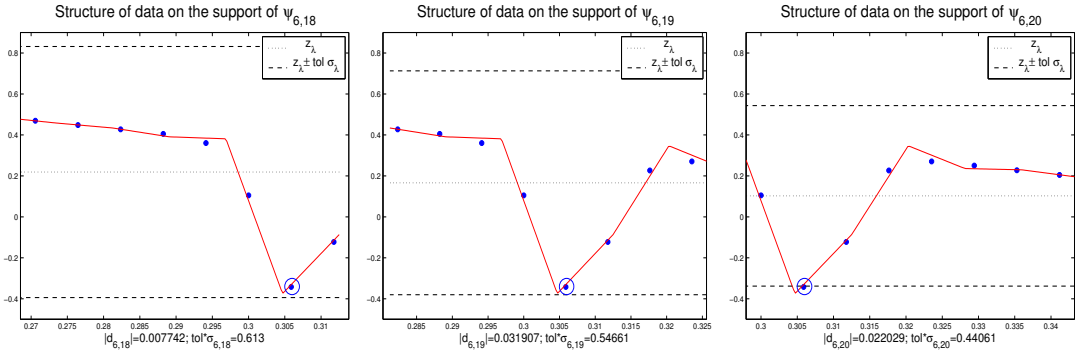


Figure 5.3.3: Local variation in conflict areas.

We could think of an alternative procedure where we take direct advantage of the adaptivity of wavelets. If we just accept that the wavelet reconstruction will try to reproduce every local feature of the data, we end up with a set of coefficients that can give us a clue about the location of the outliers. Intuitively, an outlier will cause in its vicinity a severe oscillation of the approximating function. This will cause some local wavelet coefficients to attain a high value. We can therefore expect the outliers to be in the supports of wavelets showing a big coefficient, and locate them by an adequate inspection of the points lying in the supports of those wavelets.

Once we know which data areas are likely to present corrupted measurements, we locally analyze the data and determine for each point if it is likely to be an outlier. This requires constructing some criterion that decides this likeliness using only the local characteristics of the data.

We could complete this method with a complementary inspection on the points selected in the previous step, comparing the likeliness to be an outlier received by a point when inspected in the support of a wavelet and when inspected in the support of other wavelets.

Let us fix ideas and notation by considering the following algorithm:

1. Construct a Least Squares approximation $f = \sum_{\lambda \in \Lambda} d_\lambda \, \psi_\lambda$.

2. Select the set of indices Λ_ϵ of wavelets of the maximal level J that are bigger than some threshold

$$\Lambda_\epsilon := \{\lambda : |d|_\lambda \geq \epsilon, |\lambda| = J\}. \tag{5.3.1}$$

3. For each $\lambda \in \Lambda_\epsilon$, pick the set of points that could be considered as outliers and denote them by O_λ. This can be attained in the following steps:

   (a) Consider the set P_λ of indices of points lying in the support of ψ_λ

$$P_\lambda := \{i : x_i \in \text{supp } \psi_\lambda\}. \tag{5.3.2}$$

   (b) Compute the local variance and mean value of the vertical coordinates of these points

$$\sigma_\lambda := \text{var}(\{z_i\}), \; i \in P_\lambda, \tag{5.3.3}$$

$$z_\lambda := \text{mean}(\{z_i\}), \; i \in P_\lambda. \tag{5.3.4}$$

   (c) Mark as possible outliers the points in P_λ that deviate too much from the general local trend. That is, for a fixed parameter tol, select

$$i \in O_\lambda \Longleftrightarrow \{|z_i - z_\lambda| \geq tol \cdot \sigma_\lambda\}. \tag{5.3.5}$$

4. For each detected i in some O_λ, check whether or not other wavelets also mark this point as an outlier. This can be done as follows:

(a) Denote by $\Lambda_{\epsilon,i}^{outlier}$ the indexes of wavelets including the point x_i that *mark* it as an outlier,

$$\Lambda_{\epsilon,i}^{outlier} := \{\lambda \in \Lambda_\epsilon \ : \ i \in P_\lambda, x_i \in O_\lambda\}. \tag{5.3.6}$$

(b) Denote by $\Lambda_{\epsilon,i}^{regular}$ the indexes of wavelets including the point x_i that *do not mark* it as an outlier,

$$\Lambda_{\epsilon,i}^{regular} := \{\lambda \in \Lambda_\epsilon \ : \ i \in P_\lambda, \ i \notin O_\lambda\}. \tag{5.3.7}$$

(c) Compute a wavelet-weighted criterion to definitively classify the point as an outlier or not, i.e.,

$$i \text{ is an outlier } \iff \sum_{\lambda \in \Lambda_{\epsilon,i}^{outlier}} |d_\lambda| \geq \sum_{\lambda \in \Lambda_{\epsilon,i}^{regular}} |d_\lambda|. \tag{5.3.8}$$

Step 4 of the previous algorithm is to be understood as an attempt to prevent getting ride of actual, valuable information. Consider Figure 5.3.1. The synthetic data presents four actual outliers, located at $x = 0.1, 0.5, 0.7, 0.8$ and a little cusp located at $x = 0.3$. This cusp is formed by several points forming a local trend in the data. The spectrum of the reconstruction is given at the right of the figure. (By the way, note that the ramification of the wavelet-coefficient-tree caused by the cusp looks very similar to the ones caused by the outliers. It does not seem easy to track outliers only by inspecting the wavelet coefficients).

The use of the former algorithm without Step 4 produced the result on the left of Figure 5.3.2. Local checking on the supports of the wavelets with the highest coefficients selected the points marked with a circle. Outliers are effectively selected, but a real-feature point in the cusp was also marked. This is inconvenient, because the sampling has very few data points in the proximities of the cusp, where we would like to have a better resolution. We would like to keep precisely these data points, and not those in the smooth areas, where the information is redundant.

This mistake would be detected by the additional Step 4. A look at Figure 5.3.3 gives us a clearer insight into what is happening. The deepest point of the cusp is marked by the wavelet $j = 6, k = 20$ as an outlier (right of the figure, point marked by a circle). But wavelets $k = 18, k = 19$ (left and center), whose support includes the point as well, consider it as regular data.

The reason is that the wavelet $j = 6, k = 20$ includes in its support many well behaved points, and only a point coming from the cusp. Hence, the local tolerance (given by the local variance) is bound to be stricter as for the other two wavelets, which include other points in the cusp and therefore "know better" what is really happening. The translation of this "better knowing" is they let a wider tolerance area around the local mean value, as the variance is bigger, and the conflict point is inside the tolerance range. Therefore, we see that it is useful to involve several wavelets to decide the final deletion of a point out of the data.

The point 4.c of the algorithm is included trying to help that this final decision is made penalizing those wavelets that have less to do with the artifact. Shortly, this means that the more affected a wavelet is by a local structure (outlier or cusp), the larger is its coefficient, in comparison to the wavelets in the neighborhood, and also the more entitled is the wavelet to "judge about the outlierness" of the point.

Several advantages of this way to inspect the data using simultaneously the data itself and its wavelet coefficients can be mentioned:

- We can construct criteria to control the size and location of the analyzing windows and their interplay.

- Most importantly for unorganized distribution of points, the method allows to restrict the inspection to some areas of the domain. The required data structures coincide with the ones needed to construct the Least Squares reconstruction and do not require additional grid operations.

5.4 A Least Squares Specific Methodology

We try to develop an outlier-detection procedure that exploits our Least Squares procedure.

How can one define an outlier? Before we try to construct a mathematical definition which one can plug in an algorithm, we just try to mimic the process in which the eye tells us what an outlier is: The presence of an outlier must create a "cusp" in the approximating function. That is: an artifact extremely well located in space *and* frequency. That is why the wavelet framework naturally suits the outlier location problem.

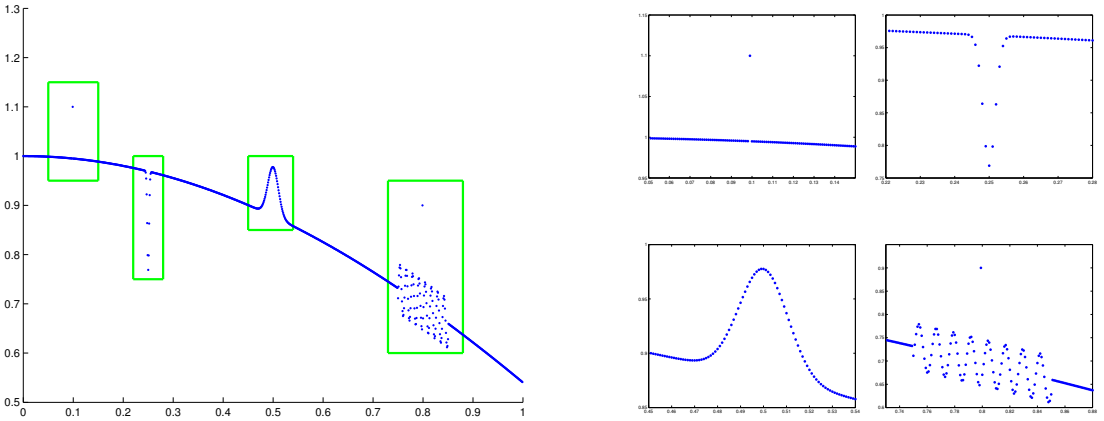
Several frameworks separate in fact the removal of outliers and the wavelet analysis of the data. We rather want to build the outlier detection into the wavelet representation, and take advantage of it. As the end product of the data fitting procedure is the construction of a function f that represents the data, it seems natural to transfer the problem of the outlier definition from the point of view of the raw data to the point of view of this approximating f.

If the approximating function correctly catches the data features, the presence of an outlier must create a local jump in the approximating function. Otherwise, if f is not affected by the outlier, there is no point in worrying about it.

This leads to reformulate the problem into the following questions:

1. how one can define these jumps in a rigorous mathematical way that is simultaneously easy to use, and

2. how one can distinguish jumps created by outliers and jumps really contained in the data.



Data set 1. local details of the data

Figure 5.4.1: Model data set 1.

To fix ideas in this issue, let us consider in the data set in Figure 5.4.2, where we have marked four areas representing different prototype situation.

1. In the area marked as 1 we find a point that is definitely classifiable as an outlier. Or to put it another way: we fix the criterion that a well performing method should mark the point (x_{100}, y_{100}) as an outlier.

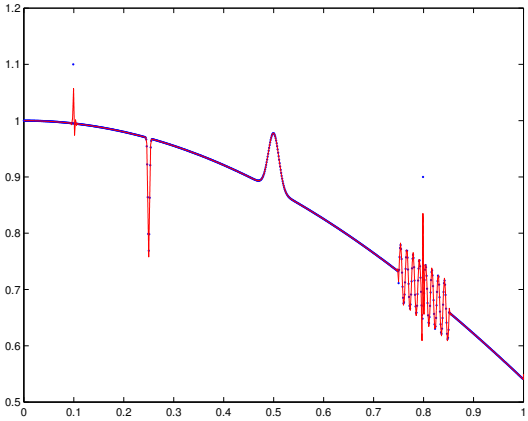
2. This area would represent a cusp really represented in the data. The points in this area are not to be marked as outliers. Moreover, they are the representatives of a high frequency phenomenon on the data. The accidental removal of points in this area would eliminate significant information of this local structure.

3. Like in the previous zone, the data presents a spatially located high frequency feature. In this case the frequency is lower as in zone 2, so that more points from the data set are involved in the representation of the local structure. This could represent a non critical area of the domain. No outlier is present, and anyway wrong removal of points on this area does not necessarily eliminate significant information, as the remaining points would certainly reproduce the local features of the data.

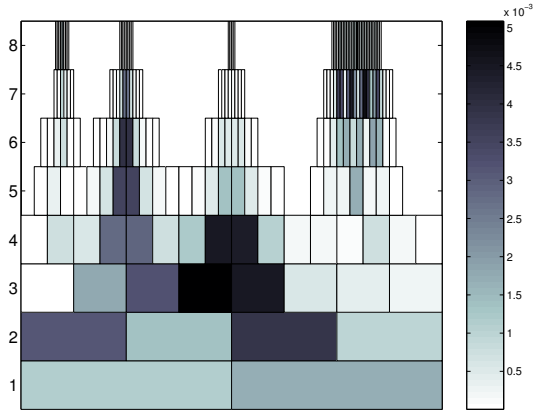
4. The point (x_{800}, y_{800}) is an outlier embedded in a highly energetic zone. These kind of points pose the hardest difficulties to outlier detection algorithms. For one side the neighborhood can mask the effect of the outlier. On the other hand, neighboring points carry significant information about the local structure on the data, and false removal in this area should be avoided.

The wavelet representation of the data allows for an extremely practical answer to both questions.

- Concerning the identification of jumps, there is a rich literature. The basic idea is that the presence of a jump is reflected by large wavelet coefficients.
- Outlier caused jumps cannot be distinguished by data inherent jumps only by inspection of the wavelet representation of the data: one has to go back to the point themselves and analyze how individual points influence the wavelet representation. This task can be easily done after having constructed the Least Squares approximation to the data, see Figure 5.4.2, as all the information needed to perform this analysis has been already processed in the data structures.



Adaptive Reconstruction with 190 Wavelets.



Wavelet Coefficients

Figure 5.4.2: Reconstruction of example data set 1 from Figure 5.4.1.

5.5 Basics of the Approach

At the end of the least squares data fitting algorithm we end up with two objects:

1. the coefficients $\{d_\lambda\}_{\lambda\in\Lambda}$ of the constructed approximation, as main product, and
2. an efficient encoding of the observation matrix A, which was used and updated in the process.

The main computational effort has been done in the construction of A, which contains implicitly a very complete analysis of the data. We obviously would like to envisage to a method that reuses all this information.

In our approach we separate the outlier finding procedure into two steps:

1. Locate the areas in which an outlier can be present.

   As explained previously, this can be done by simple inspection of the coefficients $\{d_\lambda\}_{\lambda\in\Lambda}$, as the presence of outliers must cause large wavelet coefficients at high levels. One evident possibility could be to choose a thresholding parameter ϵ and a j_{depth} that models the expected penetrating depth of the outliers and identify a set of wavelets indexes

   $$\Lambda_{\epsilon,j_{depth}} := \{\lambda \in \Lambda;\, |d_\lambda| \geq \epsilon, |\lambda| \geq j_{depth}\} \tag{5.5.1}$$

   Now we extract out of the original N points those embedded in the support of some wavelet in $\Lambda_{\epsilon,j_{depth}}$ and record their indexes in the set S:

   $$i \in S \leftrightarrow \exists\lambda \in \Lambda_{\epsilon,j_{depth}} \text{ so that } x_i \in \text{supp}\psi_\lambda. \tag{5.5.2}$$

   This step serves only to reduce the number of points in outlier-affected areas.

2. Test *all* the points in the areas identified in the first steps.
   Basically, a merit figure is computed for each point included in S, some kind of outlierness coefficient $\omega(i)$, and points whose ω is above a predefined threshold τ are discarded from the data. That is the decisive part of the algorithm, as the concrete mathematical translation of the outlier concept has to be built into the computation of the merit figure. In the next sections we describe and analyze several ways to construct meaningful outlierness profiles $\omega(i)$, $i \in S$.

5.6 Global Refitting Criterion

We have a set S of indices of points that have been identified as possible outliers by the first step, as they lie in the support of wavelets with large coefficients. Let us take one of these points, (x_i, y_i) $i \in S$, and test if it is an outlier. As mentioned before, in our approach, the basis of this check would be to measure to which extent its presence alters the shape of the approximation. This goal can be attained in two steps:

1. Construct an approximation to the whole data set.

2. Construct an approximation to the whole set of data *except* the point to be checked. To do so we use the same configuration of wavelets as used to compute the approximation to the whole data set.

   We compute therefore

$$f_{[i]} := \arg \min_{g=\sum_{\lambda\in\Lambda} d_\lambda \psi_\lambda} \sum_{l\neq i} (z_l - g(x_l))^2 \tag{5.6.1}$$

3. Compare the behavior of f and $f_{[i]}$ in the neighborhood of x_i.

   As explained above, this can be done in a natural way using the wavelet coefficients. First of all we need an interpretation of the neighborhood concept. To do that, we define the *influence set* of a point (x_i, y_i) up to level j_{cut} in the index set Λ as the subset of Λ that includes indexes of the wavelets whose support overlap x_i. We denote it as

$$\Lambda_{j_{cut},i} := \{\lambda \in \Lambda;\ |\lambda| \geq |j_{cut}|;\ x_i \in \text{supp}\psi_\lambda\} \tag{5.6.2}$$

   or shortly Λ_i when the subscript j_{cut} is clear or irrelevant.

   Now, we want to compare the local behavior of f and $f_{[i]}$. We define the *local energy* of a function by means of a weighted summation of a subset of its wavelet coefficients. That is, for a function $g = \sum_{\lambda\in\Lambda} d_\lambda \psi_\lambda$ and a set $\Lambda' \subset \Lambda$, we define

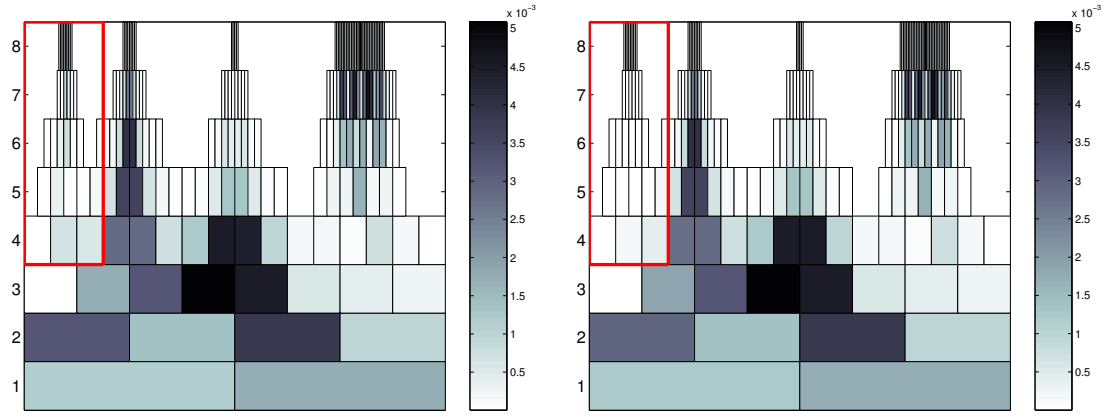
$$E^{\Lambda'}_{\alpha,p,q}(g) := \left[\sum_j \left(2^{j(\alpha+1/2-1/p)} \left(\sum_{\mathbf{k},\mathbf{e},(j,\mathbf{k},\mathbf{e})\in\Lambda'} |d_{j,\mathbf{k},\mathbf{e}}|^p \right)^{\frac{1}{p}} \right)^q \right]^{\frac{1}{q}} \tag{5.6.3}$$

   This definition relies obviously in the norm equivalence relation between Besov seminorm and wavelet coefficients, see pag. 123 of [92]. In view of this interpretation, if (x_i, y_i) is indeed an outlier, in the neighborhood of x_i the local energy of $f_{[i]}$ should be much smaller than the local energy of f. This motivates the following definition. We define the merit profile of point i according to a global criterion as

$$\omega_{global}(i) := \log \left(\frac{E^{\Lambda_{j,i}}_{\alpha,p,q}(f)}{E^{\Lambda_{j,i}}_{\alpha,p,q}(f_{[i]})} \right). \tag{5.6.4}$$

   With this definition, typical good performing τ thresholding values are expectable in the order of magnitude of unity. This would mean that in our model, we expect the presence of an outlier to cause a noticeable increase of the local energy.

We revisit now our example data set and explore in detail how this method would work on the different situations represented in the data. If we take the outlier (x_{100}, y_{100}) and compute the global approximations f and $f_{[100]}$, we get the wavelet coefficients pictured in Figure 5.6.1. As expected, no difference is visible outside the red line, which encloses the local wavelet coefficients. The presence of the outlier does really act locally.



Wavelet Coefficients of f. Wavelet Coefficients of $f_{[100]}$

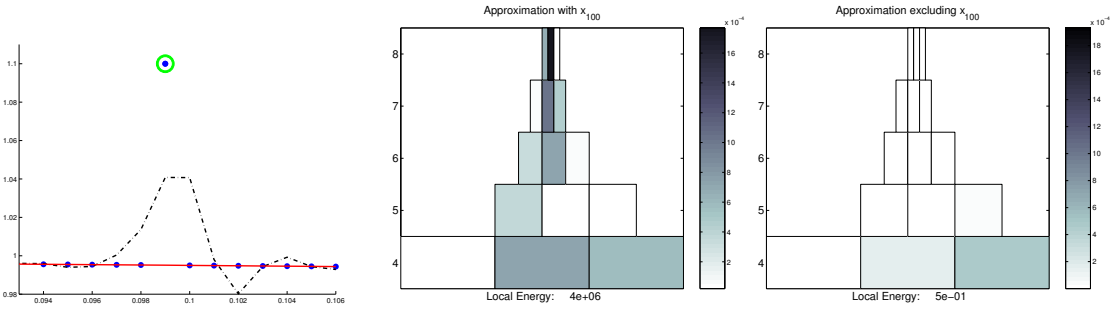
Figure 5.6.1: Global removal criterion on (x_{100}, y_{100}).

We describe this action in more detail in Figure 5.6.2. In the first plot we can compare f (dashed line) and f_{100} (solid line) in the neighborhood of (x_{100}, y_{100}). The former does obviously not need to create the jump that tries to reproduce the point (x_{100}, y_{100}). This is reflected in the coefficients of the wavelets in Λ_{100}, as one can see in the following plots of the same Figure: the energy content of the set Λ_{100} is practically empty after subtracting (x_{100}, y_{100}) from the data. In the energy computation, the cutting level j_{cut} of the influence set in (5.6.2) was taken as $j = 3$ and the local measure of energy has parameters $\alpha = 5/2, p = 2$. The criterion (5.6.4) would give a merit figure of $\omega_{global}(100) = 7.02$ for this point.

The same computation for the regular neighboring point (x_{102}, y_{102}) gives $\omega_{global}(102) = 0.0078$, as its removal does not critically vary the local energy, as one can appreciate in Figure 5.6.2.

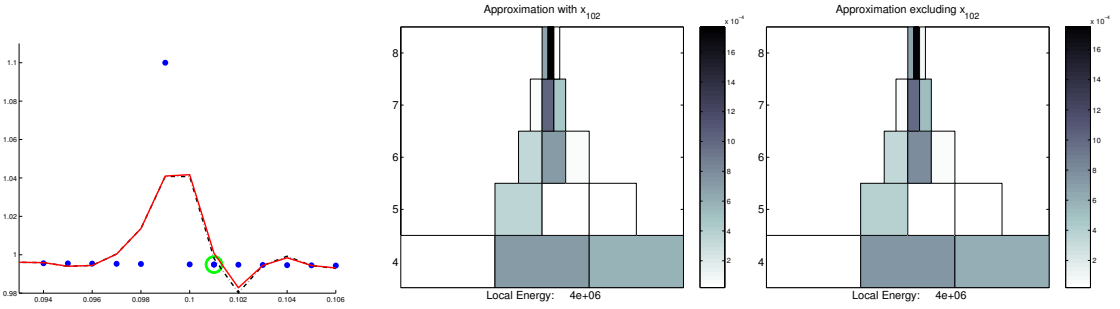
Furthermore, the point (x_{251}, y_{251}) is located in the middle a fine structure feature of the data. The removal of the point damages but does not destroy the structure, as can be seen in Figure 5.6.4. The similarity of the two approximations gives an outlierness coefficient of $\omega_{global}(251) = -0.27$, so that the algorithm will mark it as a regular point, preventing the loss of information.

Finally we consider the outlier (x_{800}, y_{800}). As it is located in a high energetic environment, it is more complicated to disentangle its effects from those of the data features, as one can see in Figure 5.6.4, where one can appreciate that the local



Local view of f and $f_{[100]}$. Coefficients of wavelets in Λ_{100} Coefficients of wavelets in Λ_{100}

Figure 5.6.2: Global removal criterion on (x_{100}, y_{100}).



Local view of f and $f_{[102]}$. Coefficients of wavelets in Λ_{102} Coefficients of wavelets in Λ_{102}

Figure 5.6.3: Global removal criterion on (x_{102}, y_{102}).

energy decay caused by the subtraction of the point is not so dramatic as in the case of the outlier in a flat background (compare with Figure 5.6.2).

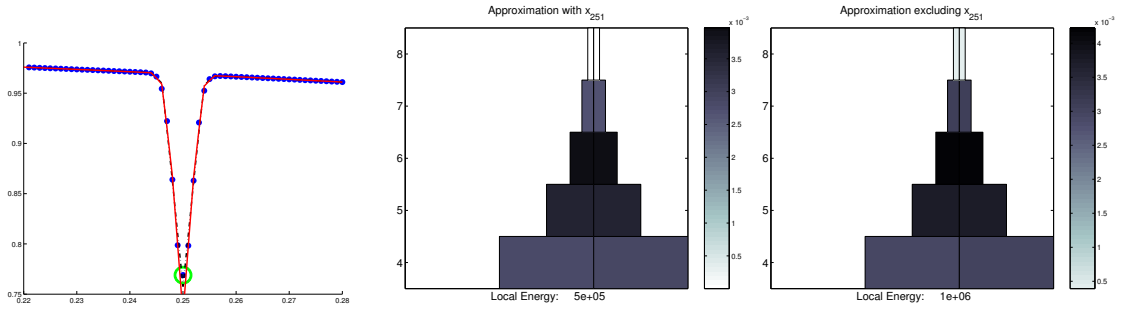
Still, (x_{800}, y_{800}) attains a merit figure of $\omega_{global}(800) = 1.2$, so that the method would classify the point correctly as an outlier.

5.6.1 Removal of Points in Normal Equations

The former strategy has an obvious drawback. It requires the computation of $f_{[i]}$ for every suspicious i. This amounts to construct and solve a different set of normal equations for every i.

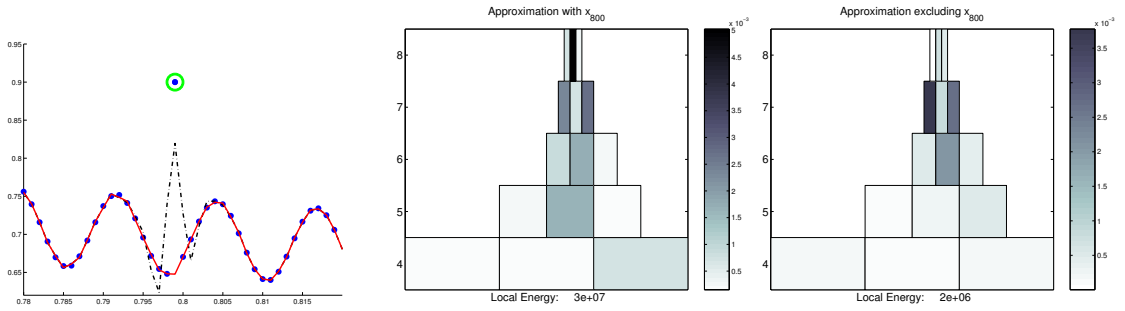
Anyway, the structure of the problem allows for some possible simplifications. If the normal equations of the original problem are

$$Md = b, \tag{5.6.5}$$



Local view of f and $f_{[251]}$. Coefficients of wavelets in Λ_{251} Coefficients of wavelets in Λ_{251}

Figure 5.6.4: Global removal criterion on (x_{251}, y_{251}).



Local view of f and $f_{[800]}$. Coefficients of wavelets in Λ_{800} Coefficients of wavelets in Λ_{800}

Figure 5.6.5: Global criterion on (x_{800}, y_{800}).

and the normal equations of the problem with the point i removed are

$$M_{[i]} d_{[i]} = b_{[i]}, \tag{5.6.6}$$

the relation between them can be expressed as

$$M_{[i]} = M - a_i^T a_i, \tag{5.6.7}$$

and

$$b_{[i]} = b - a_i^T z, \tag{5.6.8}$$

where a_i is the i-th row of the observation matrix A.

This structure makes the construction of the new normal equations a trivial task, and allows for two possible ways two simplify the solution process:

1. Under our assumption of an outlier having only an effect local in scale and space, one expects d to be a good approximation to $d_{[i]}$, so that an iterative

method needs only few iterations to find $d_{[i]}$ having d as starting guess, especially if the matrix M is good conditioned, as it is the case when working with wavelet bases.

2. The inverse of $M_{[i]}$ is available from the inverse of M by means of the Sherman-Morrison formula

$$M_{[i]}^{-1} = (M - a_i^T a_i)^{-1} = M^{-1} + \frac{M^{-1} a_i^T a_i M^{-1}}{1 - a_i M^{-1} a_i^T}. \tag{5.6.9}$$

The direct application of this formula is any case not generally advisable, as it requires the inversion and storage of the inverse or M (or of its QR factorization). This carries a complexity of $O(L^2 log(L))$ (recall $L = \#\Lambda$) for the sparsity structure of M. In any case, this computation would indeed pay off in cases in which adaptivity also pays off, that is, in cases where a big amount of data is describable with a small number of wavelets: $L << N$.

Although we are not bound to always find suitable conditions to apply directly the Sherman-Morrison formula, we still can adapt it to make applicable even in more general conditions, as we explain in Section 5.8.

5.7 Local Criterion

The method described in the above section seems to give a very reliable characterization of outliers. It attains high outlier detection rates in all our experiments. The drawbacks of this method are therefore not in the conceptual sphere, but merely computational, as it can involve a severe cost when recomputing a new approximating function $f_{[i]}$ for every point (x_i, y_i) to be checked.

In this section we present a modification of the above algorithm that uses the orthogonality property of the wavelet basis to simplify the computations.

The basics of the outlier detection was the comparison of f and $f_{[i]}$. This comparison is done locally in space (selecting wavelets located around the location of the outlier) and frequency (picking the higher scales). According to this implicit characterization of outliers based on *local* features, the idea now is to replace the comparison of f and $f_{[i]}$ with a comparison of functions that approximate f and $f_{[i]}$ only locally. To this end, we consider an approximation to the data up to some coarse level $j < J$, where J is the maximal resolution level included in the original set Λ, by simple restriction of the original function f

$$f^j := \sum_{\lambda \in \Lambda;\ |\lambda| \leq j} d_\lambda \psi_\lambda. \tag{5.7.1}$$

Now, we construct local approximations to f and $f_{[i]}$ in the neighborhood of x_i by keeping the coefficients of f up to level j and adding only *local wavelets* to describe

higher detail features. These local wavelets are the ones that we included in the set $\Lambda_{j,i}$ defined in (5.6.2).

We parallel now the method of the previous section by building two approximations, one that takes into account the point, $f^{j,i}$ and one that does not, $f^{j,i}_{[i]}$. That is, we define

$$f^{j,i} := \sum_{\lambda\in\Lambda;\ |\lambda|\leq j} d_\lambda \psi_\lambda + \sum_{\lambda\in\Lambda_{j,i}} d^{j,i}_\lambda \psi_\lambda. \tag{5.7.2}$$

where the vector $d^{j,i} = \{d^{j,i}_\lambda\}_{\lambda\in\Lambda_{j,i}}$ is so computed that

$$\sum_{l=1}^{N} \left(z_l - f^{j,i}(x_l)\right)^2 \tag{5.7.3}$$

attains its minimum. Note that the locality of wavelets included in $\Lambda_{j,i}$ forces the most terms in the summation to be zero, and the non zero elements are directly accessible from the sparsity pattern in which we have coded the observation matrix.

Likewise we define

$$f^{j,i}_{[i]} := \sum_{\lambda\in\Lambda;\ |\lambda|\leq j} d_\lambda \psi_\lambda + \sum_{\lambda\in\Lambda_{j,i}} (d^{j,i}_{[i]})_\lambda \psi_\lambda, \tag{5.7.4}$$

where the vector $d^{j,i}_{[i]} = \{(d^{j,i}_{[i]})_\lambda\}_{\lambda\in\Lambda_{j,i}}$ is so that

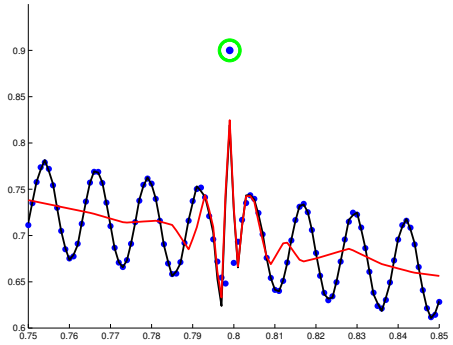
$$\sum_{l\neq i} \left(z_l - f^{j,i}_{[i]}(x_l)\right)^2 \to \text{min!}. \tag{5.7.5}$$

Now, we can define a merit figure based on this local refitting criterion:

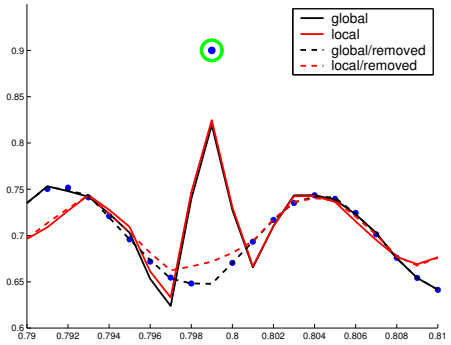
$$\omega_{local}(i) := \log\left(\frac{E^{\Lambda_{j,i}}_{\alpha,p,q}(f^{j,i})}{E^{\Lambda_{j,i}}_{\alpha,p,q}(f^{j,i}_{[i]})}\right). \tag{5.7.6}$$

One can see this process as "freezing" f^j and "gluing" onto it a local approximation to the set $\{(x_i, z_i - f^j(x_i))\}_{i=1,\ldots,N}$. The implicit assumption behind it is that "freezing" and "gluing" will maintain a similar spectrum of local energies. As pointed out above, the assumption is justified by our use of a wavelet basis. The semi-orthogonality property allows us to operate this level surgery, treating the different scales separately.

We can see now an example of this procedure in Figure 5.7.1. On the left, the black line represents the global approximation to the data f and the red line represents the local approximation to the data $f^{3,800}$. We observe that the degrees of freedom used in the construction of $f^{j,i}$ seems to perfectly fit our purposes, as the local approximation near the outlier artifact is nearly indistinguishable from the global approximation, obviously diverging from it outside of this narrow area.



Local view of f and $f^{3,800}$.

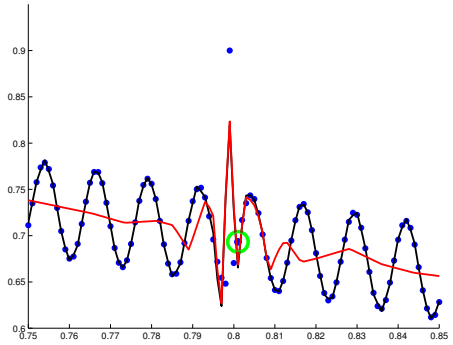


Removal of point in global and local approximations.

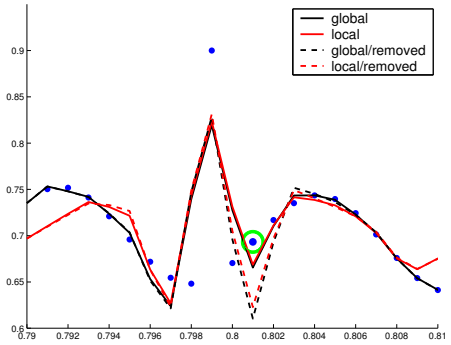
Figure 5.7.1: Global and local reconstructions near (x_{800}, y_{800}).

On the right we see a closer view of the different functions near the point. Solid lines include information of the point and dashed lines do not. We observe that global (black) and local (red) approximations yield similar results (dashed and solid lines diverge near the outlier and approach each other away from it) and, as one could expect in view of the plot, the two outlierness coefficients are correspondingly similar ($\omega_{global}(800) = 1.2$, $\omega_{local}(800) = 1.7$). These coefficients reflect a decrease of more than one order of magnitude in the respective local energies by removal of the point (x_{800}, y_{800}).

The merit figures for the regular neighboring point (x_{802}, y_{802}) are $\omega_{global}(802) = -0.15$ and $\omega_{global}(802) = -0.13$ by the local criterion, indicating that the presence of this point causes just a minor readjustment of the local reconstruction (see Figure 5.7.2), amounting only a slight local energy variation.



Local view of f and $f^{3,802}$.



Removal of point in global and local approximations.

Figure 5.7.2: Global and local reconstructions near (x_{802}, y_{802}).

This procedure eases obviously the computation costs of the global method, as the number of degrees of freedom involved on the computation of $f^{j,i}$ and $f^{j,i}_{[i]}$ is just $\#\Lambda_{j,i}$, a fraction of the total number of wavelets $\#\Lambda$.

A further implicit supposition in this procedure is that the coarse scale projection f^j is in fact not affected by the outlier. That is, the effect of every outlier is restricted to dyadic levels higher than j. This makes the selection of j an important issue. In the global algorithm, the importance of a good selection of j is relative, as it only concerns the way in which the functions f and $f_{[i]}$, are compared, not the functions themselves.

5.8 Local Corrected Criterion

At the end of the previous section we pointed out the main drawback of the local relaxation method: the possibility of the effect of outliers filtering down to the coarse frequencies, so that a local approximation to the data by constructing upon frozen coarse levels cannot unmask the outliers.

The obvious way to circumvent this problem is to construct a coarse level approximation that is not influenced by the point. We seek for a

$$\hat{f}^{j,i} := \sum_{\lambda\in\Lambda;\ |\lambda|\le j} \hat{d^{j,i}}_\lambda \psi_\lambda \tag{5.8.1}$$

so that

$$\sum_{i=1}^{N} \Big(z_l - \hat{f}^{j,i}(x_l)\Big)^2 \to \text{min!}. \tag{5.8.2}$$

The next step is as in the previous section: extend locally the degrees of freedom and compute on this configuration an approximation to the data with and without the point (x_i, z_i),

$$\tilde{f}^{j,i} := \sum_{\lambda\in\Lambda;\ |\lambda|\le j} \hat{d}_\lambda \psi_\lambda + \sum_{\lambda\in\Lambda_{j,i}} \tilde{d}^{j,i}_\lambda \psi_\lambda, \tag{5.8.3}$$

where the vector $\tilde{d}^{j,i} = \{\tilde{d}^{j,i}_\lambda\}_{\lambda\in\Lambda_{j,i}}$ is so computed that

$$\sum_{l} \Big(z_l - \tilde{f}^{j,i}(x_l)\Big)^2 \to \text{min!}. \tag{5.8.4}$$

Likewise we define

$$\tilde{f}^{j,i}_{[i]} := \sum_{\lambda\in\Lambda;\ |\lambda|\le j} \hat{d}_\lambda \psi_\lambda + \sum_{\lambda\in\Lambda_{j,i}} (\tilde{d}^{j,i}_{[i]})_\lambda \psi_\lambda. \tag{5.8.5}$$

where the vector $\tilde{d}^{j,i}_{[i]} = \{(\tilde{d}^{j,i}_{[i]})_\lambda\}_{\lambda\in\Lambda_{j,i}}$ is so that

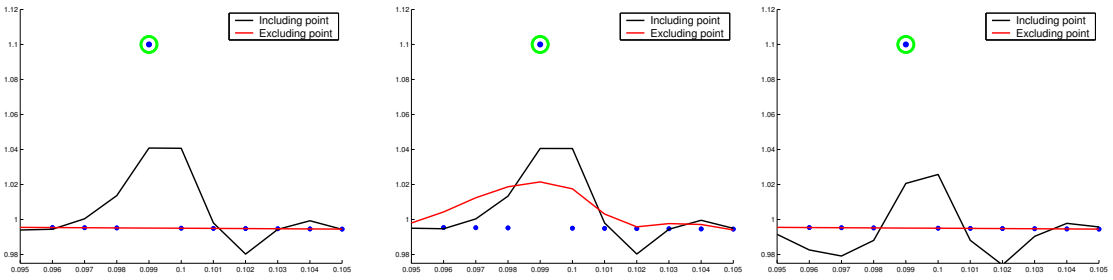
$$\sum_{l\neq i} \Big(z_l - \tilde{f}^{j,i}_{[i]}(x_l)\Big)^2. \tag{5.8.6}$$

Then one can compute the local energies and state a merit figure based in this local corrected fitting of the data like in the previous sections

$$\omega_{local+}(i) := \log\left(\frac{E^{\Lambda_{j,i}}_{\alpha,p,q}(\tilde{f}^{j,i})}{E^{\Lambda_{j,i}}_{\alpha,p,q}(\tilde{f}^{j,i}_{[i]})}\right). \tag{5.8.7}$$

See the plots in Figure 5.8.1. They represent the approximations for the environment of (x_{100}, y_{100}) required by the three methods presented so far, that is global, local and local-corrected from left to right. In the two former j_{cut} is fixed to 7.

The global approximation criterion yields $\omega_{global}(100) = 8.4$. The local approximation criterion gives a much smaller value, $\omega_{local}(100) = 1.7$. The reason can be inferred from the central plot of the figure: the red line, which ideally should not be affected by the outlier, is clearly affected by the presence of it, as it is constructed starting from a coarse-scale approximation to the *whole* data, also including the outlier. We could formulate this effect saying that the prescribed $j_{cut} = 7$ does not correspond to the actual penetration depth of the outlier. One possible solution could be to vary this j_{cut}. The other one is the one presented in this section: the use of an outlier-free coarse-scale approximation. The result is given in the right plot. The red line does not get any undesirable influence from (x_{100}, y_{100}) and succeeds not to produce artifacts in its environment. The outlier coefficient arising from this criterion is 8.3.



Global criterion Local criterion Local corrected criterion

Figure 5.8.1: Comparison of criteria for (x_{100}, y_{100}) with $j_{cut} = 7$.

Obviously, we are re-introducing some amount of computational overhead, as now a $f^j_{[i]}$ has to be computed for every $i \in S$. The point is that this overhead is much more affordable, as it affects a reduced number of coefficients. The observations made in Section 5.6 about point removal procedures become now especially relevant.

5.9 Robust Approximation in High Energetic Environments

In this section we want to envisage some extremal cases of the methods. These refer to the very definition of the outlier. The basic intuition supporting the outlier concept is to be a point whose presence in the data creates an outburst of the local energy of the approximating functions.

This perspective carries the obvious consequence that outliers lying in domain areas where regular other points also can produce similar energy variations would be masked, so that some further refinement might be necessary in the outlier defining criterion.

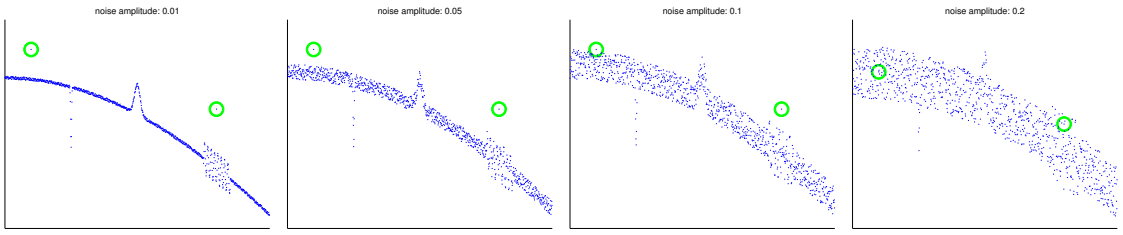
There are several situations in which this effect may occur:

- Outliers embedded in areas of rapid spacial variability.
- Simultaneous presence of other noise sources.
- High rate of outlier contamination.

5.9.1 Presence of Noise

We illustrate this with the following experiment. We add to the data in Figure 5.4.1 some quantity of random noise, whose amplitude is controlled by the parameter σ. We obtain the data distributions of Figure 5.9.1 for different noise amplitudes.

In cases $\sigma = 0.01$ and $\sigma = 0.05$ the two outliers (x_{100}, y_{100}) and (x_{800}, y_{800}) are perfectly distinguishable from the noisy background, and it is expected that the criteria mentioned above will give good estimations. In the case $\sigma = 0.1$ (x_{100}, y_{100}) is embedded into the noise and a correct outlier finding criterion should not point it out as an outlier, whereas (x_{800}, y_{800}) should be marked. Finally, in the case $\sigma = 0.2$ both original outliers are included in the point cloud and no outlier identification would make sense.

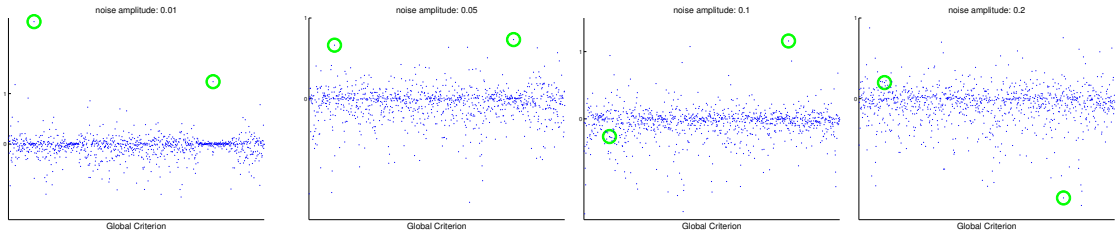


$\sigma = 0.01$ $\sigma = 0.05$ $\sigma = 0.1$ $\sigma = 0.2$

Figure 5.9.1: Noisy data.

Correspondingly with the above description of rough features of the data we get the global criterion given in Figure 5.9.2 for the whole bunch of points. We see

that in the two extreme cases the criteria works well. $\sigma = 0.01$ works out well, discrimination of outliers is not so clear as in the previous case. The results for the case $\sigma = 0.2$ also fit the idea of outlierness as the criterion does not recognize any special feature in the marked points, accordingly to their inclusion in the noisy background.



$\sigma = 0.01$ $\sigma = 0.05$ $\sigma = 0.1$ $\sigma = 0.2$

Figure 5.9.2: ω_{global} for data sets in Figure 5.9.1.

The intermediate cases also work: At $\sigma = 0.05$ both outliers are marked, where as at $\sigma = 0.1$ point (x_{100}, y_{100}) is correctly ignored by the criterion, as it is embedded in the surrounding noise, and (x_{800}, y_{800}) is also successfully provided with a large value of ω_{global}.

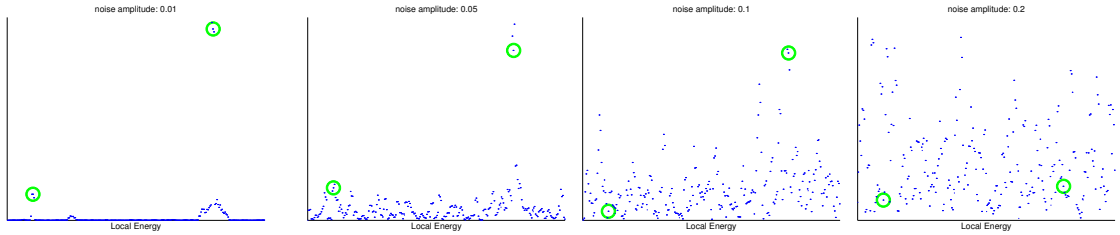
However in these cases the discrimination of outliers is not so clear as in the previous case: observe that a number of regular data points also attain a large outlierness coefficient. It is not a mayor problem, by very definition a false detection amounts to eliminate from the data a point that carries redundant information. As long as the remaining points still reproduce the whole set of significant data features, the loss of a moderate number of data points can be admissible. This issue is revisited in the following Sections.

Anyway, the criterion can be refined in order to reduce the loss of actual information. In this case, false detections affect points whose removal of the data originates a noticeable decrease of the local energy. As our criterion measures this decrease in relation to the original energy, in areas where this is very small, the local reconfiguration of the wavelet spectrum after removal of a data point can happen to produce a still lower local energy, without this decrease being significant.

We can cope with this situation in different ways. Firstly, we can impose stricter thresholding policies in the processing step of Section 5.5 to rule out points lying in flatter areas. In the present case, where the data is corrupted by high-frequency noise one should filter it with classical wavelet smoothing procedure, as long as one has a statistical model for this noise.

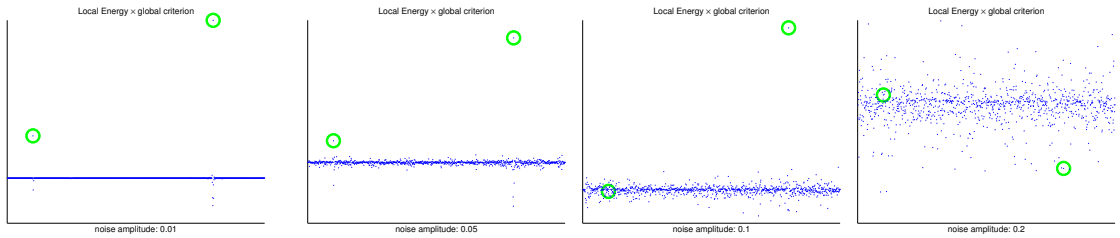
A second strategy would be to simply build the local energy factor into the criterion.

Let us denote by $e_{loc}(i)$ the local energy defined in (5.6.3) for the influence set of the point x_i. In 5.9.3 we see the values of this local energy, for each point of the



$\sigma = 0.01$ $\sigma = 0.05$ $\sigma = 0.1$ $\sigma = 0.2$

Figure 5.9.3: e_{loc} profile for data sets in Figure 5.9.1.



$\sigma = 0.01$ $\sigma = 0.05$ $\sigma = 0.1$ $\sigma = 0.2$

Figure 5.9.4: Profile of the product $\omega_{global} \cdot e_{loc}$ for data sets in Figure 5.9.1.

data set in the four level-of-noise situations.

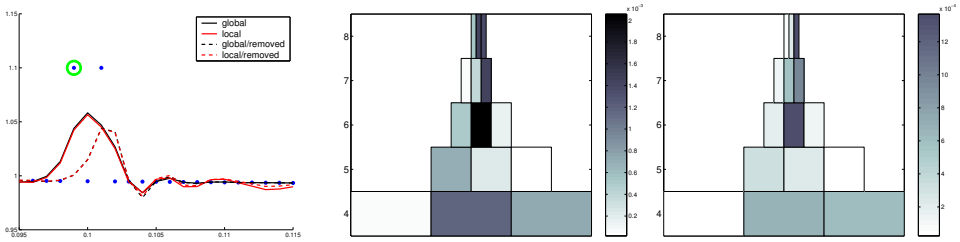
If we multiply the global criterion profile with the local energy profile we get the plots of Figure 5.9.4, where the discrimination of outliers appears much clearer than in Figure 5.9.2.

5.9.2 Large Number of Outliers

Other possible source of problems is the outlier density. The capability of the method to disentangle an outlier from the surrounding signal lies at the end in a characterization of the local energy of this surrounding signal. If further outliers are present in the surroundings, this characterization fails, and consequently the outlier marking criterion as well.

We see an example in Figure 5.9.5. We add to the original data a new outlier by imposing the value $y_{102} = 1.1$. This represents an outlier in the immediate neighborhood of the original outlier (x_{100}, y_{100}). If we compute now the outlierness profile for each of these points, we find that the outlierness coefficient of (x_{100}, y_{100}) in this data set is 0.3, computed by both global and local criteria. Recall that this point attained an outlierness coefficient of 7.2 when it was isolated. The reason for these low values can be read from the reconstructions given in the figure: the suppression of the outlier (x_{100}, y_{100}) does not relax locally the approximation, as

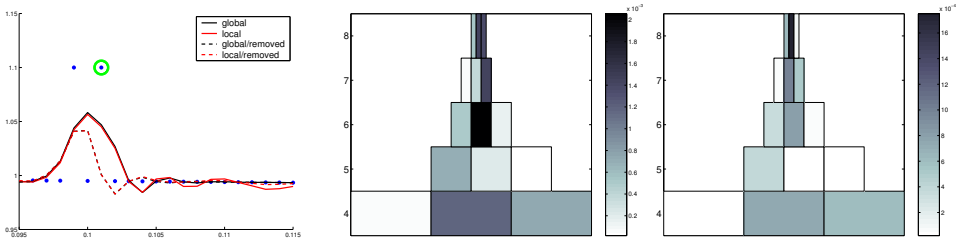
the remaining outlier (x_{102}, y_{102}) still twists the approximation toward himself.



Local view behavior of approximants | Coefficients of wavelets in Λ_{100} | Coefficients of wavelets in Λ_{100} after removal of (x_{100}, y_{100})

Figure 5.9.5: Proximity of outliers: analysis on (x_{100}, y_{100}).

There is indeed some energy decay, as one can deduce from the wavelet spectrum in the central and right plots of the same Figure, but not as severe as when no further outlier corrupted the background: compare Figure 5.9.5 with Figure 5.6.1.



Local view behavior of approximants | Coefficients of wavelets in Λ_{102} | Coefficients of wavelets in Λ_{102} after removal of (x_{102}, y_{102})

Figure 5.9.6: Proximity of outliers: analysis on (x_{102}, y_{102}).

This means that the number of outliers that can be present in a data set without corrupting it depends (evidently as well as on the data set) on their distribution profile, as it is the proximity of outliers to each other which drives the method to fail.

We perform the following series of experiment: we corrupt the original data by a fixed number M of outbursts for two different designs. In one design, the outliers are equidistantly placed and in the other one, outliers are randomly distributed. We run then our outlier finding procedure for different choices of M.

In Table 5.9.1 we give the percentages of successful outlier detection for different choices of τ and different number of outliers in equidistant design. The same information for randomly distributed outliers is given in Table 5.9.2. The percentages of

false detection for the same choices of M and τ are given in Table 5.9.3 (equidistant design) and Table 5.9.4 (random design).

τ	M=10	M=20	M=50	M=100	M=200	M=300	M=500
0.1	100.0	100.0	100.0	98.0	42.0	4.3	0.2
0.2	100.0	100.0	100.0	98.0	38.5	1.0	0.2
0.3	100.0	100.0	100.0	97.0	17.5	0.3	0.0
0.4	100.0	100.0	100.0	96.0	3.0	0.0	0.0
0.5	100.0	100.0	96.0	96.0	0.5	0.0	0.0

Table 5.9.1: Outlier detection percentage in equidistant design.

τ	M=10	M=20	M=50	M=100	M=200	M=300	M=500
0.1	80.0	70.0	82.0	70.0	45.5	46.0	16.4
0.2	70.0	70.0	72.0	61.0	35.5	30.3	8.2
0.3	60.0	65.0	64.0	55.0	26.0	21.3	3.2
0.4	60.0	65.0	60.0	47.0	20.0	15.7	2.2
0.5	60.0	65.0	52.0	41.0	16.5	11.3	1.8

Table 5.9.2: Outlier detection percentage in random design.

τ	M=10	M=20	M=50	M=100	M=200	M=300	M=500
0.1	0.3	0.5	1.2	0.0	2.7	18.0	0.0
0.2	0.0	0.0	0.0	0.0	0.9	11.7	0.0
0.3	0.0	0.0	0.0	0.0	0.0	6.8	0.0
0.4	0.0	0.0	0.0	0.0	0.0	5.5	0.0
0.5	0.0	0.0	0.0	0.0	0.0	4.3	0.0

Table 5.9.3: False detection percentage in periodic design.

The results are quite expectable. In the equidistant case the outliers are located quite well when they are distanced (low values of M), up to the critical distance in which every outlier suffers the influence of two neighbors and the method collapses abruptly. In the random design, the outlier detection rate is not so successful for the small concentration of outliers. This is caused by the number of outliers that result to occur close to each other, in spite of a low total number of outliers. In compensation, the method attains a higher detection rate when the total number of outliers is larger, as a number of outliers occurs isolated from the others.

τ	M=10	M=20	M=50	M=100	M=200	M=300	M=500
0.1	0.0	0.7	1.4	1.5	3.1	3.4	6.4
0.2	0.0	0.0	0.2	0.6	1.0	1.4	3.0
0.3	0.0	0.0	0.1	0.3	0.3	0.5	1.4
0.4	0.0	0.0	0.1	0.2	0.2	0.3	0.7
0.5	0.0	0.0	0.1	0.1	0.1	0.1	0.3

Table 5.9.4: False detection percentage in random design.

5.10 Energy Criterion

The three methods explained until now are based on the construction of a couple of functions (one that sees the whole action of the possible point, and one that mollifies it) and its comparison.

At the end, the rational for this criteria is provided by the Riesz Basis property of wavelets, that leads the description of the norm of a wide range of spaces in terms of the wavelet coefficients, as exploited in the definition (5.6.3). In the criteria (global, local and local corrected) explained so far we compared $e_{loc}(f)$ and $e_{loc}(f_i)$, for some appropriate definition of f_i. We call these "direct" criteria. However we could use the same rational provided by the Riesz Basis property to propose the use of $e_{loc}(f - f_i)$ as an outlier finding criteria. This is also a natural choice, and we call them "residual criteria", according to choice of f_i. In any case, there are some differences to the previous strategy.

- Variable order of magnitude of adequate thresholding parameters.

  In direct methods the order of magnitude of the τ parameter appears to be quite intuitive: the energy change must be numerically noticeable. In the residual methods, on the contrary, one usually finds appropriate values for τ, but they are obviously very sensitive to the data and the underlying function.

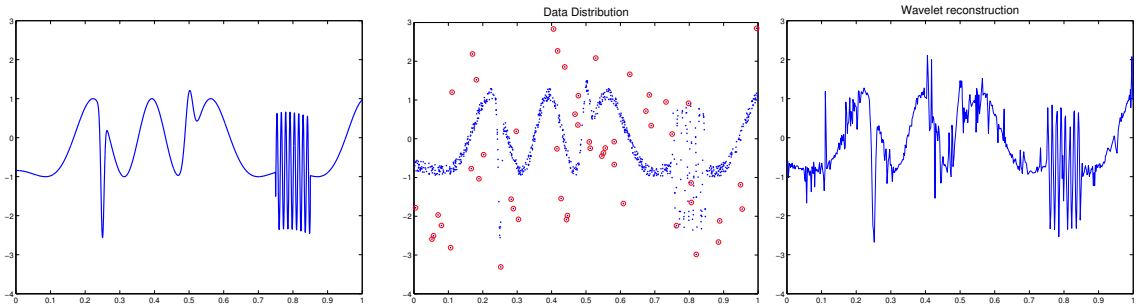
- Different performance.

  If the addition of an outlier imposes a *redistribution* of local energy rather that an *increase* of it (see Section 5.9 for situations in which this may occur), the direct methods will fail to detect it, as explained above, but residual methods do still have a chance. The prize for it is a higher risk to eliminate correct data.

Consider for instance the function on the right of Figure 5.10.1. We add a background noise overall and a 5% of outliers randomly distributed of diverse amplitude, as plotted in the center of the Figure. The Least Squares method without outlier detection gives the reconstruction on the right.

If we compute our full set of criteria on our reconstruction, we get the successful detection and wrong elimination percentages given in Table 5.10.1 for direct methods, and in Table 5.10.2 or residual methods, whose figures confirm our expectations. Also the reconstructions given in Figure 5.10.2 behave as expected. Direct methods fail to find the full set of outliers. Residual methods filter more outburst but cannot avoid throwing away more data points and possibly relevant information. Compare the reconstruction of the high energetic feature located at $x = 0.8$, which appears much more damaged in the right plot than in the left.

As a final remark, note that the use of $f - f_i$, that is, residual methods, would entitle us to use splines as ansatz basis, as the measure of $f - f_i$ in L_2 could be reasonably understood as an indicator for outlier presence, whereas direct methods are only meaningful in a wavelet ansatz.



Original function | Irregular sampling with noise and outliers | Wavelet reconstruction

Figure 5.10.1: Model data set 2.

	% detected outliers			% eliminated data		
τ	Global	Local	Local +	Global	Local	Local +
0.000	75.0	73.1	75.0	34.7	35.7	35.0
0.050	69.2	67.3	67.3	11.1	9.6	9.4
0.100	65.4	59.6	63.5	5.9	5.6	5.6
0.200	63.5	51.9	51.9	2.5	3.1	3.3
0.300	57.7	46.2	44.2	1.7	2.0	1.6
0.400	42.3	40.4	36.5	1.2	1.3	1.0

Table 5.10.1: Performance of direct methods in the analysis of the outlier-corrupted data from Figure 5.10.1.

	% detected outliers			% eliminated data		
τ	Global	Local	Local +	Global	Local	Local +
1.0e+05	100.0	100.0	100.0	75.7	76.9	77.5
5.0e+06	96.2	98.1	98.1	28.5	29.3	29.5
1.0e+07	92.3	90.4	92.3	22.4	23.1	23.6
1.5e+07	90.4	88.5	88.5	19.5	19.2	19.4
5.0e+07	76.9	65.4	69.2	10.6	10.5	10.8
1.0e+08	65.4	57.7	61.5	6.7	6.3	7.3
1.5e+08	57.7	46.2	50.0	5.3	4.7	5.5
2.0e+08	42.3	36.5	38.5	4.6	3.9	4.8

Table 5.10.2: Performance of residual methods in the analysis of the outlier-corrupted data from Figure 5.10.1.

5.11 Several Dimensions

The method can be extended naturally to higher dimensions. We see an example of the procedure with the data in Figure 5.11.1. At the right plot we see a view of the well known Franke function. We will try to reconstruct it from the 2000 scattered random sampling points given in the central plot. In the 20 points marked in red we will add a constant value, creating thereby a random distribution of outliers in the original data. We see in the right plot a wavelet reconstruction found by the Least Squares method, where we can observe how the presence of the outbursts creates local oscillations in the surface in the proximity of each outlier.

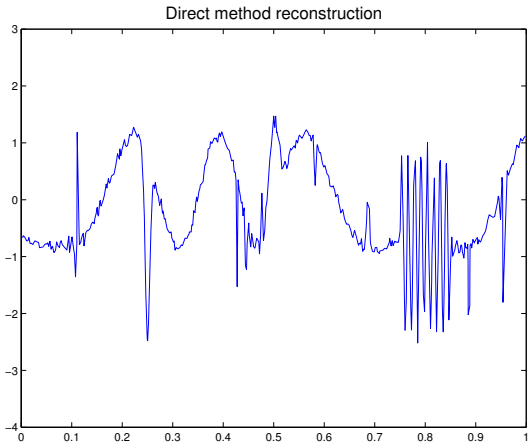
To assert the performance of the method in this data set we compute the percentage of found outliers and false detections for the several criteria we have discussed.

In Table 5.11.1 we see the results obtained by the direct methods. The results assert the likeliness of the three criteria. The three of them give a successful rate of outlier finding with minor losses of non-corrupted data-points. Note, however, that the method does not attain the complete detection of outliers, as the outlier interaction effect described in Section 5.9 attain to mask a couple of them. A second run of this method on the data after removal of the outliers detected in the first run detects successfully the remaining outliers.

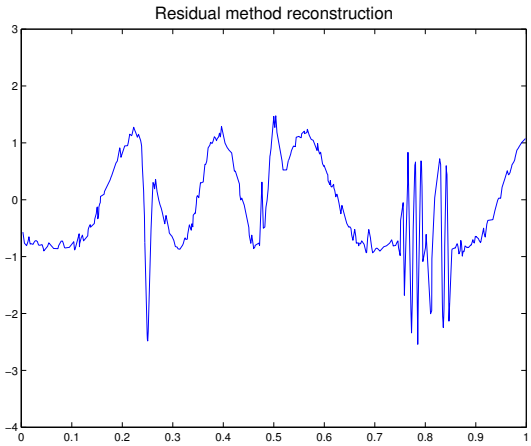
According to our argumentation in Section 5.10, the residual methods can disentangle better this interaction and consequently do detect all the outliers in just one run. The price of it is a slightly higher rate of false detections.

A further example is provided in the analysis of a geophysical data set from [85]. The set includes 18634 points ordered in a square grid, plotted in the right of Figure 5.11.2. We add 1000 randomly distributed outbursts to this data, yielding the data on the left of the same Figure.

In the left of Figure 5.11.3 we can see visually the performance of the method

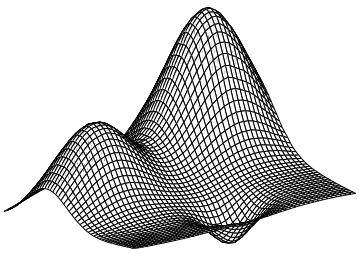


Data cleaned following a direct method

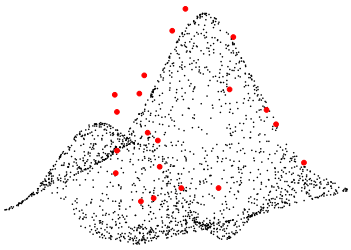


Data cleaned following a residual method

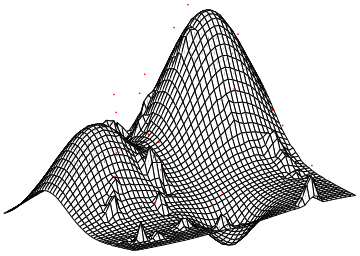
Figure 5.10.2: Reconstructions of example data set 2 after removal of points marked as outliers.



Franke data



Sampling geometry with outliers (red points)



Wavelet reconstruction without outlier detection

Figure 5.11.1: Outlier distribution in a scattered sampling of the Franke data.

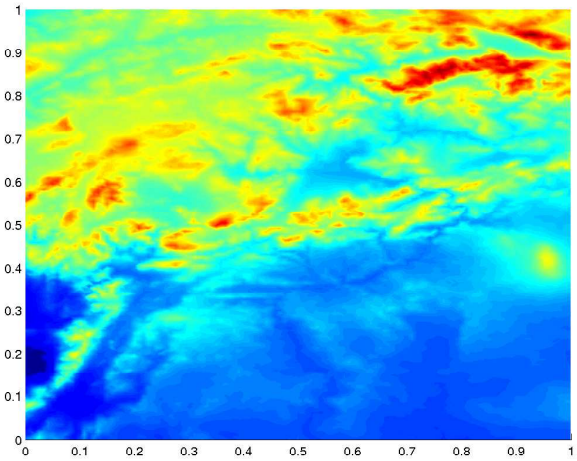
with one run. A first run eliminates 75% of the outliers, while the data eliminated by false detection does not appear to damage the reconstruction. A second run of the method, that is, a run on the cleaned data, offers the reconstruction on the right of the Figure. As we start from a situation where the density of outliers has been reduced, further outliers that were previously masked by neighboring ones have been now successfully detected.

	% detected outliers			% eliminated data		
τ	Global	Local	Local +	Global	Local	Local +
0.000	90.0	90.0	90.0	34.2	35.1	34.5
0.025	90.0	90.0	90.0	1.2	2.0	2.0
0.050	85.0	85.0	85.0	0.5	0.4	0.4
0.075	85.0	85.0	85.0	0.3	0.2	0.2
0.100	85.0	85.0	85.0	0.1	0.1	0.1
0.125	75.0	80.0	80.0	0.1	0.1	0.1
0.150	70.0	75.0	80.0	0.1	0.0	0.0
0.175	70.0	70.0	70.0	0.0	0.0	0.0
0.200	65.0	70.0	70.0	0.0	0.0	0.0

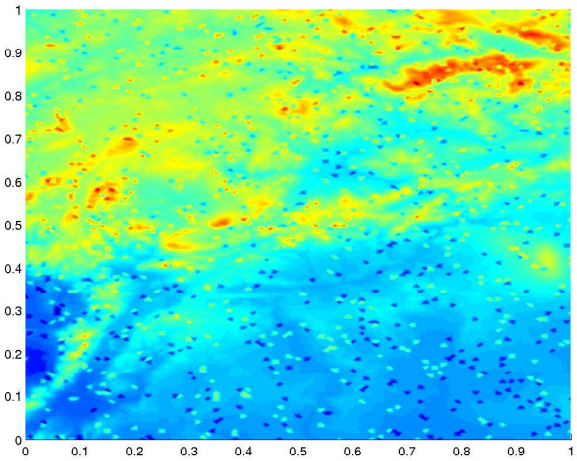
Table 5.11.1: Performance of direct criteria in the analysis of the outlier-corrupted data from Figure 5.11.1.

	% detected outliers			% eliminated data		
τ	Global	Local	Local +	Global	Local	Local +
0.000	100.0	100.0	100.0	99.0	99.0	99.0
0.500	100.0	95.0	100.0	3.1	1.1	1.9
1.000	95.0	90.0	95.0	0.9	0.4	0.8
2.000	85.0	80.0	90.0	0.6	0.2	0.4
3.000	75.0	70.0	80.0	0.5	0.1	0.4
4.000	75.0	60.0	75.0	0.4	0.1	0.4
5.000	65.0	55.0	70.0	0.4	0.1	0.4

Table 5.11.2: Performance of residual criteria in the analysis of the outlier-corrupted data from Figure 5.11.1.

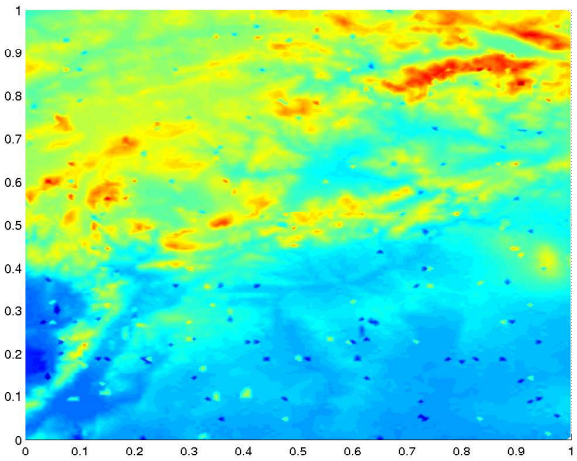


Vertical view.

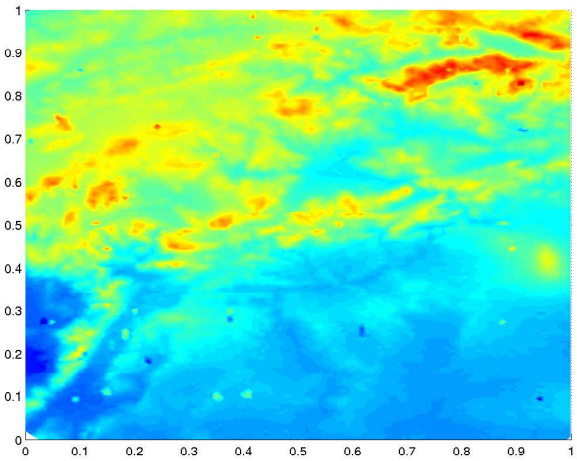


Data set corrupted with 1000 outliers (% 5.6 of the data)

Figure 5.11.2: Geophysical data set: 18605 points.



First run



Second run

Figure 5.11.3: Reconstruction of data in Figure 5.11.2 after robust cleaning.

Chapter 6

Application to PDEs: Evaluation of Nonlinear Terms

In this chapter we propose the application of the data fitting methodology to the numerical solution of certain nonlinear PDE problems.

In the first section, we will briefly discuss the use of adaptive wavelet strategies in operator equations, with a special emphasis in some recent methods which show optimal approximation rates and optimal complexity, in a sense to be explained later.

The application of these recent developments to nonlinear problems requires the evaluation of terms of the type $\{\langle \psi_\lambda, \mathcal{G}(u)\rangle\}_{\lambda\in\Lambda}$, where u is a finite linear combination of wavelets, Λ is a finite set of wavelet indices, $\mathcal{G}$ is a nonlinear function, and $\langle\cdot,\cdot\rangle : \mathcal{H}\times\mathcal{H}' \to I\!R$ is a duality product for a Hilbert space $\mathcal{H}$ and its dual $\mathcal{H}'$. We will describe how the data fitting algorithm provides a simple method to perform the computation of these terms.

In the second section, we use this technique of evaluation of nonlinearities in wavelet coordinates by Least Squares approximation in a numerical study of the behavior of the solution of a nonlinear evolution equation, the well known Burgers' equation, by means of a wavelet discretization.

6.1 Wavelets and Differential Operator Equations

In the last years the use of wavelets in scientific computing has spread beyond Signal Analysis and Image Processing and became a subject of intense research of the PDE community as some properties of wavelets find special relevance in the Numerical Analysis: the possibility to characterize a wide variety of functional spaces (see e.g. [58],[119]), the good conditioning of systems arising in stationary elliptic problems (see e.g. [47],[102]), and specially the simultaneous locality in frequency and space, a property that makes wavelets a natural choice for construction of adaptive algorithms. We will focus on this last point and describe in the following the most relevant facts for the monovariate case.

All along this chapter, we will be using the notational convention of C being any positive constant independent of any indexing variables. We will also write $A \lesssim B$ to indicate that such a C can be found for the relation $A \leq CB$. Moreover, we will write $A \sim B$ meaning that $A \lesssim B$ and $B \lesssim A$.

6.1.1 Approximation of Functions in Wavelet Bases

We shall briefly recall some results from the theory of nonlinear approximation which will help us gaining understanding about what exactly are the benefits that one can expect from an adaptive approximation of a PDE problem.

Linear and Nonlinear Approximation

We consider first approximation in *linear* spaces. Consider, for instance, a uniform discretization of the interval $(0, 1)$ into subintervals of diameter h and denote by V_h the space of piecewise linear continuous functions defined with respect to this partition. The best approximation u_h to a function u from the Sobolev space H^α

$$\|u - u_h\|_{L_2} := \inf_{g_h \in V_h} \|u - g_h\|_{L_2} \tag{6.1.1}$$

fulfills the following classical estimate for the rate of approximation as h tends to zero (see [13] or [32])

$$\|u - u_h\|_{L_2} \lesssim h^\alpha \|u\|_{H^\alpha} \tag{6.1.2}$$

for $\alpha \leq 2$. For $h = 2^{-j}$, $j = 1, \ldots$, the (preorthogonal linear spline-) wavelets described in Chapter 1 provide bases for these spaces,

$$V_{2^{-j}} = span\{\psi_\lambda\}_{\lambda \in \Lambda_j},\ \Lambda_j = \{\lambda : \lambda \in I\!I, |\lambda| \leq j\}. \tag{6.1.3}$$

As the number N of basis elements in each $V_{2^{-j}}$ is proportional to 2^j, one can write

$$\|u - u_{2^{-j}}\|_{L_2} \lesssim N^{-\alpha} \|u\|_{H^\alpha}. \tag{6.1.4}$$

The approximation rate is thus governed by the Sobolev regularity α of the target function $u \in H^\alpha$. This constitutes a weakness of this type of approximation for functions with singularities (i.e., localized areas of the domain where the function or its gradient tend to infinity) which have small Sobolev regularity regardless of their possible smoothness away from the singularities.

This motivates the introduction of other finer tools for measuring smoothness: the *Besov* spaces $B^\alpha_{p,q}$ introduced in [139], which are endorsed with the norm

$$\|f\|_{B^\alpha_{p,q}} := \|f\|_{L_p} + \left(\sum_{j>0} \left(2^{\alpha j} \omega_m(f, 2^{-j}) \right)^q \right)^{\frac{1}{q}} \tag{6.1.5}$$

with $m \in I\!N, m > \alpha$. Here, $\omega_m(f,t)_p$ is the L_p modulus of smoothness of m-th order, defined as

$$\omega_m(f,t)_p := \sup_{|h|\leq t} \left\| \sum_{k=0}^{m} \left(\begin{array}{c} m \\ k \end{array} \right) (-1)^k f(\cdot - kh) \right\|_{L_p} . \tag{6.1.6}$$

As we advanced in Section 4.1, this norm can be represented by a norm equivalence that relates the norm of a function $f = \sum_{(j,k)\in I\!I} d_{j,k}\psi_{j,k}$ in $B^{\alpha}_{p,q}$ with the coefficients of its wavelets expansion by

$$\|f\|_{B^{\alpha}_{p,q}} \sim \left(\sum_{j>0} 2^{jq\left(\alpha+\frac{1}{2}-\frac{1}{p}\right)} \left(\sum_{k=0,\ldots,2^j-1} |d_{j,k}|^p \right)^{\frac{q}{p}} \right)^{\frac{1}{q}} , \tag{6.1.7}$$

which generalizes the norm equivalence (4.1.3) seen in the context of regularization in Sobolev spaces. Roughly speaking, this equivalence relation essentially holds when the members of the wavelet family themselves have regularity $r > \alpha$ measured in L_p. Details can be found in e.g. [33], [58] or [59].

We define now the best *N-term* approximation to an element u of the Hilbert space $\mathcal{H}$ with a basis $\{\psi_\lambda\}_{\lambda\in I\!I}$ as an element u_N from the *nonlinear* manifold

$$\Sigma_N := \left\{ \sum_{\lambda\in\Lambda} d_\lambda \psi_\lambda;\ \Lambda \subset I\!I;\ \leq N \right\}. \tag{6.1.8}$$

that realizes the best N-term error

$$\sigma_N(u) := \inf_{g\in\Sigma_N} \|u - g\|_{\mathcal{H}}. \tag{6.1.9}$$

A rich theory (see e.g. [58],[60]) develops the intimate connection between wavelets, Besov spaces and this kind of approximation. We will present in the following the two main points: the *characterization of approximation spaces* by means of the norm equivalence relations and the *thresholding* technique.

Characterization of Approximation Spaces

The use of the norm equivalence relations is a powerful tool to characterize the classes of functions that can be approximated with a given accuracy. For a simple example, consider the behavior of the best N-term error σ_N measured in L_2, and assume for simplicity that $\{\psi_\lambda\}_{\lambda\in I\!I}$ is a orthonormal basis. The best N-term approximation of a function f is in this case achieved by keeping the N largest coefficients of its wavelet expansion $\sum_{\lambda\in I\!I} d_\lambda\psi_\lambda$. Now we introduce an indexing $\{d_m\}_{m\geq 1}$ that represents an *decreasing rearrangement* of the absolute values of $\{d_\lambda\}_{\lambda\in I\!I}$, so that d_m is the m-th largest $|d_\lambda|$. With this rearrangement, the error fulfills

$$\sigma_N(f) = \left(\sum_{m>N} d_m^2 \right)^{1/2} . \tag{6.1.10}$$

Moreover, we can bound each component by

$$md_m^\tau \leq \sum_{k=1}^{m} d_k^\tau \leq \sum_{k\geq 1} d_k^\tau \tag{6.1.11}$$

for all $m = 1, \ldots$ and for all $\tau \geq 1$. If $f \in B^\alpha_{\tau,\tau}$ with $1/\tau = 1/2 + \alpha$, the norm equivalence relation (6.1.7) yields

$$\|f\|_{B^\alpha_{\tau,\tau}} \sim \left(\sum_{\lambda \in I\!I} |d_\lambda|^\tau \right)^{1/\tau} \tag{6.1.12}$$

and we get the decay rate

$$d_m \lesssim m^{-1/\tau} \|f\|_{B^\alpha_{\tau,\tau}} \tag{6.1.13}$$

for the rearranged coefficients. We get thus the following bound for the approximation error

$$\sigma_N(f) \lesssim \left(\sum_{m\geq N} m^{-\frac{2}{\tau}} \right)^{1/2} \|f\|_{B^\alpha_{\tau,\tau}} \lesssim N^{\frac{1}{2}-\frac{1}{\tau}} \|f\|_{B^\alpha_{\tau,\tau}} \sim N^{-\alpha} \|f\|_{B^\alpha_{\tau,\tau}}. \tag{6.1.14}$$

In fact, one can extend this result to measure the error in Sobolev norms. Defining

$$\sigma_{N,t}(g) := \inf \left\{ \|g - \sum_{\lambda\in\Lambda} d_\lambda \psi_\lambda\|_{H^t},\ \#\Lambda \leq N \right\}, \tag{6.1.15}$$

one can get a full characterization of the best N-term approximation of a function in terms of its Besov regularity:

Proposition 6.1.1. *([43]) Under adequate assumptions on the wavelet family used in (6.1.15), and assuming $\frac{1}{\tau} = \alpha - t + \frac{1}{2}$, the best N-term approximation to a function v relative to $\|\cdot\|_{H^t}$ fulfills*

$$\sum_{N=1}^{\infty} \left(N^{\alpha-t} \sigma_{N,t}(v) \right)^\tau \frac{1}{N} < \infty \tag{6.1.16}$$

if and only if $v \in B^\alpha_{\tau,\tau}$.

Thresholding

For usual error norms like L_p, $W^{\alpha,p}$ or $B^\alpha_{p,q}$ the use of a wavelet basis allows to get *near optimal* approximations by simple thresholding in the following sense: For a $u = \sum_{\lambda\in I\!I} d_\lambda \psi_\lambda$, the approximation $T_N(u) := \sum_N \text{largest}_{\|d_\lambda\psi_\lambda\|_{\mathcal{H}}} d_\lambda \psi_\lambda$ constructed by selecting the N largest contributions of the wavelet expansion is comparable to the best N-term approximation u_N

$$\|u - T_N(u)\|_{\mathcal{H}} \lesssim \|u - u_N\|_{\mathcal{H}}. \tag{6.1.17}$$

Adaptivity

If we compare (6.1.4) obtained with a linear approximation technique with (6.1.14), obtained with the best N-term approximation by redistribution of degrees of freedom with respect to a nonuniform grid, we see that both techniques yield the same approximation rate. The difference is the space to which the target function is required to belong in order to attain the corresponding approximation rate. This is the point in using Besov spaces, as Besov spaces of smoothness index α are substantially larger than the corresponding H^α spaces (see [58]). For instance, functions in H^α are necessarily continuous for $\alpha > 1/2$, whereas the Besov space $B^\alpha_{\tau,\tau}$ with $1/\tau = 1/2 + \alpha$ contains discontinuous functions for arbitrary values of α.

In such situations, Besov regularity is a much more appropriate tool to measure smoothness. The basic idea is that functions with singularities typically have small Sobolev regularity but still offer a high Besov regularity ([44], [58]). This is the case, for instance, of the solution of scalar elliptic problems with singularities on the right hand side, or defined on a polygonal shaped domain, as explained in [42] and [44].

6.1.2 Adaptive Wavelet Schemes

The results of the nonlinear approximation theory express the *ideal* approximation rate that one can attain using an adaptive scheme. They constitute the benchmark to which one should measure the performance of such an approach. We discuss now how usual adaptive strategies work in a wavelet setting.

A unified point of view

In the numerical treatment of stationary problems of the form

$$\mathcal{F}(u) = 0 \tag{6.1.18}$$

or even evolution problems like

$$\frac{\partial u}{\partial t} = \mathcal{F}(u) \tag{6.1.19}$$

one typically constructs a succession of approximations $u^n = \sum_{\lambda \in \Lambda^n} {d_\lambda}^n \psi_\lambda$ that converges to u as n goes to infinity. As noted in [37], the usual setting for the construction of an approximation u^{n+1} and its wavelet configuration Λ^{n+1} from the previous ones are *dynamically adaptive schemes* introduced in [115], which exploit the fact that wavelet coefficients convey fine information on the local regularity of a function, (see e.g. [1], [8], [9], [21], [40], [49], [121]). Generally, this task is done in three steps:

- Refinement: use an a-posteriori analysis to enlarge the old configuration Λ^n to an extended configuration $\tilde{\Lambda}$.
- Computation: compute a new solution $\tilde{u}$ on the enlarged configuration $\tilde{\Lambda}$.

- Coarsening: threshold the smallest coefficients of $\tilde{u}$ to produce the new approximation u^{n+1}.

Note that this methodology follows the same strategy as in the data fitting algorithms presented in Chapter 2 of this work. We will be using this strategy later in Section 6.2.3 in the context of a nonlinear evolution equation.

An optimal algorithm

A special mention deserves the methodology explained in [34] for it describes a constructive way to find approximative solutions for elliptic operator equations of the type (6.1.18) that recover asymptotically the best approximation rate attainable by a wavelet scheme. We will sketch now how this methodology works.

As a guiding example, consider the boundary value problem

$$\mathcal{F}(u) = -\Delta u - f = 0 \text{ in } (0,1),\ u(0) = u(1) = 0, \tag{6.1.20}$$

whose variational formulation in H_0^1 is

$$\int_0^1 \nabla u \cdot \nabla v dx = \int_0^1 f v dx \tag{6.1.21}$$

for all $v \in H_0^1$. Defining the linear operator $\mathcal{A}$ by $\langle \nabla v, \nabla u \rangle = \langle v, \mathcal{A}u \rangle$, where $\langle \cdot, \cdot \rangle$ is to be understood as the dual form on $H_0^1 \times H^{-1}$ induced by the standard L_2-inner product, the boundary problem (6.1.20) can be expressed as the equivalent operator equation

$$\mathcal{A}u = f \tag{6.1.22}$$

where the homogeneous boundary conditions have been incorporated into $\mathcal{A}$. We stop here to discuss the *well-possedness* of the problem in the following sense:

Definition 6.1.2. *An operator equation of the form (6.1.22), with $\mathcal{A}$ mapping $\mathcal{H}$ onto $\mathcal{H}'$ is called well posed, if there exist two finite constants $c_{\mathcal{A}}, C_{\mathcal{A}} > 0$ such that*

$$c_{\mathcal{A}} \|v\|_{\mathcal{H}} \leq \|\mathcal{A}v\|_{\mathcal{H}'} \leq C_{\mathcal{A}} \|v\|_{\mathcal{H}} \, \forall v \in \mathcal{H}. \tag{6.1.23}$$

Obviously, (6.1.23) ensures that for any right hand side $f \in \mathcal{H}'$, there is a unique solution u to (6.1.22) which depends continuously on the data f. The problem (6.1.22) is well-posed in this sense, which can be easily checked.

Now, one can transform (6.1.22) into an equivalent operator equation in ℓ_2 which is also well posed. To this end, we recall that a family $\Psi := \{\psi_\lambda\}_{\lambda \in I\!I}$ of wavelets is said to have the Riesz basis property for a Hilbert space $\mathcal{H}$ if every element $v \in \mathcal{H}$ possesses a unique expansion in terms of Ψ and there exist finite positive constants $c_\Psi, C_\Psi > 0$ such that

$$c_\Psi \|\mathbf{v}\|_{\ell_2} \leq \|v\|_{\mathcal{H}} \leq C_\Psi \|\mathbf{v}\|_{\ell_2} \, \forall v \in \mathcal{H} \tag{6.1.24}$$

holds. An important remark is that, wherever (6.1.24) holds, one has for the dual product $\langle \cdot, \cdot \rangle : \mathcal{H} \times \mathcal{H}' \to I\!R$ the equivalence

$$C_\Psi^{-1} \|\langle \Psi, v \rangle\|_{\ell_2} \leq \|v\|_{\mathcal{H}'} \leq c_\Psi^{-1} \|\langle \Psi, v \rangle\|_{\ell_2} \, \forall v \in \mathcal{H}' \tag{6.1.25}$$

where the λ-th component of the vector $\langle \Psi, v \rangle$ is the dual form $\langle \psi_\lambda, v \rangle$.

Recall that the family of preorthogonal spline–wavelets presented in Chapter 1 has the Riesz basis property for $\mathcal{H} = L_2$. In fact, an intense research (see e.g. [22],[39],[50],[52]) has been devoted to the construction of families of wavelets that use the Riesz basis property for L_2 as an anchor to establish the validity of this property for a wider scale of Sobolev spaces. For such families, one can find positive constants $\gamma, \tilde{\gamma}$ such that the norm equivalence

$$\|\mathbf{v}\|_{\ell_2} \sim \|v\|_{H^s} \tag{6.1.26}$$

holds for $s \in (\tilde{\gamma}, \gamma)$ under the assumption that the wavelets have been normalized in $\|\cdot\|_{H^s}$. Here, for $t < 0$, H^t is the dual of the space H^{-t}.

Now, we define the wavelet representation of (6.1.22) as

$$\mathbf{F}(\mathbf{u}) := \mathbf{A}\mathbf{u} - \mathbf{f} = 0, \tag{6.1.27}$$

where $\mathbf{u}$ is the vector in ℓ_2 that contains the coefficients $\{u_\lambda\}$ of the wavelet expansions of u, $\mathbf{f} \in \ell_2$ contains the terms $\{\langle \psi_\lambda, f \rangle\}_{\lambda \in I\!I}$ and the wavelet representation $\mathbf{A}$ of $\mathcal{A}$ is an infinite matrix with components $\mathbf{A} := \{\langle \psi_\lambda, \mathcal{A}\psi_{\lambda'} \rangle\}_{\lambda, \lambda' \in I\!I}$. All along this chapter we will using the boldface notation $\mathbf{v}$ for vectors in ℓ_2.

Theorem 6.1.3 ([34]). *Provided that the Riesz basis property (6.1.24) holds, the problem (6.1.22) is equivalent to its wavelet representation (6.1.27) in the basis Ψ and there exist finite positive constants $c_{\mathbf{A}}$, $C_{\mathbf{A}}$ such that*

$$c_{\mathbf{A}} \|\mathbf{v}\|_{\ell_2} \leq \|\mathbf{A}\mathbf{v}\|_{\ell_2} \leq C_{\mathbf{A}} \|\mathbf{v}\|_{\ell_2} \, \forall \mathbf{v} \in \ell_2. \tag{6.1.28}$$

Proof:
Let $\mathbf{v} \in \ell_2$ contain the coefficients of the wavelet expansion of $v \in \mathcal{H}$. Then

$$\|\mathbf{v}\|_{\ell_2} \overset{6.1.24}{\leq} c_\Psi^{-1} \|v\|_{\mathcal{H}} \overset{6.1.23}{\leq} c_\Psi^{-1} c_{\mathcal{A}}^{-1} \|Av\|_{\mathcal{H}'} \overset{6.1.25}{\leq} c_\Psi^{-2} c_{\mathcal{A}}^{-1} \|(\langle \psi_\lambda, \mathcal{A}v \rangle)_{\lambda \in I\!I}\|_{\ell_2} = c_\Psi^{-2} c_{\mathcal{A}}^{-1} \|\mathbf{A}\mathbf{v}\|_{\ell_2}. \tag{6.1.29}$$

The converse follows analogously. ∎

In contrast to typical finite element discretizations, the strategy in [34] is to start out with the formulation of a convergent iteration on the infinite-dimensional system in wavelet coordinates (6.1.27). The numerical stage will be envisaged only after the analysis of the *full* infinite-dimensional problem is done. This methodology works along the following lines.

1. Remain in the *infinite dimensional* space of wavelet coordinates where the transformed problem is still well posed and consider the iteration

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \omega(\mathbf{A}\mathbf{u}^n - \mathbf{f}) \tag{6.1.30}$$

which converges to the solution $\mathbf{u}$ with fixed error reduction rate $\rho < 1$ for a suitable ω.

2. Construct a particular coarsening-thresholding policy for each step n that preserves convergence under approximate application of the operators in (6.1.30). That is, one constructs *finitely supported* approximations $[\mathbf{A}\mathbf{u}^n]_\epsilon$ and $[\mathbf{f}]_\epsilon$ that fulfill

$$\|[\mathbf{A}\mathbf{u}^n]_\epsilon - \mathbf{A}\mathbf{u}^n\|_{\ell_2} \leq \epsilon \text{ and } \|[\mathbf{f}]_\epsilon - \mathbf{f}\|_{\ell_2} \leq \epsilon \tag{6.1.31}$$

and performs the finite dimensional iteration

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \omega([\mathbf{A}\mathbf{u}^n]_\epsilon - [\mathbf{f}]_\epsilon) \tag{6.1.32}$$

which converges with reduction rate ρ until the error is of order ϵ. In particular, the construction of $[\mathbf{A}\mathbf{u}^n]_\epsilon$ follows by matrix compression techniques analyzed in [34] and implemented in [6], which neglect small entries of $\mathbf{A}$ in the matrix-vector multiplication.

The construction of these approximations is not related on any a-priori chosen discretization, but dynamically adapted to yield at each step the required accuracy at the expense of possibly few parameters.

The results in [34] state therefore that the construction of optimal wavelet approximations to the solution of operator equations is indeed possible just by updating the solution at each iteration with a thresholding procedure driven by a-posteriori error estimators. Here, optimal is to be understood in the two following senses:

- *Optimal order.* For every required tolerance $\epsilon > 0$ one can find a $n(\epsilon)$ that fulfills

$$\|u - u^n\|_{L_2} \leq \epsilon,\ n \geq n(\epsilon). \tag{6.1.33}$$

Moreover, from the computational point of view, optimality of order $\alpha > 0$ is attained: If the solution belongs to the Besov space $B^\alpha_{\tau,\tau}$ with $\tau := (\alpha+1/2)^{-1}$, then

$$\#\Lambda_{n(\epsilon)} \lesssim \|u\|^{1/\alpha}_{B^\alpha_{\tau,\tau}} \epsilon^{-1/\alpha}, \tag{6.1.34}$$

being $\Lambda_{n(\epsilon)}$ the index configuration of u^n. Moreover, the realization of this rate does *not* require the knowledge of the Besov regularity of the solution.

- *Optimal complexity.* The number of arithmetic operations needed to compute u^n is proportional to $\#\Lambda_{n(\epsilon)}$. The number of sorting operations originally required by the algorithm in [34] scales as $\#\Lambda_{n(\epsilon)} \log(\#\Lambda_{n(\epsilon)})$. However it has been shown (see [5]) that this log term can be in fact removed.

6.1.3 Nonlinear Variational Problems

This strategy has been extended ([35]) to cope with the numerical treatment of nonlinear variational problems, showing the same optimality properties under appropriate assumptions. Let us recover the example (6.1.20) of the linear case and add to it a nonlinear term to get the boundary value problem

$$\mathcal{F}(u) = -\Delta u + \mathcal{G}(u) - f = 0, \text{ in } (0,1),\ u(0) = u(1) = 0. \tag{6.1.35}$$

For a $\mathcal{G}$ mapping H_0^1 onto H^{-1}, we can write its variational formulation in H_0^1 as

$$\int_0^1 \nabla u \cdot \nabla v dx + \int_0^1 \mathcal{G}(u) v dx = \int_0^1 f v dx \tag{6.1.36}$$

for all $v \in H_0^1$. Under adequate assumptions on $\mathcal{G}$, this problem has an unique solution u and the Riesz basis property allows to transform this problem into an equivalent one in wavelet coordinates which is well posed, see [35] for details. We look now for a $\mathbf{u} \in \ell_2$ that meets

$$\mathbf{F}(\mathbf{u}) = \mathbf{A} + \mathbf{G}(\mathbf{u}) - \mathbf{f} = 0, \tag{6.1.37}$$

where the infinitely dimensional vector $\mathbf{G}(\mathbf{u}) \in \ell_2$ represents the nonlinearity in the wavelet coordinates and contains the elements $\{\langle \psi_\lambda, \mathcal{G}(u)\rangle\}_{\lambda \in I\!I}$. Like in the linear case, in view of the well posedness of (6.1.37), the iteration in *infinite dimension*

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \mathbf{B}_n \mathbf{F}(\mathbf{u}^n) \tag{6.1.38}$$

converges to the solution $\mathbf{u}$ for adequate choices of the infinite matrix $\mathbf{B}_n$. For instance, the choice $\mathbf{B}_n = \omega \mathbf{I}$ is the Richardson iteration, and to get the Newton's method, which can be shown to converge quadratically for a good starting value $\mathbf{u}^0$, we should plug $\mathbf{B}_n = D\mathbf{F}(\mathbf{u}^n)^{-1}$. Here, the action of the Jacobian $D\mathbf{F}(\mathbf{u})$ on a $\mathbf{v} \in \ell_2$ is defined as $\{\langle \psi_\lambda, D\mathcal{F}(u)(w)\rangle\}_{\lambda \in \Lambda}$, where $D\mathcal{F}(u)$ is the Frechét derivative of $\mathcal{F}$ at u.

Like in the linear case, the iteration (6.1.38) will be evaluated in an approximative way.

Evaluation of nonlinear terms As the authors note, the implementation of this optimal algorithm requires an efficient method to compute or approximate the quantities $\{\langle \psi_\lambda, \mathcal{G}(u^n)\rangle\}_{\lambda \in \tilde{\Lambda}^n}$ for some configuration $\tilde{\Lambda}^n$ and for $u^n = \sum_{\lambda \in \Lambda^n} u_\lambda^n \psi_\lambda$. This task is split into two steps:

1. *Construction of a wavelet configuration.* Find a finite set of wavelet indices ensuring that the restriction of $\mathbf{G}(\mathbf{u}^n)$ to this configuration approximates $\mathbf{G}(\mathbf{u}^n)$ within a given tolerance. In fact, a constructive way to select an index configuration $\mathcal{T}$ that fulfills

$$\|\mathbf{G}(\mathbf{u}^n) - \mathbf{G}(\mathbf{u}^n)|_{\mathcal{T}}\|_{\ell_2} \leq \epsilon \tag{6.1.39}$$

for whichever $\epsilon > 0$ needed to ensure convergence, is provided in [36] for the class of nonlinearities that satisfies growth conditions at infinity of the type

$$|\mathcal{G}^{(n)}(x)| \lesssim (1 + |x|)^{[p-n]_+},\ x \in I\!R,\ n = 0, 1, \ldots, \tilde{n} \tag{6.1.40}$$

for some $p \geq 0$ and $\tilde{n}$. This includes $\mathcal{G}(u) = u^p$ for all $\tilde{n}$ if p is an integer and with $\tilde{n}$ the integer part of p.

The prediction of a structure is based in estimations that relates *locally* the size of the wavelet coefficients of u^n to the decay rate of the wavelet coefficients of $\mathcal{G}(u)$. Denoting by S_λ the support of a wavelet ψ_λ, these estimates are of the type

$$\mathbf{G}(\mathbf{u}^n)|_\lambda \lesssim \sup_{S_\lambda \cap S_{\lambda'} \neq \emptyset} |u^n_{\lambda'}| 2^{-\gamma(|\lambda| - |\lambda'|)} \tag{6.1.41}$$

where γ depends on the nonlinearity and the wavelet family. This induces the construction of the predicted configuration $\mathcal{T}$ by selecting the relevant coefficients in Λ^n and gathering their descendants in a tree structure whose depth is governed by γ.

2. *Efficient computation of coefficients.* Once the configuration of interest $\mathcal{T}$ is known, [51] and more recently [11] explain how to use local transformation techniques to get an interpolation to the function $\mathcal{G}(u^n)$ in terms of scaling functions. Suitable quadrature techniques are then used to approximate the numerical value of the wavelet coordinates. As the exact coefficients of $\mathbf{G}(\mathbf{u}^n)|_\mathcal{T}$ are not computed, but only an approximation $\mathbf{w}$, this introduces an *approximation error* $\|\mathbf{w} - \mathbf{G}(\mathbf{u}^n)|_\mathcal{T}\|_{\ell_2}$. This error is shown to be controlled by the *truncation error* made in (6.1.39), that is $\|\mathbf{w} - \mathbf{G}(\mathbf{u}^n)|_\mathcal{T}\|_{\ell_2} \lesssim \epsilon$ which is enough to ensure the overall convergence of the algorithm by the triangle inequality $\|\mathbf{w} - \mathbf{G}(\mathbf{u}_n)\|_{\ell_2} \leq (1 + C)\epsilon$.

The methodology developed in [36] and [51] states the feasibility of optimal adaptive wavelet schemes for a wide range of nonlinear problems. Nevertheless, its numerical implementation based on local transformations is rather involved. Moreover the predicted configuration $\mathcal{T}$ is based on a worst case estimation and is usually larger than actually required to represent $\mathcal{G}(u)$ up to required tolerance. Numerically, one has to assemble and allocate an oversized structure, with the corresponding damage to performance and storage issues.

6.1.4 LS Approach

We discuss in the following an alternative way to evaluate nonlinear terms of the form $\{\langle \psi_\lambda, \mathcal{G}(u)\rangle\}_{\lambda \in \Lambda}$, where u is a known function that can be expressed as a finite linear combination of wavelets and Λ is a given set of wavelet indices. For instance, Λ can be the set $\mathcal{T}$ in (6.1.39) if we envisage the use of the converging algorithm given in [35].

The knowledge of the wavelet representation of $\mathcal{G}(u)$

$$\mathcal{G}(u) = \sum_{\lambda \in I\!I} g_\lambda \psi_\lambda, \tag{6.1.42}$$

or in vector form $\mathbf{g} := \{g_\lambda\}_{\lambda \in I\!I}$, would allow to evaluate $\mathbf{G}(\mathbf{u})$ *exactly* by noting that

$$\begin{aligned} \int_0^1 \mathcal{G}\left(u(x)\right) \psi_\lambda(x)\, dx &= \int_0^1 \sum_{\lambda' \in I\!I} g_{\lambda'} \psi_{\lambda'}(x) \psi_\lambda(x)\, dx \\ &= \sum_{\lambda' \in I\!I} g_{\lambda'} \int_0^1 \psi_{\lambda'}(x) \psi_\lambda(x)\, dx \qquad (6.1.43) \\ &= \sum_{\lambda' \in I\!I} g_{\lambda'}\ M_{\lambda,\lambda'}, \qquad (6.1.44) \end{aligned}$$

where $M_{\lambda,\lambda'}$ are the entries of the mass matrix $\mathbf{M}$ of the problem. We would then simply have:

$$\mathbf{G}(\mathbf{u}) = \mathbf{M}\mathbf{g}. \tag{6.1.45}$$

In principle, we need infinitely many wavelets to represent $\mathcal{G}(u)$ exactly. However, we are interested only in those that are not orthogonal in L_2 to the wavelets indexed by the given set Λ, as only they can contribute in 6.1.43. We can henceforth work with a maximal set of indexes $\bar{\Lambda} := \Lambda \cup \partial\Lambda$. Here, $\partial\Lambda$ stands for the set of indices of the wavelets that are not orthogonal to the ones in Λ. For preorthogonal wavelets in L_2, $\partial\Lambda$ includes just the neighbors on the same resolution level of the wavelets in Λ,

$$\partial\Lambda = \{\lambda \in I\!I : \exists \lambda' \in \Lambda, |\lambda| = |\lambda'|,\ \text{supp}\ \psi_\lambda \cap\ \text{supp}\ \psi_{\lambda'} \neq \emptyset\} \tag{6.1.46}$$

and henceforth, if Λ is finite, so is $\bar{\Lambda}$ as well.

At this stage one can think of constructing an approximation to $\mathbf{g}|_{\bar{\Lambda}}$ by means of a Least Squares technique, just by selecting some appropriate set of points $\hat{X}$, evaluating $\mathcal{G}(u(x_i))$ at each $x_i \in \hat{X}$, and using Algorithm 2.3.2 of Chapter 2 to get the configuration $\hat{\Lambda} \subseteq \bar{\Lambda}$ and the coefficients $\{\hat{g}_\lambda\}_{\lambda \in \hat{\Lambda}}$ of the approximation $\hat{g}(x) := \sum_{\lambda \in \hat{\Lambda}} \hat{g}_\lambda \psi_\lambda(x)$. Here arises the question of how to choose the set of points. First, note that the stability results given in Section 2.4 provide us with a constructive way to select an appropriate set of points $\hat{X}$. Let us introduce the following definitions.

Definition 6.1.4 (Point representation of a wavelet). *We define the point representation X_λ^m of order $m, m \in I\!N$, of a wavelet ψ_λ, with $\lambda = (j,k)$ as the set*

$$X_\lambda^m := \left\{x :\ x \in \{2^{-(j+m)}k\}_{k=0,1,\ldots},\ x \in\ \textit{supp}\ \psi_\lambda\right\}. \tag{6.1.47}$$

Definition 6.1.5 (Point representation of a set of wavelets). *We define the point representation $X_\Lambda^{(m)}$ of order m of a set Λ of wavelet indices as the union of the point representations of all wavelets indexed by Λ excluding repetitions:*

$$X_\Lambda^m := \{x : \exists \lambda \in \Lambda\ \textit{such that}\ x \in X_\lambda^m\} \tag{6.1.48}$$

One can easily check that point representations of order $m > 0$ generate invertible systems.

Proposition 6.1.6. *If $m > 0$, the matrix $A^T_{\Lambda,X^m_\Lambda} A_{\Lambda,X^m_\Lambda}$, where A_{Λ,X^m_Λ} is the observation matrix of Λ with respect to X^m_Λ as defined in (2.2.2), is nonsingular.*

Proof:
For $m > 1$, the point distribution X^m_λ with $\lambda = (j,k)$ includes the three points $\{2^{-j}k, 2^{-j}(k{+}1/2), 2^{-j}(k{+}1)\}$. Recalling the Definition 2.4.1 of the internodal spaces $I^{(i)}_\lambda, i = 1,\ldots,6$ of a wavelet, we can check that

$$2^{-j}k \in I^{(3)}_\lambda,\ 2^{-j}(k{+}1/2) \in I^{(3,4)}_\lambda \text{ and } 2^{-j}(k{+}1) \in I^{(4)}_\lambda. \tag{6.1.49}$$

Thus, ψ_λ is balanced with respect to X^m_Λ (recall Definition 2.4.2). This holds for all $\lambda \in \Lambda$ and the invertibility of $A^T_{\Lambda,X^m_\Lambda} A_{\Lambda,X^m_\Lambda}$ follows using the same argumentation leading to Theorem 2.4.13. ∎

Thus, one could simply consider taking the point representation of the maximal configuration $X^m_{\bar\Lambda}$ for some $m > 0$ as the set of sampling points to be used as input for the Algorithm 2.3.2 to construct $\hat{\mathbf{g}}|_{\hat\Lambda}$. Nevertheless, such a choice has a clear flaw in relation to the complexity, as it does not fully benefit from the adaptivity of the Least Squares method. The reason is that the configuration $\hat\Lambda$ found by the LS algorithm may indeed contain much less degrees of freedom than the predicted maximal configuration $\bar\Lambda$, but the number of sampling points, and in consequence the overall complexity of the algorithm, still scales linearly with $\#\bar\Lambda$. On the other side, a selection of $\hat X$ should not risk to miss sampling points on local important features of $\mathcal{G}(u)$, for instance jumps or high gradients supported on small subintervals of the domain, whose location is not known a priori. Such an undersampling would result in a wrong estimation of the tree structure.

The need to balance these two factors motivates an adaptive way to select the configuration *and* also the sampling points during the growth of the tree in the LS algorithm.

We modify correspondingly Algorithm 2.3.2 in order to take up these considerations: First, as we can determinate ourselves the observation points, we will obviously eliminate the horizontal thresholding. Further, we adapt the set of observations to the current configuration, so that the number of used samples scales with the number of wavelets actually used. Finally, as we are given a maximal configuration $\bar\Lambda$ in which relevant coefficients are to be found, we will not consider wavelets not belonging to this index set.

With these modifications, and recalling the notation $\delta(\Lambda)$ for the children of the last-level-wavelets of a set Λ as in Section 2.3.1, the resulting algorithm reads

Algorithm 6.1.7. LS - Evaluation of nonlinearities

1. Fix maximal configuration $\bar{\Lambda}$, starting level j_0, order m of the point representations and thresholding parameter $\epsilon > 0$.
2. Create $\hat{\Lambda}_{j_0}$.
3. For each $j = j_0, \ldots$.
   (a) Create $\delta(\hat{\Lambda}_j)$.
   (b) Construct $\tilde{\Lambda}_{j+1} = \hat{\Lambda}_j \cup (\delta(\hat{\Lambda}_j) \cap \bar{\Lambda})$.
   (c) Compute $\{\hat{g}_\lambda^{j+1}\}_{\lambda \in \tilde{\Lambda}_{j+1}}$ that solves

$$\sum_{x_i \in X^m_{\tilde{\Lambda}_{j+1}}} \left(\mathcal{G}(u(x_i)) - \sum_{\lambda \in \tilde{\Lambda}_{j+1}} \hat{g}_\lambda^{j+1} \, \psi_\lambda(x_i) \right)^2 \to \min! \tag{6.1.50}$$

   (d) Select $\Lambda^\epsilon_{j+1} = \{\lambda \in \delta(\hat{\Lambda}_j) \cap \bar{\Lambda} : |\hat{g}_\lambda^{j+1}| \geq \epsilon\}$.
   (e) If $\Lambda^\epsilon_{j+1} = \emptyset$ stop, else let $\hat{\Lambda}_{j+1} = \hat{\Lambda}_j \cup \Lambda^\epsilon_{j+1}$.

Note that this approach differs from the one in [36] and [51] explained in Section 6.1.3. In the LS approach the construction of the wavelet configuration $\hat{\Lambda}$ and the computation of the corresponding coefficients $\{\hat{g}_\lambda\}$ occur simultaneously, as it works creating a hierarchy of configurations $\hat{\Lambda}_{j_0} \subset \hat{\Lambda}_{j_0+1} \subset \ldots \subseteq \hat{\Lambda}$ by using the coefficients on each level j to predict the significant ones on the next level $j+1$. The use of the information about $\mathcal{G}(u)$ gained at each level might allow for a possible ending of the algorithm at some early level before having computed the full configuration $\bar{\Lambda}$, so that $\#\hat{\Lambda} < \#\bar{\Lambda}$. Note that the size of the trial configuration $\bar{\Lambda}$ can have been overestimated by the algorithms described in [51], as $\bar{\Lambda}$ is ensured to be large enough to fulfill the approximation requirement, but not necessarily to contain the smallest possible number of parameters.

Remark 6.1.8. *A correction of the algorithmic drawbacks of [51] and [36] has been envisaged recently in [7]. Although the strategy operates in a sweep from fine to coarse scales and is therefore opposite to our LS approach, it also relays on reusing the information gained in the analysis of each level.*

Reproduction Properties

The exact wavelet representation of the function $\mathcal{G}(u)$ in (6.1.42) contains infinitely many terms. In order to discuss the effect of cutting the infinite expansion and keeping a finite configuration Λ we split $\mathcal{G}(u(x))$ into two terms

$$g(x) := \mathcal{G}(u(x)) = g^\Lambda(x) + g^+(x) \tag{6.1.51}$$

with

$$g^{\Lambda}(x) := \sum_{\lambda\in\Lambda} g_\lambda \psi_\lambda(x) \text{ and } g^{+}(x) := \sum_{\lambda\in\Lambda^+} g_\lambda \psi_\lambda(x) \tag{6.1.52}$$

where $\Lambda^+ := I\!I/\Lambda$. The error made by using $g^{\Lambda}(x)$ as approximation to $\mathcal{G}(u(x))$ neglecting $g^+(x)$ is the *truncation error*, denoted by

$$\epsilon_\Lambda(\mathcal{G}(u)) := \left(\sum_{\lambda\in\Lambda^+} g_\lambda^2\right)^{\frac{1}{2}} = \|\mathbf{g}|_{\Lambda^+}\|_{\ell_2(\Lambda^+)} \tag{6.1.53}$$

or simply by ϵ_Λ when the referred function is clear. Now, an approximation to the true coefficients of the configuration Λ is computed by an LS method using some given set X of sampling points, that is, we approximate g^{Λ} by the function $\hat{g}$, whose wavelet expansion are contained in the vector $\hat{\mathbf{g}} \in \ell_2$. Its components $\{\hat{g}_\lambda\}_{\lambda\in\Lambda}$ are nonzero only on a finite Λ and solve

$$\sum_{i=1}^{\#X} \left(\mathcal{G}(u(x_i)) - \sum_{\lambda\in\Lambda} \hat{g}_\lambda \psi_\lambda(x_i)\right)^2 \to \min!. \tag{6.1.54}$$

This introduces an *approximation error*, which in this context is defined as

$$\hat{\epsilon}_{\Lambda,X}(\mathcal{G}(u)) := \left(\sum_{\lambda\in\Lambda} (\hat{g}_\lambda - g_\lambda)^2\right)^{\frac{1}{2}} = \|\hat{\mathbf{g}}|_\Lambda - \mathbf{g}|_\Lambda\|_{\ell_2(\Lambda)}. \tag{6.1.55}$$

We will denote it by $\hat{\epsilon}_{\Lambda,X}$ when the reference to the function is not necessary. We show now that the approximation error can be controlled by the truncation error. The vector $\hat{g}$ fulfills

$$A_{\Lambda,X}^T A_{\Lambda,X} \hat{g} = A_{\Lambda,X}^T z, \tag{6.1.56}$$

where the observation matrix is defined componentwise as usual as

$$(A_{\Lambda,X})_{i,\lambda} = \psi_\lambda(x_i), \text{ for } x_i \in X, \lambda \in \Lambda \tag{6.1.57}$$

and the vector $z \in I\!R^{\#X}$ has components

$$z_i = \mathcal{G}(u(x_i)) = \sum_{\lambda\in I\!I} g_\lambda \psi_\lambda(x_i). \tag{6.1.58}$$

Separating in (6.1.58) the components from Λ from those from Λ^+ one gets the representation

$$z = A_{\Lambda,X}\mathbf{g}|_\Lambda + A_{\Lambda^+,X}\mathbf{g}|_{\Lambda^+}. \tag{6.1.59}$$

If $A_{\Lambda,X}^T A_{\Lambda,X}$ is invertible, inserting (6.1.59) into (6.1.56) and taking norms in ℓ_2 we get the expression

$$\|\hat{\mathbf{g}}|_\Lambda - \mathbf{g}|_\Lambda\|_{\ell_2(\Lambda)} = \left\|(A_{\Lambda,X}^T A_{\Lambda,X})^{-1} A_{\Lambda,X}^T A_{\Lambda^+,X}\mathbf{g}|_{\Lambda^+}\right\|_{\ell_2(\Lambda)}, \tag{6.1.60}$$

which tells us that under suitable selection of the set of samples X, the LS error can be indeed controlled by the truncation error: In the trivial case where the truncation error is zero, that is, if the approximated function is actually a finite linear combination of the wavelets indexes by Λ, the LS method recovers it exactly, independently of how the sampling points X are chosen, as long as their distribution makes $(A_{\Lambda,X}^T A_{\Lambda,X})^{-1}$ invertible. Otherwise, the effect of the truncation error will be modulated through the term $(A_{\Lambda,X}^T A_{\Lambda,X})^{-1} A_{\Lambda,X}^T A_{\Lambda^+,X}$, which is due to the sampling geometry. We will analyze it first for uniform distributions of sampling points, which we will denote by

$$X_*^m := \{2^{-m} i\}_{i=0,\ldots,2^m}. \tag{6.1.61}$$

We will say that m is the order of the grid X_*^m.

Theorem 6.1.9. *Let $\mathcal{G}(u(x))$ be a function with a possibly infinite wavelet expansion contained in the vector $\mathbf{g} \in \ell_2(I\!I)$ and $\Lambda \in I\!I$ a finite configuration of wavelets that realizes a given maximal truncation error $\epsilon > 0$, that is, for $\Lambda^+ = I\!I/\Lambda$, we have*

$$\|\mathbf{g}|_{\Lambda^+}\|_{\ell_2(I\!I)} = \epsilon_\Lambda(\mathcal{G}(u)) \leq \epsilon. \tag{6.1.62}$$

Then, for m large enough, the approximation error $\hat{\epsilon}_{\Lambda,X_*^m}$ is bounded by a multiple of the maximal truncation error ϵ. One has

$$\lim_{m\to\infty} \hat{\epsilon}_{\Lambda,X_*^m} \leq C\epsilon, \tag{6.1.63}$$

where the constant $C > 0$ depends only on the used wavelet family and not on the particular structure Λ used to meet (6.1.62). An upper bound for C is given by

$$C \leq \sqrt{2}\check{\sigma}_{\mathbf{M}}^{-1} C_\Psi^2, \tag{6.1.64}$$

where $\check{\sigma}_{\mathbf{M}}$ is the smallest eigenvalue of the mass matrix $\mathbf{M}$ of the used wavelet family Ψ, and C_Ψ is the smallest constant that ensures a norm equivalence of the type (6.1.24) for $\mathcal{H} = L_2$.

Proof:

As seen in the previous discussion, the coefficients $\hat{\mathbf{g}} \in \ell_2(I\!I)$, ($\hat{g}_\lambda = 0$ if $\lambda \notin \Lambda$), of the LS approximation to $\mathcal{G}(u(x))$ computed using the wavelets from the index set Λ and the set of points X_*^m for any m fulfills

$$A_{\Lambda,X_*^m}^T A_{\Lambda,X_*^m} (\hat{\mathbf{g}}|_\Lambda - \mathbf{g}|_\Lambda) = A_{\Lambda,X_*^m}^T A_{\Lambda^+,X_*^m} \mathbf{g}|_{\Lambda^+}. \tag{6.1.65}$$

The term $A_{\Lambda^+,X_*^m}\mathbf{g}|_{\Lambda^+} \in I\!R^{\#X_*^m}$ contains the evaluations of $g^+(x) = \sum_{\lambda\in\Lambda^+} d_\lambda \psi_\lambda(x)$ at all the points in X_*^m, so that the right hand side of (6.1.65) is the following vector in $\ell_2(\Lambda)$:

$$A_{\Lambda,X_*^m}^T A_{\Lambda^+,X_*^m} \mathbf{g}|_{\Lambda^+} = \begin{pmatrix} \sum_{i=0}^{2^m} g^+(x_i)\psi_{\lambda_1}(x_1) \\ \vdots \\ \sum_{i=0}^{2^m} g^+(x_i)\psi_{\lambda_1}(x_1) \end{pmatrix}. \tag{6.1.66}$$

Each component can be writen as

$$\sum_{i=0}^{2^m} g^+(x_i)\psi_\lambda(x_i) = \sum_{i=0}^{2^m} g^+(x_i)\psi_\lambda(x_i)\frac{1}{\Delta_i}\Delta_i = 2^m \sum_{i=0}^{2^m} g^+(x_i)\psi_\lambda(x_i)\Delta_i \tag{6.1.67}$$

being $\Delta_i := 2^{-m}$. As the points are equispaced, Δ_i is the length between two measurements and in the limit

$$\lim_{m\to\infty} \sum_{i=0}^{2^m} g^+(x_i)\psi_\lambda(x_i)\Delta_i = \int_0^1 g^+(x)\psi_\lambda(x)dx. \tag{6.1.68}$$

Hence, for any $\lambda \in \Lambda$ and any tolerance $\eta_1 > 0$, one can always find a finite uniform grid of order $\tilde{m}_\lambda(\eta_1)$ such that for all $\tilde{m} \geq \tilde{m}_\lambda(\eta_1)$

$$\begin{aligned} \left| \sum_{i=0}^{2^{\tilde{m}}} g^+(x_i)\psi_\lambda(x_i) - 2^{\tilde{m}} \int_0^1 g^+(x)\psi_\lambda(x)dx \right| &\leq \\ \left|2^{\tilde{m}}\right| \left| \sum_{i=0}^{2^{\tilde{m}}} g^+(x_i)\psi_\lambda(x_i)\Delta_i - \int_0^1 g^+(x)\psi_\lambda(x)dx \right| &\leq 2^{\tilde{m}}\eta_1 \end{aligned} \tag{6.1.69}$$

holds with $\{x_i\}_{i=0,\ldots,2^{\tilde{m}}} = X_*^{\tilde{m}}$. Selecting

$$m_1(\eta_1) := \max_{\lambda\in\Lambda} \tilde{m}_\lambda(\eta_1), \tag{6.1.70}$$

$X_*^{m_1(\eta_1)}$ is the coarsest grid that can ensure (6.1.69) for the whole set Λ and for a given η_1. For the points of any uniform grid of order $m \geq m_1(\eta_1)$ we can write

$$\left(\sum_{i=0}^{2^m} g^+(x_i)\psi_\lambda(x_i) \right)^2 \leq 2 \cdot 2^{2m} \left(\int_0^1 g^+(x)\psi_\lambda(x)dx \right)^2 + 2 \cdot 2^{2m}\eta_1^2 \tag{6.1.71}$$

and summing on $\lambda \in \Lambda$ for all the squared components of the vector in (6.1.66) we get

$$\|A^T_{\Lambda,X_*^m} A_{\Lambda^+,X_*^m} \mathbf{g}|_{\Lambda^+}\|^2_{\ell_2(\Lambda)} \leq 2\cdot 2^{2m} \sum_{\lambda\in\Lambda} \left(\int_0^1 g^+(x)\psi_\lambda(x)dx \right)^2 + 2\cdot 2^{2m} \#\Lambda \eta_1^2 \tag{6.1.72}$$

for $m \geq m_1(\eta_1)$. The norm equivalences (6.1.24) and (6.1.25) applied on $\mathcal{H} = L_2$ allow to bound the sum of integrals in (6.1.72) with a multiple of the maximal

truncation error,

$$\begin{aligned}\left\|\left\{\int_0^1 g^+(x)\psi_\lambda(x)dx\right\}_{\lambda\in\Lambda}\right\|_{\ell_2(\Lambda)} &\leq \left\|\left\{\int_0^1 g^+(x)\psi_\lambda(x)dx\right\}_{\lambda\in I\!I}\right\|_{\ell_2(I\!I)} \\ &\overset{(6.1.25)}{\leq} C_\Psi \left\|g^+\right\|_{L_2} \\ &\overset{(6.1.24)}{\leq} C_\Psi^2 \left\|\left\{g_\lambda^+\right\}_{\lambda\in I\!I}\right\|_{\ell_2(I\!I)} \\ &= C_\Psi^2\epsilon_\Lambda \leq C_\Psi^2\epsilon,\end{aligned}$$

and one finally gets the bound

$$\|A^T_{\Lambda,X^m_*}A_{\Lambda^+,X^m_*}\mathbf{g}|_{\Lambda^+}\|^2_{\ell_2(\Lambda)} \leq 2\cdot 2^{2m}(C_\Psi)^4\epsilon^2 + 2\cdot 2^{2m}\#\Lambda\eta_1^2 \tag{6.1.73}$$

provided $m \geq m_1(\eta_1)$. This gives us an upper estimate on the right hand side of (6.1.65) in terms of the maximal truncation error where the tree dependence no longer appears explicitely. The tree is however still present, as the order of refinement $m_1(\eta_1)$ required to guarantee (6.1.73) for a given η_1 depends on the particular tree structure Λ.

We proceed now in the other direction and get in a similar way a lower bound for the left hand side of (6.1.73) in terms of known quantities. The structure of the elements of the coefficient matrix $A^T_{\Lambda,X^m_*}A_{\Lambda,X^m_*}$, suggests comparing it with the mass matrix $\mathbf{M}$. Denoting by M_Λ the restriction $[\mathbf{M}]_\Lambda$ of the infinite matrix $\mathbf{M}$ to the finite configuration Λ, one can write for all $v \in \ell_2(\Lambda)$ and for all $m \in I\!R$

$$\|2^m M_\Lambda v\|_{\ell_2(\Lambda)} \leq \|A^T_{\Lambda,X^m_*}A_{\Lambda,X^m_*}v\|_{\ell_2(\Lambda)} + \|2^m M_\Lambda v - A^T_{\Lambda,X^m_*}A_{\Lambda,X^m_*}v\|_{\ell_2(\Lambda)}. \tag{6.1.74}$$

We will see now that $\|2^m M_\Lambda v - A^T_{\Lambda,X^m_*}A_{\Lambda,X^m_*}v\|_{\ell_2(\Lambda)}$ can be controlled by suitable choices of m. To this end, we study the relation between the individual entries of $A^T_{\Lambda,X^m_*}A_{\Lambda,X^m_*}$ and M_Λ. For every couple of wavelets indexed by $\lambda,\lambda' \in \Lambda$ and for every tolerance $\eta_2 > 0$, one can find an order of refinement $\tilde{m}_{\lambda,\lambda'}(\eta_2) < \infty$ meeting

$$\begin{aligned}\left|\sum_{i=0}^{2^{\tilde{m}}}\psi_\lambda(x_i)\psi_{\lambda'}(x_i) - 2^{\tilde{m}}\int_0^1 \psi_\lambda(x)\psi_{\lambda'}(x)dx\right| &\leq \\ 2^{\tilde{m}}\left|\sum_{i=0}^{2^{\tilde{m}}}\psi_\lambda(x_i)\psi_{\lambda'}(x_i)\Delta_i - \int_0^1 \psi_\lambda(x)\psi_{\lambda'}(x)dx\right| &\leq 2^{\tilde{m}}\eta_2,\end{aligned} \tag{6.1.75}$$

for the points of all X^m_* with $m \geq \tilde{m}_{\lambda,\lambda'}(\eta_2)$. Selecting

$$m_2 = m_2(\eta_2) := \max_{\lambda,\lambda'\in\Lambda}\tilde{m}_{\lambda,\lambda'}(\eta_2), \tag{6.1.76}$$

follows the uniform bound for all the components of $2^m M_\Lambda - A^T_{\Lambda,X^m_*} A_{\Lambda,X^m_*}$

$$\max_{\lambda,\lambda'} \left| \sum_{i=0}^{2^m} \psi_\lambda(x_i)\psi_{\lambda'}(x_i) - 2^m \int_0^1 \psi_\lambda(x)\psi_{\lambda'}(x)dx \right|, \leq 2^m \eta_2,\ m \geq m_2(\eta_2). \quad (6.1.77)$$

This uniform estimation for the components can be translated into an estimation for the matrix norm $\|\cdot\|_{\ell_2(\mathbb{R}^n)\to\ell_2(\mathbb{R}^n)}$, which we need to bound the second term on the right hand side of (6.1.74). Indeed, for any matrix $B \in \mathbb{R}^{n\times n}$, $n \in \mathbb{N}$, with components $(b_{i,j})$, one can check that $\max_{i,j} |b_{i,j}|$ is a matrix norm with the property (see [81])

$$\max_{i,j} |b_{i,j}| \leq \|B\|_{\ell_2(\mathbb{R}^n)\to\ell_2(\mathbb{R}^n)} \leq n \max_{i,j} |b_{i,j}|, \quad (6.1.78)$$

and consequently for $B = 2^m M_\Lambda v - A^T_{\Lambda,X^m_*} A_{\Lambda,X^m_*} v$

$$\begin{aligned} \|2^m M_\Lambda v - A^T_{\Lambda,X^m_*} A_{\Lambda,X^m_*} v\|_{\ell_2(\Lambda)} &\leq \|2^m M_\Lambda - A^T_{\Lambda,X^m_*} A_{\Lambda,X^m_*}\|_{\ell_2(\Lambda)\to\ell_2(\Lambda)} \|v\|_{\ell_2(\Lambda)} \\ &\leq \#\Lambda \cdot 2^m \eta_2 \|v\|_{\ell_2(\Lambda)} \end{aligned} \quad (6.1.79)$$

holds for all $v \in \ell_2(\Lambda)$ and $m \geq m_2(\eta_2)$. Now, by the Rayleigh Ritz Theorem $\|M_\Lambda v\|_{\ell_2(\Lambda)}$ can be bounded from below using the smallest eigenvalue of M_Λ, denoted by $\check{\sigma}_{M_\Lambda}$,

$$\check{\sigma}_{M_\Lambda} \|v\|_{\ell_2(\Lambda)} \leq \|M_\Lambda v\|_{\ell_2(\Lambda)}. \quad (6.1.80)$$

This lower bound still depends on Λ, as M_Λ is the restriction of $\mathbf{M}$ to the particular configuration Λ. However,

as M_Λ is a principal submatrix of $\mathbf{M}$, the interlacing theorem of Cauchy (see for instance [81]) shows that the smallest eigenvalue of $\mathbf{M}$, denoted by $\check{\sigma}_{\mathbf{M}}$, fulfills

$$\check{\sigma}_{\mathbf{M}} \leq \check{\sigma}_{M_\Lambda}, \quad (6.1.81)$$

giving a bound independent of Λ. With (6.1.80), this yields

$$\|v\|_{\ell_2(\Lambda)} \leq \check{\sigma}^{-1}_{\mathbf{M}} \|M_\Lambda v\|_{\ell_2(\Lambda)} \leq \check{\sigma}^{-1}_{\mathbf{M}} 2^{-m} \|2^m M_\Lambda v\|_{\ell_2(\Lambda)}, \quad (6.1.82)$$

for all $v \in \ell_2(\Lambda)$ and for all $\tilde{m} \in \mathbb{R}$. For $m \geq m_2(\eta_2)$ we can put (6.1.74), (6.1.79) and (6.1.82) together and obtain

$$\|v\|_{\ell_2(\Lambda)} \leq \check{\sigma}^{-1}_{\mathbf{M}} 2^{-m} \left(\|A^T_{\Lambda,X^m_*} A_{\Lambda,X^m_*} v\|_{\ell_2(\Lambda)} + \#\Lambda \cdot 2^m \eta_2 \|v\|_{\ell_2(\Lambda)} \right). \quad (6.1.83)$$

If we choose η_2 small enough to meet

$$1 - \check{\sigma}^{-1}_{\mathbf{M}} \#\Lambda \eta_2 > 0 \quad (6.1.84)$$

this can be written as

$$\|v\|_{\ell_2(\Lambda)} \leq \left(1 - \check{\sigma}^{-1}_{\mathbf{M}} \#\Lambda \cdot \eta_2\right)^{-1} \check{\sigma}^{-1}_{\mathbf{M}} 2^{-m_2} \|A^T_{\Lambda,X^{m_2}_*} A_{\Lambda,X^{m_2}_*} v\|_{\ell_2(\Lambda)} \quad (6.1.85)$$

provided $m \geq m_2(\eta_2)$. We can now glue together the upper and lower bounds for (6.1.65). For a required tolerance $\eta := \min\{\eta_1, \eta_2\}$ we select an $\tilde{m}(\eta) = \max\{m_1(\eta), m_2(\eta)\}$. For this choice, (6.1.65) and (6.1.73) yield

$$\|A^T_{\Lambda,X^m_*} A_{\Lambda,X^m_*}(\hat{\mathbf{g}}|_\Lambda - \mathbf{g}|_\Lambda)\|^2_{\ell_2(\Lambda)} \leq 2 \cdot 2^{2m}(C_\Psi)^4\epsilon^2 + 2 \cdot 2^{2m}\#\eta^2 \tag{6.1.86}$$

for all $m \geq \tilde{m}(\eta)$. Now, we can select $v = \hat{\mathbf{g}}|_\Lambda - \mathbf{g}|_\Lambda$, where the LS approximation $\hat{\mathbf{g}}|_\Lambda$ is computed using X^m_* as sampling points. For this choice $\|v\|_{\ell_2(\Lambda)} = \hat{\epsilon}_{\Lambda,X^m_*}$ and (6.1.86) together with (6.1.85) yields

$$\hat{\epsilon}^2_{\Lambda,X^m_*} \leq \left(1 - \check{\sigma}^{-1}_{\mathbf{M}}\#\Lambda\eta\right)^{-2} \check{\sigma}^{-2}_{\mathbf{M}} 2\left(C^4_\Psi\epsilon^2 + \#\Lambda\eta^2\right), m \geq \tilde{m}(\eta). \tag{6.1.87}$$

As we can select an arbitrarily small η and still find a order of refinement m large enough to meet (6.1.87), in the limit $\eta \to 0$, we get the simple bound

$$\lim_{m\to\infty} \hat{\epsilon}_{\Lambda,X^m_*} \leq \sqrt{2}\check{\sigma}^{-1}_{\mathbf{M}} C^2_\Psi \epsilon. \tag{6.1.88}$$

■

Note that according to the discussion at the end of Section 6.1.3, this theorem states that the LS data fitting procedure can be reliably inserted as technique for evaluation of nonlinearities in the methodology described in [35] as long as the distribution of sampling points is dense enough. This requirement is also present in quadrature based methods, see [7].

Theorem 6.1.9 applies for a lacunary tree Λ sampled on a uniform grid of points. In the adaption of the proof for adaptive samplings of the type X^m_Λ one finds the difficulty that the terms of the type $\sum_{i=0}^{\#X^m_\Lambda} g^+(x_i)\psi_\lambda(x_i)$ (respectively $\sum_{i=0}^{\#X^m_\Lambda} \psi_\lambda(x_i)\psi_{\lambda'}(x_i)$) are no longer comparable in the limit to integrals of the type $\int_0^1 g^+(x)\psi_\lambda(x)dx$ (respectively $\int_0^1 \psi_\lambda(x)\psi_{\lambda'}(x)dx$), as the point distribution is no longer equispaced. To compensate for the irregular distribution, one must introduce in the computations some *density* function to measure the inverse of the distance between to neighboring samples of the point distribution X^m_Λ. Accounting for this term makes the analysis rather clumsy. A more direct way to translate the results of Theorem 6.1.9 to the adaptive sampling point sets X^m_Λ actually used in computations of Algorithm 6.1.7 is to consider that for any Λ and any m, one can always find two uniform grids $X^{m_1}_*$, $X^{m_2}_*$ fulfilling

$$X^{m_1}_* \subseteq X^m_\Lambda \subseteq X^{m_2}_*. \tag{6.1.89}$$

If we assume that the approximation error for a given configuration obtained with some set of sampling points X decreases if we add further observation points to X, we have

$$\hat{\epsilon}_{\Lambda,X^{m_2}_*} \leq \hat{\epsilon}_{\Lambda,X^m_\Lambda} \leq \hat{\epsilon}_{\Lambda,X^{m_1}_*} \tag{6.1.90}$$

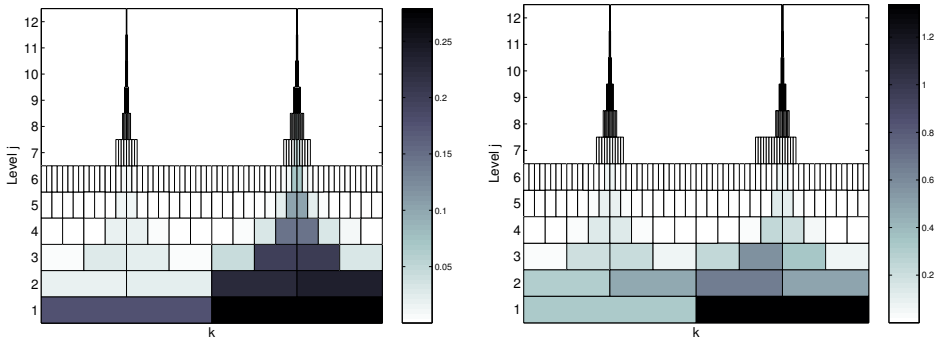
and the asymptotics $\lim_{m\to\infty} \hat{\epsilon}_{\Lambda,X^m_\Lambda} \lesssim \epsilon_\Lambda$ follows from this and (6.1.88) .

In the practice, moderate choices of m do assure the behavior (6.1.88) even for adaptive point distributions X^m_Λ. We illustrate this with the following example.

Example 6.1.10.

$$u(x) = \begin{cases} \exp(4x-2) & x \in \left(\frac{1}{4}, \frac{3}{4}\right) \\ 0 & x \in (0,1)/\left(\frac{1}{4}, \frac{3}{4}\right) \end{cases}, \mathcal{G}(\cdot) = 4\sin(2\cdot). \tag{6.1.91}$$

Figure 6.1.1 shows the energy distribution of $u(x)$ and $\mathcal{G}(u(x))$. In order to have access to the true errors, we have computed an accurate approximation $\tilde{u}$ to u projecting it onto a fine grid (131073 points). We project then $\mathcal{G}(\tilde{u})$ on the same grid to get an accurate approximation $\tilde{\mathcal{G}}(\tilde{u})$. We will use the so computed coefficients as reference values.



Spectrum of u Spectrum of $\mathcal{G}(u)$

Figure 6.1.1: Wavelet coefficients of the functions in Example 6.1.91. For levels $j > 6$, only wavelet coefficients with absolute value above 10^{-4} are displayed.

We take now a series of wavelet configurations Λ_ϵ of decreasing truncation error ϵ with respect to $\mathcal{G}(u)$ and for each ϵ we construct a Least Squares approximation using its point representation of order m, $X^m_{\Lambda_\epsilon}$ for several values of m. The so obtained approximation errors $\hat{\epsilon}_{[\Lambda_\epsilon, X_{\Lambda^m_\epsilon}]}(\mathcal{G}(u))$ are given in Table 6.1.1 for several values of ϵ and m . As a first observation, we see that the point representation of order 0 does not improve the approximation error as the truncation error decreases. As we remarked before, this type of sampling does not provide an invertible system.

For point representations of higher order, enlarging the configuration Λ_ϵ does improve the approximation error provided by Least Squares. This effect can be better appreciated in Figure 6.1.2.

Theorem 6.1.9 assures that the LS approximation error $\hat{\epsilon}$ corresponding to a given truncation error ϵ has to lie in the shadowed area under the line $\hat{\epsilon} = \sqrt{2}\check{\sigma}^{-1}_{\mathbf{M}} C^2_\Psi \epsilon$ assuming that m is large enough. The figure shows that moderate choices of m already attain the asymptotic behavior. A further observation is that asymptotics occurs the sooner, the larger is the truncation error, in the left side of the plot. As we move to the right, we need a larger m to get both into the shadowed area predicted by theory, and into the domain where the approximation error effectively

grows linearly with the truncation error. This effect is expectable: m is the order of the coarsest refinement that, roughly speaking, allows to approximate integrals with summatories within an uniform tolerance η for *all* the configuration. The simpler is the configuration, the smallest is the required m.

ϵ	$\#\Lambda_\epsilon$	$\hat{\epsilon}^{(0)}$	$\hat{\epsilon}^{(1)}$	$\hat{\epsilon}^{(2)}$	$\hat{\epsilon}^{(3)}$	$\hat{\epsilon}^{(4)}$	$\hat{\epsilon}^{(10)}$	$\hat{\epsilon}^{(*)}$
1e-01	41	1.3	6.1e-02	3.3e-02	2.1e-02	1.8e-02	1.8e-02	1.8e-02
1e-02	116	2.2	1.3e-02	4.1e-03	2.6e-03	2.5e-03	2.2e-03	2.0e-03
1e-03	215	2.2	2.3e-03	5.3e-04	2.5e-04	2.2e-04	1.9e-04	1.8e-04
1e-04	358	2.3	1.2e-03	1.7e-04	2.3e-05	1.6e-05	1.3e-05	1.3e-05
1e-05	683	2.3	1.3e-03	1.6e-04	6.9e-06	1.2e-06	9.2e-07	9.1e-07
1e-06	1592	2.3	1.4e-03	2.1e-04	6.7e-06	8.0e-08	6.1e-08	6.0e-08
1e-07	4059	2.3	9.6e-04	8.1e-05	1.5e-06	8.0e-09	3.8e-09	3.7e-09
1e-08	12574	2.3	3.4e-04	2.4e-05	4.2e-07	1.0e-09	2.3e-10	2.3e-10

Table 6.1.1: Approximation error $\hat{\epsilon}^{(m)} := \hat{\epsilon}_{[\Lambda_\epsilon, X_{\Lambda_\epsilon^m}]}(\mathcal{G}(u))$ attained approximating $\mathcal{G}(u)$ from Example 6.1.10 with configurations Λ_ϵ decreasing truncation error ϵ and with points representations of increasing order m. $\hat{\epsilon}^{(*)}$ represents the use of the full reference grid as sampling geometry.

ϵ	$\#\Lambda_\epsilon$	$\#X^0_{\Lambda_\epsilon}$	$\#X^1_{\Lambda_\epsilon}$	$\#X^2_{\Lambda_\epsilon}$	$\#X^3_{\Lambda_\epsilon}$	$\#X^4_{\Lambda_\epsilon}$	$\#X^{10}_{\Lambda_\epsilon}$	# full grid
1e-01	41	35	64	127	253	505	26625	131073
1e-02	116	88	160	319	625	1185	40961	131073
1e-03	215	140	260	519	985	1849	61441	131073
1e-04	358	210	401	801	1517	2857	93185	131073
1e-05	683	371	724	1447	2777	5297	102401	131073
1e-06	1592	836	1647	3293	6425	12497	106497	131073
1e-07	4059	2063	4104	8207	16205	31961	110593	131073
1e-08	12574	6320	12618	25235	44849	59193	114689	131073

Table 6.1.2: Number of sampling points used when approximating the function composition $\mathcal{G}(u)$ from Example 6.1.10 with decreasing truncation error and with increasingly denser sampling sets .

Remark 6.1.11. *As a short comment about the number of sampling points used at each configuration, given in Table 6.1.2, we recall from Chapter 2, that checking if every wavelet is X–balanced gives sufficient but not necessary conditions to ensure the nonsingularity of the normal equations. In view of Proposition 6.1.6, $m = 1$ is the first m that guarantees that the use of a point representation of this order provides stable normal equations, but not necessarily that this point representation contains the minimal number of observations necessary for the invertibility*

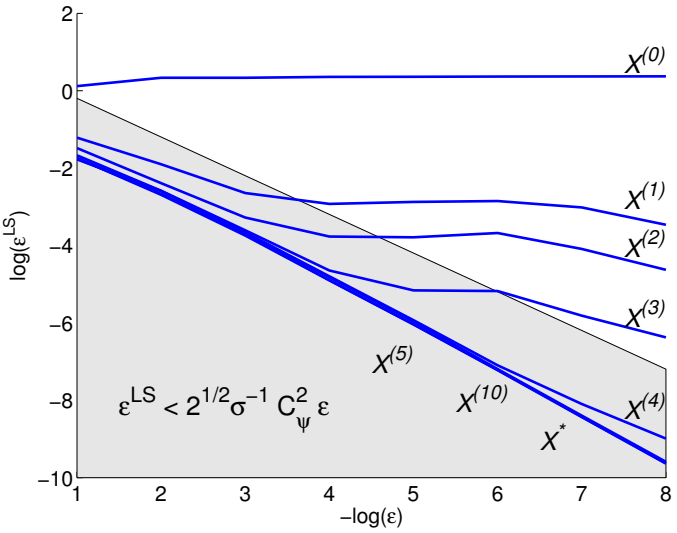


Figure 6.1.2: Decay rate of the Least Squares approximation error of the function $\mathcal{G}(u)$ with $\mathcal{G}$ and u defined in Example 6.1.10 for increasing larger configuration Λ_ϵ. The abscissas represent (minus logarithm of the) truncation error ϵ of a configuration Λ_ϵ. $\mathcal{G}(u)|_{\Lambda_\epsilon}$ is approximated in a Least Squares sense yielding the approximation error ϵ^{LS} represented in the vertical values. Each line represents a different order of the point representation of Λ_ϵ employed as set of sampling points in the LS algorithm. Here, X^* represents the full grid of $2^{17} = 131072$ points used as a benchmark.

of $A^T_{\Lambda,X} A_{\Lambda,X}$ (which must be equal to the number of degrees of freedom, $\#\Lambda$). However, in the practice the order $m = 1$ is tightly related to the minimal conditions of invertibility, as $m = 0$ gives indeed rise to singular equations, and the cardinality of point representations X^m_Λ of order $m = 1$ is only slightly larger than $\#\Lambda$, see Table 6.1.2.

6.2 The Burgers' Equation

The Burgers' equation, formulated as

Problem 6.2.1. Burgers' Equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) = 0 \text{ on } \Omega_T := (0,T) \times (0,1), \tag{6.2.1}$$

$$u(0,t) = u(1,t) \text{ in } (0,T) \text{ and } u(x,0) = g \text{ in } (0,1) \tag{6.2.2}$$

with appropriate boundary and initial conditions, was introduced as a simplified model for fluid flow [17] and provides an interesting benchmark to test the approach described above. Certain features of its numerical behavior make the use of

a wavelet-ansatz, as done by several authors (see e.g. [29], [95], [112], [145]), a very favorable choice:

- **Singularity formation.** Depending on initial conditions, the solutions are known to develop sharp structures which may give rise to discontinuities, remaining smooth away from these areas. This makes adaptive methods capable to concentrate degrees of freedom in the neighborhood of the shock a natural way for the numerical treatment of this problem.

- **Gibbs Oscillations.** Spectral solutions induce Gibbs oscillations near a discontinuity, which causes the method to show poor convergence, unless some *post-processing* is performed in order to smooth the Gibbs artifacts out. Post processing schemes can be enhanced by locating the discontinuities (as in [79]), which can be easily done in the wavelet representation.

- **Frequency localized regularization.** The treatment of the Burger's equation using classical Galerkin finite element methods causes strong numerical instabilities. Often a stabilizing term in the form of a diffusion operator is artificially added to the equation, along the lines of the elliptic regularization theory. This procedure sacrifices too much accuracy. Better results are attained by *spectral viscosity* methods, in which the diffusion is added only to the highest frequencies (see [138]).

  Of particular interest for the wavelet framework are recent analytical convergence techniques ([19],[20]) envisaging the characterization of solutions computed with bases of multiscale finite elements. In principle, the regularization can be restricted to the highest frequencies *and* to the spatial proximity of the singularity.

In the remainder of this chapter, we will adapt the formulation given on [19] and [20] to construct a converging wavelet-based method to solve numerically the Burgers' equation.

We will then operate an elementary smoothing on the attained solution based on elementary properties of the wavelet description of singularities.

After having studied in these two points the *quality* attainable in the wavelet formulation, we will turn our attention to its *efficiency* by proposing and testing an adaptive algorithm that aims to concentrate the degrees of freedom in the neighborhood of the singularity, following it as it evolves with time.

6.2.1 Convergence of the Restricted Viscosity Method

First, we recall in a simplified way the main result attained in [19] and [20] about the convergence of approximations to the solution of (6.2.1) constructed with uniform

grids of hierarchical finite elements. In the monovariate case, the ansatz function of the approach can be written as

$$u_J(x,t) = \sum_{\lambda \in \Lambda_J} d_\lambda^{hb}(t)\psi_\lambda^{hb}(x), \tag{6.2.3}$$

where the $\{\psi_\lambda^{hb}\}_{\lambda\in\Lambda_J}$ are the hierarchical basis defined in Section 3.1.1 for the space V_J of continuous functions piecewise linear on each subinterval $2^{-(j)}[i, i+1]$ for $i = 0, \ldots, 2^j - 1$ (see Figure 3.1.1).

The hierarchical basis approximation of (6.2.1) is thus given by a $u_J(x,t), u_J(\cdot,t) \in V_J, u_J(\cdot,0) = g_J$ that meets the *regularized weak formulation*

$$\int_0^1 \left[\frac{\partial u_J}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u_J^2}{2}\right)\right] v dx + \nu_J \int_0^1 \left[\frac{\partial}{\partial x}\left(Q_J^{j_{cut}} u_J\right)\frac{\partial v}{\partial x}\right] v dx = 0 \tag{6.2.4}$$

for every $v \in V_J$. Here, the operator $Q_J^{j_{cut}}$ eliminates the components whose scale is smaller than j_{cut}, that is

$$Q_J^{j_{cut}}\left(\sum_{\lambda\in\Lambda_J} d_\lambda^{hb}\psi_\lambda^{hb}\right) = \sum_{\lambda\in\Lambda_J, |\lambda|\geq j_{cut}} d_\lambda^{hb}\psi_\lambda^{hb} \tag{6.2.5}$$

and g_J is a projection of g on V_J. With these ingredients, we can state the following convergence result.

Theorem 6.2.2. *([20]) Let $\{u_J\}_{J=j_0}^\infty$ denote a sequence of hierarchical basis approximations determined by 6.2.4. Assume that u_J is uniformly bounded in $L_\infty(\Omega_T)$ and that*

$$\nu_J \to 0 \text{ as } J \to \infty, \tag{6.2.6}$$

$$\nu_J 2^J \geq C, \tag{6.2.7}$$

$$\sqrt{\nu_J}\left\|\frac{\partial}{\partial x}\left[(I - Q_J)u_J\right]\right\|_{L_2(0,1)} \lesssim \|u_J\|_{L_2(0,1)}, \tag{6.2.8}$$

$$\sqrt{\nu_J}\left\|\frac{\partial}{\partial x}\left[(I - Q_J)\frac{\partial u_J}{\partial t}\right]\right\|_{L_2(0,1)} \lesssim \left\|\frac{\partial u_J}{\partial t}\right\|_{L_2(0,1)}, \text{ and} \tag{6.2.9}$$

$$\left\|\frac{d}{dx}\left(Q_J g_J\right)\right\|_{L_2(0,1)} \lesssim \left\|\frac{dg_J}{dx}\right\|_{L_2(0,1)}. \tag{6.2.10}$$

Then there exists a subsequence of $\{u_J\}_{J=j_0}^\infty$ that converges strongly in $L_2(\Omega_T)$ to a solution $u \in L_2(\Omega_T)$ of 6.2.1.

Note that in the full grid approximation the convergence results expressed in Theorem 6.2.2 are directly applicable to a wavelet formulation. A u_J satisfying (6.2.4) can be expressed as

$$u_J(x,t) = \sum_{\lambda \in \Lambda_J} d_\lambda(t)\psi_\lambda(x), \tag{6.2.11}$$

where the $\{\psi_\lambda\}_{\lambda\in\Lambda_J}$ is the linear spline–wavelet basis of V_J, as noted in Section 3.1.1. The operator $Q_J^{j_{cut}}$ that kills oscillations on scales smaller than j_{cut} has the same form as in (6.2.5)

$$Q_J^{j_{cut}}(\sum_{\lambda\in\Lambda_J} d_\lambda\psi_\lambda(x)) = \sum_{\lambda\in\Lambda_J,|\lambda|\geq j_{cut}} d_\lambda\psi_\lambda(x). \tag{6.2.12}$$

Particularly, we can make a simple choice of parameters that guarantee asymptotic convergence of the (Petrov-) Galerkin method in the wavelet basis.

Proposition 6.2.3. *With the choices $\nu_J = C2^{-J}$ and $j_{cut} \leq \frac{J}{2}$, the assumptions (6.2.6) to (6.2.9) on a function u_J of the form (6.2.11) satisfying the regularized weak formulation (6.2.4) are met.*

Proof:
For all $v \in V_J$ we have

$$(I - Q_J^{j_{cut}})v = \sum_{\lambda\in\Lambda,|\lambda|<j_{cut}} d_\lambda\psi_\lambda \tag{6.2.13}$$

As

$$\sum_{\lambda\in\Lambda,|\lambda|<j_{cut}} |d_\lambda|^2 \leq \sum_{\lambda\in\Lambda} |d_\lambda|^2, \tag{6.2.14}$$

by the Riesz property $\sum_{\lambda\in I\!I} |g_\lambda|^2 \sim \|g\|^2_{L_2(0,1)}$ that holds for all $g = \sum_{\lambda\in I\!I} g_\lambda\psi_\lambda$ we can write

$$\|(I - Q_J^{j_{cut}})v\|_{L_2(0,1)} \lesssim \|v\|_{L_2(0,1)}. \tag{6.2.15}$$

Now, $(I - Q_J^{j_{cut}})v$ is linear on intervals of length $2^{-(j_{cut}+1)}$, so that

$$\left\| \frac{\partial}{\partial x}\left[(I - Q_J^{j_{cut}})v\right]\right\|_{L_2(0,1)} \lesssim 2^{j_{cut}} \left\|(I - Q_J^{j_{cut}})v\right\|_{L_2(0,1)}. \tag{6.2.16}$$

Multiplying by $2^{-\frac{J}{2}}$ and using (6.2.15), we have

$$2^{-\frac{J}{2}} \left\| \frac{\partial}{\partial x}\left[(I - Q_J^{j_{cut}})v\right]\right\|_{L_2(0,1)} \lesssim 2^{-\frac{J}{2}+j_{cut}} \|v\|_{L_2(0,1)}. \tag{6.2.17}$$

Since we have chosen $j_{cut} \leq \frac{J}{2}$, taking $v \in \{u_J, \frac{\partial u_J}{\partial t}\}$, conditions (6.2.8) and (6.2.9) are satisfied. ∎

6.2.2 Smoothing of Gibbs Phenomenon

As model problem, we consider the following form of the Burgers' equation (6.2.1) with particular boundary and initial conditions.

Problem 6.2.4. Stationary Shock

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial}{\partial t}\left(\frac{u^2}{2}\right) &= 0 && \text{on } (0,1)\times(0,T), \\ u(0,t) &= u(1,t) && \text{for all } t \in (0,T). \\ u(x,0) &= \sin(2\pi x) && \text{for } x \in (0,1), \end{aligned} \tag{6.2.18}$$

which gives rise to a stationary shock: As time increases, the derivative of u grows larger in a decreasing neighborhood of $x = 0.5$, becoming a sharp jump. After this singularity is formed, the solution's amplitude starts to decrease.

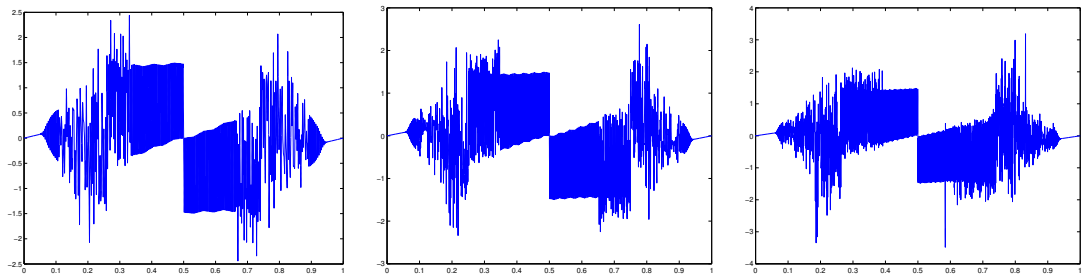
We see in Figure 6.2.3 on the left the solution computed with a slight modification of our (preorthogonal spline–)wavelets. In order to meet the periodic conditions, we have substituted the boundary adaption with a periodization of the basis ensuring that $\psi_\lambda(0) = \psi_\lambda(1)$ for all $\lambda \in \Lambda$. We have used the full grid for increasing maximal level of resolution J at time $t = 0.5$, where the sharp shock has been formed and is about to start reducing its amplitude.

As expected, numerical instability appears if no viscosity is added, see Figure 6.2.1 and no convergence occurs by refining the ansatz space. Results of Section 6.2.1 ensure convergence if viscosity is added at least to the uppest $J/2$ levels of a discretization with a maximal level of resolution J. This is confirmed by Figure 6.2.2. An importat point is that the frequency-restricted viscosity method operates beyond the convergence results from Section 6.2.1: Numerical experiments (see 6.2.3) show that adding viscosity *only* to the highest resolution level does already stabilize the computed solution. This yields better accuracy away from the singularity, as less artificial diffusion is introduced into the original problem. However, an oscillation due to the Gibbs phenomenon still concentrates around the singularity. Although its support decreases as we improve the resolution, this artifact remains for every J.

Wavelets and singularities

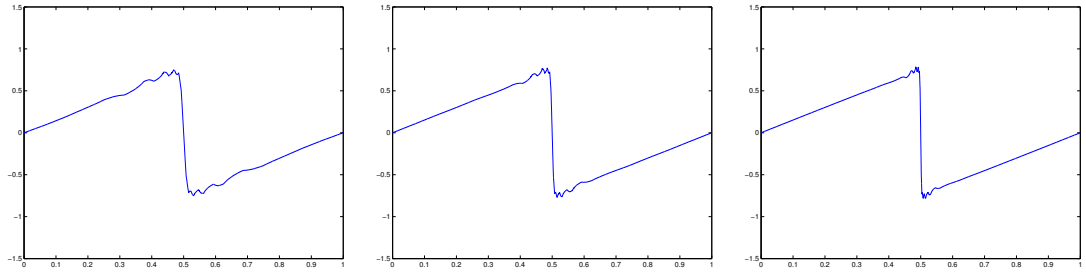
Description and detection of singularities by wavelet analysis has been envisaged (see [96], [100], [101], [103], [117], [118]) by numerous authors even since their introduction in the late 80's. Typical results herein relate the local Hölder or Lipschitz regularity of a function with the decay across scales of its local wavelet coordinates.

Now, in order to gain a look on the qualitative behavior to which one should compare the behavior of coefficients of the reconstructions in Figure 6.2.3, we briefly provide an example of the representation of a discontinuity in the wavelet basis that we are using. This will help us to disentangle the effects of the discontinuity from the



$J = 8$ $J = 9$ $J = 10$

Figure 6.2.1: Wavelet solution for the Stationary Shock Problem 6.2.4 using different resolutions. No viscosity is applied.



$j_{cut} = 3, J = 8$ $j_{cut} = 4, J = 9$ $j_{cut} = 5, J = 10$

Figure 6.2.2: Wavelet solution for the Stationary Shock Problem 6.2.4 using different resolutions. At each resolution J, a j_{cut} that meets the requirements of Theorem 6.2.2 for convergence of the Galerkin methode is applied.

effects of the Gibbs phenomenon by inspection of the coefficients in the proximity of the jump at $x = 0.5$.

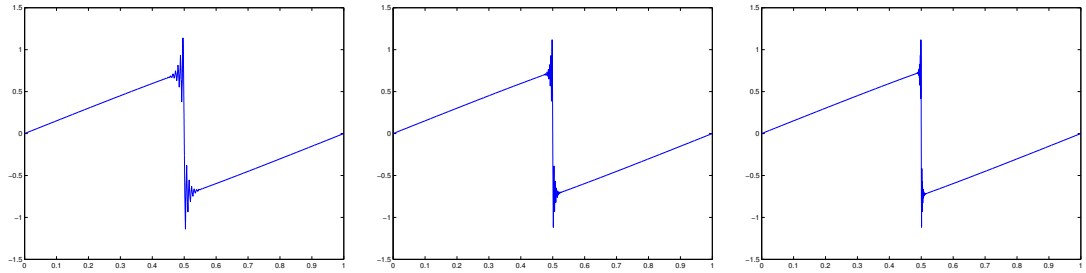
Example 6.2.5. *Decay profile of wavelet coefficients across a singularity. Let $f(x)$ be a linear function except for a discontinuity located at $x_0 = 1/2$, that is*

$$f(x) = \begin{cases} x, & x \le x_0, \\ x - 1, & x > x_0. \end{cases} \tag{6.2.19}$$

We denote its wavelet expansion as

$$f(x) = \sum_{j>0;\ k=0,\ldots,2^j-1} d_{j,k}\psi_{j,k}(x). \tag{6.2.20}$$

In view of the preorthogonal spline–wavelets employed here, the $L_2(0,1)$ scalar product of the function against all wavelets from level j are on each level zero except for the two wavelets whose support cuts the point $x_0 = 1/2$, see the left graphic in



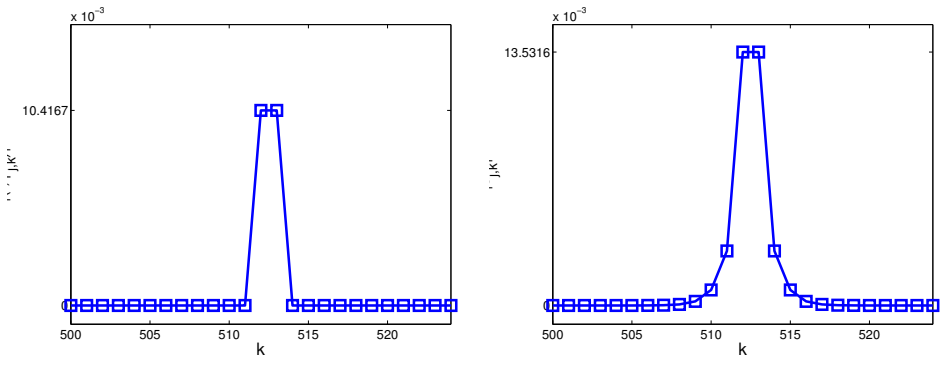
$j_{cut} = 8, J = 8$ $j_{cut} = 9, J = 9$ $j_{cut} = 10, J = 10$

Figure 6.2.3: Wavelet solution for the Stationary Shock Problem 6.2.4 using different resolutions. Viscosity is applied only on the maximal resolution level.

Figure 6.2.4 for the case $j = 10$. *Taking scalar products in* L_2 *of (6.2.20) against each wavelet on level* j', *we obtain*

$$(f, \psi_{j',k'}) = \sum_{j>0;\ k=0,\dots,2^j-1} d_{j,k}(\psi_{j,k}, \psi_{j',k'}), \qquad k' = 0, \dots, 2^{j'} - 1. \tag{6.2.21}$$

One can solve now for the wavelet coefficients $\{d_{j',k}\}_{k=0,\dots,2^{j'}-1}$ *of the expansion in (6.2.20) and obtain the right plot in Figure 6.2.4. Note that multiplication of the wavelet coefficients by the entries of the mass matrix* $(\psi_{j,k}, \psi_{j',k'})$ *introduces a smooth decay.*



Behavior of $|\int_0^1 f(x)\psi_{j,k}(x)dx|$ against k at level $j=10$.

Behavior of $|d_{j,k}|$ against k at level $j=10$

Figure 6.2.4: Behavior of wavelet coefficients of a given level ($j = 10$) across a discontinuity in a linear function.

Post Processing

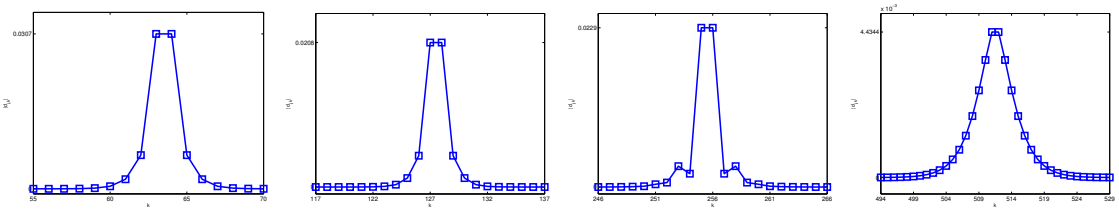
In the following, we describe an elementary technique of smoothing applied to the reduction of Gibbs' artifacts on the basis of (preorthogonal spline–)wavelets: one identifies the areas in which this phenomenon is active, eliminates the responsible wavelet coefficients and recomputes the remaining ones.

This identification is based on the inspection of the decay of wavelet coefficients at each scale, as represented in Figure 6.2.5. The shape of the decay of the wavelet coefficients $|d_{j,k}|$ across k in $j = 7$, $j = 8$ and $j = 9$ is similar to the decay shown by wavelet coefficients representing a singularity. In contrast to this, the decay of $|d_{10,k}|$ at level 10 across k is qualitatively different: since it is too wide, it does not match the decay expected from a clean discontinuity, as seen in Example 6.2.5. Let us denote by Λ_{Gibbs} the set indices of wavelets whose coefficients appear to be affected by the Gibbs phenomenon. The decay across k of the wavelets coefficients signalizes an energy concentration along an extended interval $I_{Gibbs} := \cup_{\lambda\in\Lambda_{Gibbs}} \operatorname{supp} \psi_\lambda$. We assume that it marks approximately the support of the artifact, and that the reconstruction on this interval is henceforth of no use. Now, one simply performs a local smooth re-sampling of the signal using only points lying outside I_{Gibbs} and computes the wavelet coefficients with respect to this new re-sampling. A possible choice is $X_\Lambda^{(m)}/X_{\Lambda_{Gibbs}}^{(m)}$ for any $m > 0$, being Λ the original wavelet configuration. The right plot in Figure 6.2.6 shows the reconstruction attained by this elementary technique for the solution of highest resolution in Figure 6.2.3. Figure 6.2.7 shows that this strategy yields different results from simply discarding coefficients of the higher scales: the coefficients on levels $j = 9, 10$ do indeed disappear (and they are not depicted), but also coefficients of the remaining levels are changed. In fact, if one just threw out wavelet coefficients from higher scales, one would just rescale the Gibbs phenomenon and get a wider I_{Gibbs} (see Figure 6.2.3).

We see therefore that the Gibbs phenomenon is not located exclusively on the two upper levels ($j = 9, 10$). It leaks down through the lower scales. The correction is henceforth stronger for higher levels and fades away as we move down in the dyadic scale of resolution (compare plots for $j = 5$ and $j = 8$).

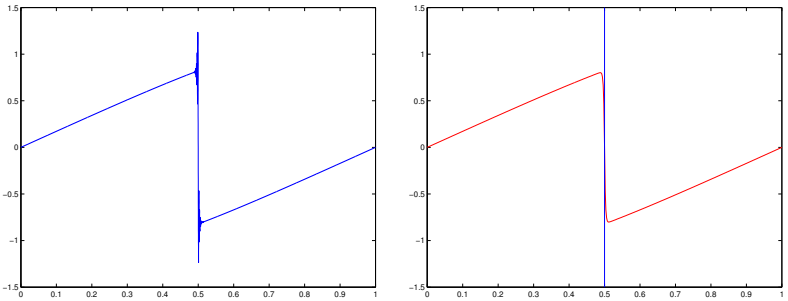
Obviously, this procedure decreases the sharpness of the peak: the final resolution in the example is $j = 8$ after this post–processing. This does not mean however, that the work done to resolve the discontinuity at level $j = 10$ is lost. In fact, if we started with a signal resolved at level $j = 8$, the smoothing would not take into account the information coming from the two last levels and we would just produce a rescaled version of the Gibbs phenomenon.

Remark 6.2.6. *Starting with a given finest level j, if one wants to suppress the artifacts and simultaneously wants to keep a sharp resolution of the discontinuity up to some level, one would need more involved techniques. Among powerful strategies from Image Processing, there is minimization of Total Variation [64, 65]), or the so–called footprint location [63], which are aimed at squeezing out the last drop of information from the given set of available points. As we are working with the*



$j = 7$ $j = 8$ $j = 9$ $j = 10.$

Figure 6.2.5: Decay of coefficients on the last four levels before any smoothing technique is applied.



Without smoothing. With elementary smoothing.

Figure 6.2.6: Smoothing the solution of level $J = 10$ of the Stationary Shock (6.2.4).

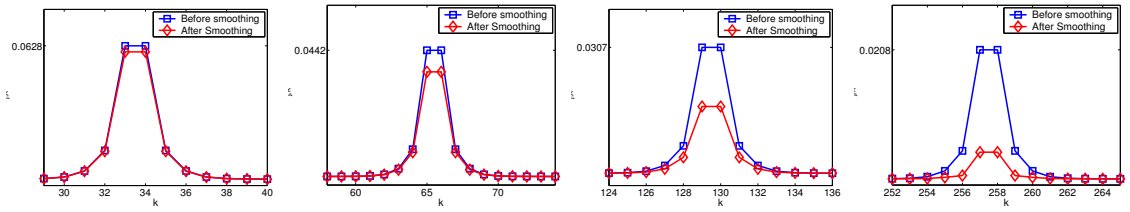
output of a PDE problem, we could gain additional information by simply producing solutions of higher resolution by locally adding degrees of freedom.

This analysis works for singularities arising in other situations: The following Problem develops a sharp shock that travels from left to right.

Problem 6.2.7. Traveling Shock

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial}{\partial t}\left(\frac{u^2}{2}\right) &= 0 && \text{on } (0,1)\times(0,T), \\ u(0,t) &= u(1,t) && \text{for all } t \in (0,T), \\ u(x,0) &= 1 + \tfrac{1}{2}\sin\left(\pi\left(2x-1\right)\right) && \text{for } x \in (0,1). \end{aligned} \tag{6.2.22}$$

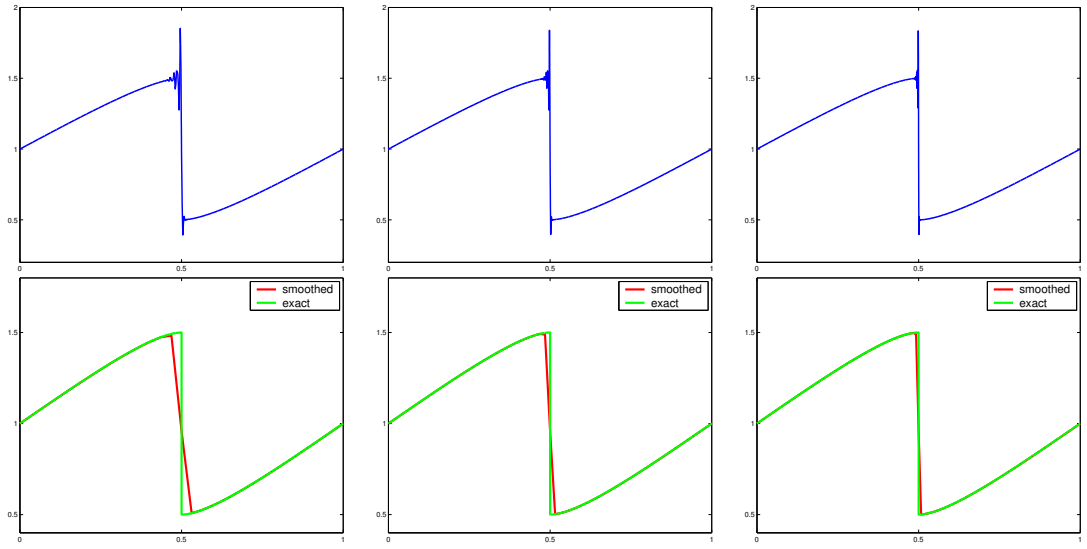
This is the same example as in [20]; exact solution is implicitly given by solving a nonlinear equation [19]. The numerical solution is computed here with (preorthogonal spline–)wavelets (uniform discretization) and explicit Euler's method for time



$j = 5$ $j = 6$ $j = 7$ $j = 8.$

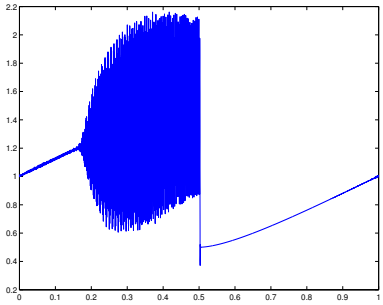
Figure 6.2.7: Decay of coefficients on the last four levels after smoothing compared to their original values.

discretization which yields qualitatively the same results as more involved schemes provided that the time step is small enough. The results at time $t = 0.5$ using diffusion on finest level only are displayed in Figure 6.2.9 for different values of J, illustrating the convergence of the method as the resolution level increases. Figure 6.2.9 shows the instabilities that appear in absence of viscosity.



$J = 8$ $J = 9$ $J = 10$

Figure 6.2.8: Wavelet solution with the full grid for Traveling Shock. The upper row shows the results at T=0.5 for different resolution levels. The lower row shows the results after smoothing, compared to the real solution.



$J = 10, T = 0.5$

Figure 6.2.9: Instabilities arising in absence of viscosity when solving Traveling Shock.

6.2.3 Adaptive Solution

Although the convergence results of Section 6.2.1 apply for a full grid discretization, we have found that also adaptive schemes benefit from the frequency-located addition of viscosity, as numerical instabilities do indeed disappear, as the numerical evidence shows.

We have implemented a slight modification of the adaptive algorithm proposed in [104]. The approximation of the solution $u(m\delta t, \cdot)$ at time $m\delta t$ is given by a $u^m = \sum_{\lambda \in \Lambda^m} d_\lambda^{m+1} \psi_\lambda$. The configuration Λ^m is updated by a dynamically adaptive scheme as described in Section 6.1.2, and the time evolution is computed by an Adams-Basforth scheme of second order.

Algorithm 6.2.8. Adaptive Wavelet-Galerkin Scheme

Fix a thresholding $\epsilon(j)$ possibly dependent on the resolution level. Given an initial solution u^0 expanded in the wavelets of some set $\tilde{\Lambda}^0$, define Λ^0 by thresholding $\tilde{\Lambda}^0$, i.e. $\Lambda^0 := \{\lambda \in \tilde{\Lambda}^0 : |d_\lambda| > \epsilon(0)\}$ and perform for each $m = 0, \ldots,$:

1. **Refinement**

   (a) *Vertical Expansion*

   The configuration at the next time step may require wavelets of a higher level. We construct a set Λ_v^m containing the wavelets in Λ^m plus all their children.

   (b) *Horizontal Expansion*

   We extend the configuration with the horizontal neighbors of the wavelets of the previous set and form $\tilde{\Lambda}^{m+1}$.

$$\tilde{\Lambda}^{m+1} := \left\{ \lambda = (j,k) \in \tilde{\Lambda}^m : \exists (j,k') \in \Lambda_v^m,\ k' \in \{k-2, k-1, k, k+1, k+2\} \right\}. \tag{6.2.23}$$

2. **Computation**

We look now for a function

$$u^{m+1} = \sum_{\lambda \in \tilde{\Lambda}^{m+1}} d_\lambda^{m+1} \psi_\lambda \tag{6.2.24}$$

that solves

$$\left(\frac{1}{\tau} \left(u^{m+1} - u^m \right) + \frac{1}{2} \frac{\partial}{\partial x} \left(u_*^m \right)^2, \psi_\lambda \right) = -\nu \left(\frac{\partial}{\partial x} Q_J(u_*^m), \frac{\partial}{\partial x} \psi_\lambda \right), \ \textit{for } \lambda \in \tilde{\Lambda}^{m+1}, \tag{6.2.25}$$

with $u_*^m = \frac{3}{2} u^m - \frac{1}{2} u^{m-1}$, *(or* $u_*^m = u^m$ *for* $m = 0$*).*

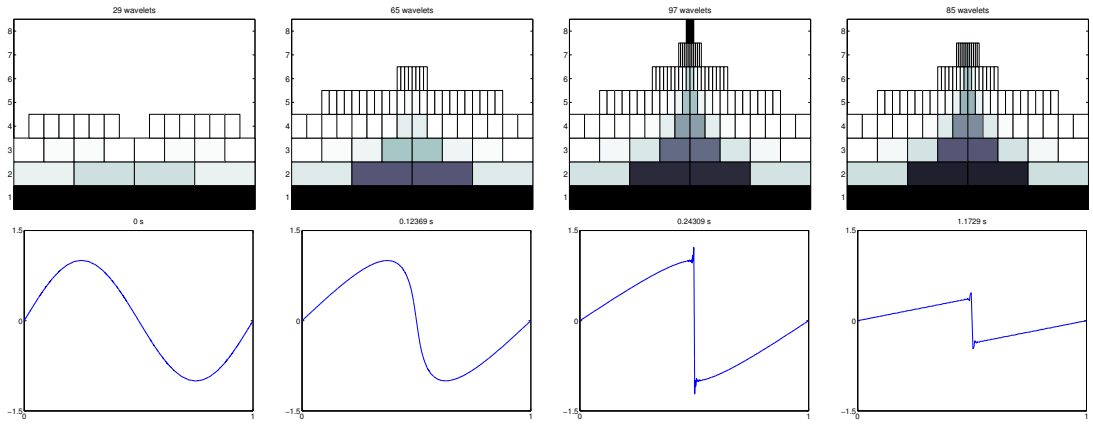
3. **Coarsening**

Compute Λ^{m+1} *by thresholding* $\tilde{\Lambda}^{m+1}$

$$\Lambda^{m+1} := \left\{ \lambda \in \tilde{\Lambda}^{m+1} : |d_\lambda| > \epsilon(|\lambda|) \right\}. \tag{6.2.26}$$

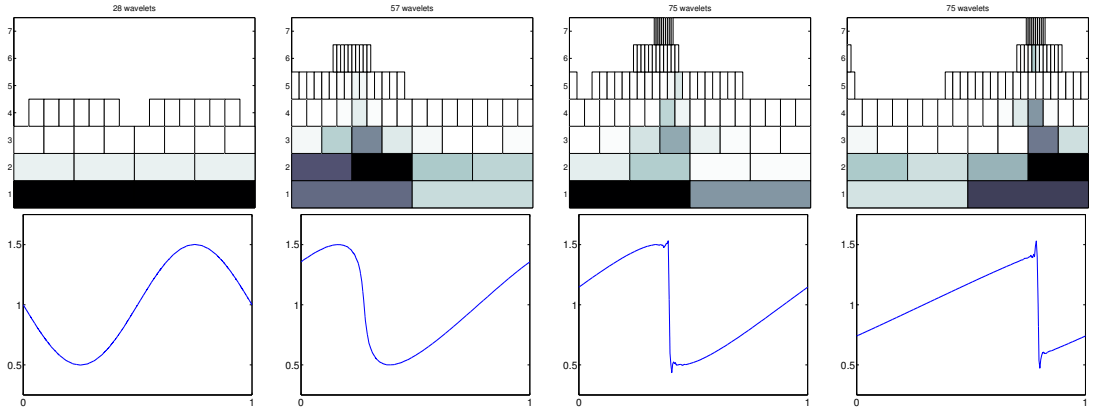
We have solved the Burgers' equation with the two different initial conditions proposed in the previous section with Algorithm 6.2.8. In both cases we use the following parameter set: $J_{max} = 10$, $\nu = 2^{J_{max}-1}$, $\epsilon(j) = 10^{-6} \cdot 2^{2j}$. We let the viscosity act only on the elements with $j > 6$. The results, in Figure 6.2.10 for the Stationary Shock and 6.2.11 for the Traveling Shock, appear to catch perfectly the dynamics of the solution.

In both cases, the initial wavelet structure expands itself to solve the forming shock, as the method selects wavelets in its neighborhood of the shock. The prescribed thresholding policy prevents the method to accumulate wavelets of higher resolution as $j = 8$. In the stationary case, the shocks remains at $x = 0.5$ and vanishes, leading to a simplification of the wavelet structure. In the traveling case, the wavelet structure travels with the shock.



$t = 0$ $t = 0.12$ $t = 0.24$ $t = 1.17$

Figure 6.2.10: Evolution of the wavelet configuration selected by the adaptive method applied to the Stationary Shock Problem 6.2.4. Corresponding solutions are displayed below.



$t = 0$ $t = 0.28$ $t = 0.39$ $t = 0.78$

Figure 6.2.11: Evolution of the wavelet configuration selected by the adaptive method applied to the Traveling Shock Problem 6.2.7. Corresponding solutions are displayed below.

Chapter 7

Implementation and Data Structures

The computations involved in the data fitting algorithms described in Chapter 2 require managing sparse matrices, as both observation and coefficient matrices present this kind of structure. Any computations with these objects should be based on adequate data structures that can favorably use this fact. The structure of the matrices is determined by the superposition of two effects.

- First, we have the deterministic sparsity induced by the locality of wavelets, that prevents every wavelet to be coupled to every other wavelet in the $A^T A$ matrix. This is the sparsity that appears already when working with the full grid. The completeness of the grid allows for a very efficient managing with static or semi-static data structures, as the ordering by indices (j, k) or (j, k_x, k_y, e) induces easily a well defined indexing to represent in memory the degrees of freedom arising in the problem and their coupling.
- Secondly, the final sparsity structure of the arising matrices is determined by two sources of indetermination:
  - the input data is unstructured, and
  - the identity of the involved wavelets is decided on-line in view of the data and the partial analysis of it performed at each growing stage of the tree.

The two factors in the second point (partly related to each other, as the selection of new wavelets relates both to the local smoothness of vertical values *and* the distribution of horizontal coordinates of the points) introduce two kinds of indeterminations.

- a point-in-wavelet uncertainty, derived from the irregularity of the data set. This means that given a wavelet indexed by λ, we cannot find out how many points and which ones are in its support, that is, the set of point indices

$$P_\lambda := \{i \in 1, \ldots, N \ : \ x_i \in \mathrm{supp}(\psi_\lambda)\}, \tag{7.0.1}$$

without traversing the entire data set P.

- a wavelet-in-point uncertainty, derived from the on-line formation of Λ. Given the x_i coordinates of a point, it would be a trivial task to find the set of wavelets out of a *full* grid whose supports cross the point. But in our case, as we do not know beforehand how many wavelets and which ones are going to be selected by the method, we cannot determine beforehand the corresponding set of wavelet indices defined by

$$\Lambda_{x_i} := \{\lambda \in \Lambda \ : \ x_i \in \operatorname{supp}(\psi_\lambda)\} \tag{7.0.2}$$

by accessing the point x_i.

As the main interest of the method is precisely the analysis of sets of highly irregularly distributed points that can foreseeable produce Λ trees which are very different to full grid trees, special attention should be paid to the use of data structures that harmonize the need to dynamical management with efficiency.

7.1 Observation Matrix A

Recall the basic growth algorithm, which intends to select the wavelets containing less than q points, q being a fixed parameter. We start at a root level composed by all the wavelets of the full grid for some fixed scale $j = j_0$ (in our experiments usually selected as $j_0 = 1$). The indices of this level are

$$\Lambda_{root} := \{\lambda = (j_0, k) \ : \ k = 0, \ldots, 2^{j_0} - 1\} \tag{7.1.1}$$

or

$$\Lambda_{root} := \{\lambda = (j_0, k_x, k_y, \mathbf{e}) \ : \ k_x, k_y = 0, \ldots, 2^{j_0} - 1; \ \mathbf{e} \in \{0,1\}^2\} \tag{7.1.2}$$

for two dimensions. Each node $\lambda = \{(j,k) : j = j_0, \text{ every } k\}$ is represented as an object containing the identification of the wavelet and an array indexing the points of P that appear on λ. This initial configuration is created in an initial loop on all the points in P, creating a pointer to the position of point x_i only in the nodes that actually cut the point, that is for $\lambda \in \Lambda_{root}$ so that $\lambda \in \Lambda_{x_i}$. This performs a kind of binning of the points at the scale j_0. Note that this operation has a complexity proportional to the number of points N.

At the next tree growing stage, we want to further bin the data in relation to the children of the nodes Λ_{root}. With this goal, we perform a binning loop on the N_λ many points in P_λ for each $\lambda \in \Lambda_{root}$, checking for each $x \in P_\lambda$ in which children of λ should be created a pointer to x. As the support of each of the children of λ is necessarily included in the support of λ, at the end of the loop, every child λ_{son} of $\lambda \in \Lambda_{root}$ has gotten his $P_{\lambda_{son}}$ set constructed, and the level j_0+1 can be constructed by rejection of the sons that do not attain to contain q-many points at their support.

And we carry on this procedure till having finished the construction of Λ, coding it in a tree structure.

The important thing to note is that this process constructs automatically an optimal bookkeeping of the sparsity structure for the matrix A: As every $\lambda \in \Lambda$ gets its array of indices P_λ connecting to the points in its support, traversing the $nnz(A)$-many nonzero elements of A amounts to traverse the tree, stopping at each node λ to traverse the corresponding points indexed by P_λ. This allows for a fast matrix-vector multiplication naturally adapted to the sparsity structure of A. Moreover, it is done without significant overhead in memory allocation.

7.1.1 Complexity

The complexity of the whole process is proportional to N, as we just accumulate levels containing pointers to a maximum of N points.

This favorable characteristic of the method is due to the fact that relative positions between points never need to be computed. The point-point type information is for us already accounted for in the ordering of the points to different wavelets of Λ *and* in the fact that the relative position between wavelets is already coded implicitly in their indices. Shortly, we need to extract less information from the unstructured cloud of points (than, for instance, when creating a triangulation) because the representation we want to create aims more to catch structures underlying in the collection of points, rather than structuring the cloud of individual points.

7.1.2 Evaluation of Matrix Elements

Once each P_λ is detected and coded in the node λ, one should evaluate the wavelet ψ_λ at the corresponding points and store the values. The option of computing them at every call to each individual element is not advisable as

- the execution of the program is severely slowed down, as every element of the matrix is bound to be used several times, mainly the elements corresponding to the coarse levels, and
- the attained memory gain is rather modest. The bookkeeping of the integers in P_λ for each λ in order to control the sparsity structure is anyway not avoidable. Its amounts to store $nnz(A)$ integers. Now, renouncing to store the $nnz(A)$ float or double variables with the wavelet values brings a gain factor of 2:1 or 3:1 respectively.

7.2 Coefficient Matrix A^TA

The numerical inconveniences that could arise if one chooses to solve the normal equations by actually forming $M = A^TA$ instead of working with A and A^T are dis-

cussed in [27]. Now we discuss some issues concerning the practical implementation of the A^TA approach.

Note that the elements of this matrix should not be computed at each time they are requested when operating a matrix-vector multiplication. Each element codes the result of some analysis of the interplay between the tree and the data. We want to reuse this analysis, not to repeat it continuously.

The tree structure as explained up to now contains the point-in-wavelet information. This is not appropriate for the assembly of the A^TA matrix. New structures have to be tailored to this task. Ideally, the formation of this matrix should follow the schema: For every $1, \ldots, N$:

1. Compute $\Lambda_{x_i} = \{\lambda \ : \ x_i \in \mathrm{supp}(\psi_\lambda)\}$.
2. For every $\lambda \in \Lambda_{x_i}$,
   compute or fetch the wavelet value at point x_i.
3. For every λ,$\lambda' \in \Lambda_{x_i}$,
   accumulate the product $\psi_\lambda(x_i)\,\psi_{\lambda'}(x_i)$ in some memory place accessible to every other point of the list.

This procedure is optimal as it requires uniquely the arithmetic operations contained in the elements of the A^TA matrix. That is, formally these elements are defined as

$$M_{\lambda,\lambda'} = \sum_{i=1}^{N} \psi_\lambda(x_i)\,\psi_{\lambda'}(x_i),$$

but only few elements of the sum do actually contribute with nonzero summands. In fact, the sum could be written as

$$M_{\lambda,\lambda'} = \sum_{i \in P_\lambda \cup P_{\lambda'}} \psi_\lambda(x_i)\,\psi_{\lambda'}(x_i).$$

The above procedure eliminates the bottleneck of search and sort operations that would occur when repeatedly examining the data looking for the points common to each pair $\lambda, \lambda' \in \Lambda$. In the following section we develop the structures that can mimic this procedure.

7.2.1 Construction of Λ_{x_i}

The tree structure is constructed over the wavelets. It only gives us the point-in-wavelet information. We have a direct access to an element λ and can then consult the set P_λ, indexing those data points attached to it. Extracting Λ_{x_i} for each $1, \ldots, N$ requires the creation of an additional structure that could give us direct access to the points. We need a point look up table.

This can be accounted for at the binning stage of each level. Every time a point is accepted in a wavelet node, that is, every time a reference to this point is created

in the memory area of the node, the node has to be correspondingly accepted by the point, that is, a pointer or index that locates the node has to be created in the memory area accessible by the point.

This operation opens the question of how to preallocate enough memory at each point x_i in order to contain the pointers/indices to the nodes Λ_{x_i}. In some situation the following immediate solutions can work:

1. Static allocation of memory for all the λ of the *full grid* that could possibly overlap the point of interest. At each scale j, this quantity is a constant depending of the wavelet family.
   Depending on the characteristics of the data distribution this policy of systematical overestimation can lead to a satisfactorily moderate overhead or to a fantastic waste of memory.

2. Dynamical linking of every new λ accepted into a point x_i. This reduces the overhead at the prize of slowing down both construction and performance of the table by adding the costs of continuous redirecting. Worse, each piece of the linked list is created while inspecting a different wavelet. This means that at each point x_i we can expect a very poor locality. Each piece of the chain is very likely to lie in a position of memory lying very far away of the previous and the next one. Traversing the list multiplies the possibilities of generating a page fault at every memory access.
   Again it is impossible to predict the performance of its approach. It can work just fine or create a bottleneck, depending of the degree of memory fragmentation introduced by the linking.

Although one could think of some hybrid strategy (dynamical linking of blocks containing several addresses), one can use a safer way. One runs a first loop on the points stored at the last level of the tree. We keep track of the number of apparitions of each point in a counter located in the point. We can then dimension an array, which is later filed in a second loop, with exactly the right number of wavelet entries for each point. The performance overhead is always the duplication of the loop.

7.2.2 Accumulation of Point Contributions

The last step of the procedure presents the hardest difficulties. The individual elements of the matrix $M_{\lambda,\lambda'}$ are to be computed by accumulating the individual contributions $\psi_\lambda(x_i)\psi_{\lambda'}(x_i)$ of the points x_i common to P_λ and $P_{\lambda'}$. Traversing the points of the above described data structure assures us to produce efficiently the $\psi_\lambda(x_i)\psi_{\lambda'}(x_i)$ terms. Now, we have the problem of how to accumulate then.

This is a common problem in computational linear algebra. One is given two sparse matrices, in our case A^T and A, and is confronted with the task of multiplying them efficiently with insight to performance, memory and storage.

- The performance efficiency amounts to create an algorithm that directly accesses only those pairs of elements of the factor matrices that produce nonzero elements of the product matrix.

- Under memory efficiency we mean not to preallocate much more memory than actually required by the nonzero elements of the matrix. Recall that the product of two sparse matrices does not have a foreseeable sparsity structure.

- Storage efficiency amounts to adopt a data structure that allows for a coherent access to every one of the accumulators when required by different points.

The problem has not been investigated as extensively as the one matrix-vector multiplication, and one does not have an optimal solution procedure for the general case. A different one has to be tailored for each situation, making use of the special characteristics of the matrices. In our case, the efficiency in the performance is attained by the table structure. When we traverse the available points in the table structure defined above, we access exactly and directly the elements of A and A^T that pairwise contribute to the nonzero entries of M.

The memory efficiency requires a new extension of the data structures mentioned so far. Most matrix multiplication packages that allow for sparse matrix matrix multiplication like MATLAB, SPARSKIT, SparseLib++ or SMMP make a previous global memory allocation that cover every position of the target matrix or preallocate for only a column or row at a time and then fill it, with forces to time consuming loops on the elements of the factor matrices looking for the corresponding elements.

In our case global preallocation can lead to memory overflow when working with data showing rich structures.

The biggest difficulty consists in arranging the location of the accumulators so that they remain equally well accessible when called by different points.

In our setting, each wavelet node $\lambda \in \Lambda$ will contain a static array with as many positions as overlapping neighbors of λ are in the *full* grid, up to the last level of the tree. Each one of these positions will act as an accumulator for the coupling of ψ_λ with the corresponding $\psi_{\lambda'}$.

The static definition allows for these accumulators to be ordered following some natural indexing of the (j, k) indices, independently of the presence of the corresponding wavelets in the tree structure. This makes these accumulators directly accessible to each point, without searching, sorting, hashing or redirecting.

The prize is a memory overhead as some $\lambda' \notin \Lambda$ *will* actually be represented in some nodes of the structure. This overhead is anyway affordable for three reasons.

- The arrays are created only in actual nodes of the tree. This bounds the memory to be preallocated to scale linearly with the actual degrees of freedom, not with the full grid complexity.

- Wavelets neighboring a wavelet present in the tree are anyway likely to be also present.

- After having run over all the points of the table structure, the matrix elements of M need no further to be represented by accumulators. They are not accepting any summands any more, and do not need to be individually accessed. Therefore, once a level is completed the part of the matrix coded in this level can be extracted and stored in a usual way. The arrays can then be freed.

In short, by static managing of arrays kept in dynamically created nodes, we combine the access efficiency provided by deterministic structures with the complexity reduction provided by the data adaptivity of the algorithm, which has to be represented with on-line constructed structures.

7.2.3 Performance Comparison between A and A^TA Methods

As we have seen, operating with the matrix A requires only to use the tree structure that arises naturally in the binning of the data, while the use of A^TA needs the addition of new nontrivial data structures in order not to generate memory overflows. The interest in working with A^TA is the expectable enhancement of efficiency at the Conjugate Gradient stages that take place when extending the tree with a new level. As pointed out before, the individual entries of this matrix code more operations on the data that the entries of A do. In some sense, the result is that the matrix A^TA shows some kind of compression of the information. The dimension of A is $N \times \#\Lambda$, while A^TA has a dimension $\#\Lambda \times \#\Lambda$. As we expect to have much less degrees of freedom than points (especially on the first levels of the tree; compare a typical input of 10^4 points with the 25 wavelets of the first level for the linear wavelets), a matrix-vector multiplication runs much faster in the A^TA matrix.

This advantage balances with the overhead of having to assemble the matrix. Note however that the numerical complexity of computing A^TA scales with N, exactly as computing A . Depending on the number of iterations we need to attain an acceptable minimum, assembling A^TA will pay off or not. Numerical experiences show some general trends:

- Typically wavelets need much less iterations than other hierarchical basis to converge. Moreover, in the case of wavelets the *nesting* strategy allows for the number of iterations needed to attain some reasonable minimum to grow only moderately with the number of levels of the tree. In the hierarchical basis the interference between levels causes this number of iterations to grow much faster from level to level, so that at the highest levels the A^TA method outperforms clearly the A method. In fact, it is often the case that we need so many iterations that the A method is not even affordable.

- Anyway, in regularization problems in which GCV or similar methods are used to find favorable smoothing parameters, one solves a series of systems which are merely translations from each other. That is, we have to solve

$$(A^T A + \nu D)d = A^T z \tag{7.2.1}$$

for different choices of ν, being D a fixed diagonal matrix. In this case one should *always* use the $A^T A$ matrix, as the overhead of assembling it is done only once for one parameter ν, with the benefit of having it available to be re-used for every choice on ν of interest.

Chapter 8

Conclusions and Outlook

In this work we have studied the use of wavelets as ansatz functions for fitting scattered data by construction of least squares approximations. This task is done in an adaptive way, avoiding the introduction of uniform grids to perform a Fast Wavelet Transform.

As initial motivation, we have derived from classical properties of wavelets potential advantages of wavelet-based methods for different aspects of the data fitting problem, and constructed specifically tailored algorithms intended to exploit them.

In order to ascertain their relevance for actual numerical applications, these algorithms have been tested on different data sets, both synthetic and stemming from data sets used in diverse scientific disciplines (as meteorology, photogrammetry, geophysics and quantum chemistry).

The results of this study, regarding the different aspects envisaged in this work can be stated as follows:

- **Performance.**

  We have presented and tested an adaptive coarse-to-fine algorithm that exploits the capacities of wavelets to structure detention performing a fast analysis of the data at each scale that allows to extract efficiently the relevant degrees of freedom of the problem. We have proven its stability and showed how the good conditioning of wavelet bases provides a better numerical performance than other multiscale bases.

- **Regularization.**

  The norm equivalence that relates the norm of a function in a wide range of function spaces with a weighted summation of the coefficients of its wavelet expansion can be used to define the classical minimization functional that balances approximation and smoothness with extended flexibility and without additional computational costs.

  Standard regularization techniques such as Generalized Cross Validation can be significantly enhanced by the use of the wavelets, whose multiscale nature

allows for a direct access to the different resolution levels. In fact, the heuristics provided by this multiscale description of the data allows to go beyond the usual regularization terms constructed on norms in Sobolev spaces, and we have shown how this approach proves itself useful to locate (and eliminate) frequency localized artifacts.

- **Robust fitting.**

  The capacity of wavelets to measure local accumulations of energy, widely exploited to detect singularities, has been used to derive an outlier finding procedure. This procedure relays on the data fitting algorithm, inheriting its good performance properties and providing in consequence an efficient schema.

- **Application on PDEs.**

  The data fitting algorithm provides a compact and easy to implement way to approximate the composition of a function with a linear combination of wavelets. This approach provides a straightaway method to cope with the nonlinear term in the Wavelet-Galerkin formulation of the Burgers' equation. This problem benefits strongly from a wavelet formulation, because of the development of singularities and the possibility of simultaneously in frequency and space located viscosity.

The general conclusion to be extracted from these results is that we can consider wavelets as a valuable tool for data analysis, beyond their traditional use in processing of gridded data by means of the Fast Wavelet Transform.

The positive results attained in this work encourage us to further research especially in the following areas of interest:

- **Characterization of the algorithm.**

  Although extensive numerical evidence of the performance of different aspects of the method has been provided, a deeper characterization of the spectral behavior of the spectrum of the coefficient matrices in presence of very irregular point distributions is still desirable.

  Further, an extension of the stability results for bivariate data and higher approximation orders is also possible.

- **Application in electron density computations**

  Electronic density computations are an active field of research in Quantum Chemistry and Solid State Physics.

  The physics of the problem leads itself quite naturally to a multiscale, adaptive formulation, as the electronic density has high variability in the neighborhood of atomic cores and quite stable values in the bulk of the elementary cell. Thus, a high resolution is only necessary in small areas of the domain.

The classical (pseudopotential)-solution of modeling separately core neighborhoods and the space between the core locations needs intensive experimental characterization of fitting parameters and a large amount of expertise. For this reason, *ab initio* computations have been introduced, which try to minimize the need of experimentally fitted parameters, [125].

In order to made these numerically practicable, diverse adaptive settings have been introduced, including wavelets: since the first experimental works in the early 90's ([30]) and later works that develop and consolidate the model on simple problems ([4],[114]), the technique has been extended and now recent works report successful application to real problems outperforming classical formulations [56],[67].

Precisely, a main obstacle for the implementation of the wavelet formulation has been the treatment of the nonlinear term. This requires a continuous switching between wavelet and real space coordinates. The traditional tool for this task is the Fast Wavelet Transformation, which is not well suited to efficient transforms on lacunary sets, as the representation of functions is compact in wavelet coordinates, but conserves its full complexity in real space coordinates (that is, in the scaling function coefficients), and consequently, this procedure cannot benefit from the adaptivity of the wavelet formulation.

We intend to circumvent this problem by working entirely in wavelet coordinates, using the LS-evaluation of the nonlinear term as explained above. In this way, the lacunarity of the set is no more a problem and its treatment does not require any special techniques. Not only a simpler formulation is envisaged. A higher flexibility can be also attained, as special families of wavelets can be constructed in order to adapt to the operator, not the possibility of performing adaptive transformations.

Appendix A

Documentation

A.1 Preface

In the last years we have been researching the use of wavelets in the scattered data fitting framework. Being a non explored field of research, (for which no tailored software was previously available), very different aspects had to be intensively explored. That led to a large number of numerical experiments of extremely variable design. This clearly causes a stark need for flexible computational tools.

On the other side, as the method is intended to work with large amounts of data, the velocity of execution is also an important issue.

In order to suit these two requirements, we have written the code `fit3` .

1. **Flexibility**

   The code has the format of a Matlab script. It can be called as a normal Matlab command, and provides the user with an easy-to-use output of Matlab variables which can be fatherly used according to current interests, from the Matlab workspace.

2. **Efficiency**

   To compensate the performance weaknesses of Matlab, the kernel of the computations is performed in C++, with specially tailored data structures.

   The communication between Matlab and C++ is made via MEX files, so that all the interchange of values occurs in rapid-access memory areas.

A.2 Adaptive Least Squares Fitting with Wavelets

We have proposed an adaptive method of least squares data fitting based on certain wavelets that works on a coarse–to–fine basis, which we recall briefly now, along with some properties of the wavelets we employ here.

Consider the set $X = \{x_i\}_{i=1,\ldots,N}$ consisting of irregularly spaced and pairwise disjoint points $x_i \in \Omega := [0,1]^n$, $n \in \{1,2\}$, denoting by $z_i \in I\!R$ for each i the corresponding data assembled in the set Z. The problem of *scattered data fitting* can be formulated as finding a function $f : \Omega \to I\!R$ that approximates the cloud of points (X, Z) in a least squares sense, that is, f minimizes the functional

$$J(f) := \sum_{i=1}^{N} (z_i - f(x_i))^2 . \tag{A.2.1}$$

Specifically, we want to construct an expansion of f of the form

$$f(x) = \sum_{\lambda \in \Lambda} d_\lambda \, \psi_\lambda(x), \quad x \in \Omega. \tag{A.2.2}$$

Here the set $\{\psi_\lambda\}_{\lambda\in\Lambda}$ consists of tensor products of certain boundary adapted B–Spline–(pre)wavelets, shortly called wavelets in the remainder of this paper, and Λ is an appropriately determined lacunary set of indices which results from an adaptive coarse–to–fine procedure which will further be explained below.

The indices $\lambda \in \Lambda$ will typically be of the form $\lambda = (j, \mathbf{k}, \mathbf{e})$, where $j =: |\lambda|$ denotes the *level of resolution* or *refinement scale*, $\mathbf{k}$ is a spatial *location*, and $\mathbf{e} \in \{0,1\}^2$ distinguishes further types of wavelets in the bivariate case which are induced by tensor products, see e.g. [46]. The infinite set of all possible indices will be denoted by $I\!I$. In view of the finite domain, there is a coarsest level j_0, so that $f(x)$ in (A.2.2) can be split into a *scaling function term* an a *wavelet term*:

$$f(x) = \sum_{\lambda\in\Lambda; j=j_0; \mathbf{e}=(0,0)} d_\lambda \, \psi_\lambda(x) \; + \sum_{\lambda\in\Lambda; j\geq j_0, e\neq(0,0)} d_\lambda \, \psi_\lambda(x). \tag{A.2.3}$$

The basis elements with multiindex $\mathbf{e} = (0,0)$ occur only on level j_0 and are called "scaling functions". In our implementation, they are tensor product of the B-splines constructed on the knot succession

$$2^{-j_0}(\overbrace{0,\ldots,0}^{m+1 \text{ times}} , 1, \ldots, 2^{j_0} - 1, \overbrace{2^{j_0},\ldots,2^{j_0}}^{m+1 \text{ times}}). \tag{A.2.4}$$

Here, m stays for the smoothness order of the constructed wavelet basis.

The basis functions of the second term, which add details of higher frequencies, are constructed as linear combinations of B-Splines.

Specifically, we work here with the wavelets $\{\psi_\lambda : \lambda \in I\!I\}$ described in [136] which have the following properties. Each ψ_λ is the tensor product of a certain linear combination of linear B–splines. This is very advantageous computationally since one can work with piecewise polynomials. In particular, the wavelets are *compactly supported* and satisfy for each $\lambda \in I\!I$ the relation $\operatorname{diam}(\operatorname{supp}\psi_\lambda) \sim 2^{-|\lambda|}$ where $a \sim b$ means that a can be estimated from above and below by a constant multiple

of b independent of all parameters on which a or b may depend. The collection $\{\psi_\lambda : \lambda \in I\!I\}$ constitutes a *Riesz basis* for $L_2(\Omega)$ and, moreover, one has *norm equivalences* for functions in Sobolev spaces $H^\alpha = H^\alpha(\Omega)$ (or even more general in Besov spaces [58]) in the range $\alpha \in [0, 3/2)$ of the form

$$\| \sum_{\lambda=(j,\mathbf{k},\mathbf{e})\in I\!I} d_\lambda \, \psi_\lambda \|^2_{H^\alpha(\Omega)} \;\sim\; \sum_{j\geq j_0} 2^{2\alpha j} \sum_{\mathbf{k},\mathbf{e}} |d_{j,\mathbf{k},\mathbf{e}}|^2. \tag{A.2.5}$$

The property of characterizing such smoothness spaces together with their compact support suggests wavelets as a powerful analysis tool for many purposes, see e.g. [46]. In addition, the wavelets we employ here are *semi–orthogonal* with respect to $L_2(\Omega)$, i.e., for $|\lambda| \neq |\mu|$ we always have $\int_\Omega \psi_\lambda(x)\, \psi_\mu(x)\, dx = 0$.

Returning to the least squares fitting problem (A.2.1), the adaptivity to the data is performed in the construction of the index set $\Lambda \subset I\!I$. In our implementation, we start with the coarse level $j_0 = 1$ and take here the set Λ_{j_0} of indices of all scaling functions and wavelets on this level. An initial fitting function $f^{j_0}(x) := \sum_{\lambda\in\Lambda_{j_0}} d^{j_0}_\lambda \, \psi_\lambda(x)$ is constructed on this set by minimizing $J(f^{j_0})$ or, equivalently, solving the normal equations

$$A^T_{\Lambda_{j_0}} A_{\Lambda_{j_0}} d^{j_0} = A^T_{\Lambda_{j_0}} z, \tag{A.2.6}$$

Here the *observation matrix* $A_{\Lambda_{j_0}}$ has entries

$$(A_{\Lambda_{j_0}})_{i,\lambda} := \psi_\lambda(x_i),\ i = 1, \ldots, N,\ \lambda \in \Lambda_{j_0}, \tag{A.2.7}$$

and z and d^{j_0} are vectors comprising the right hand side data $\{z_i\}_{i=1,\ldots,N}$ and the expansion coefficients $\{d^{j_0}_\lambda\}_{\lambda\in\Lambda_{j_0}}$. Note that the superindex j_0 identifies the maximal resolution of the whole configuration, not the scale of individual wavelets, which is contained inside of the multi-index λ.

In view of the norm equivalence (A.2.5) for $\alpha = 0$ and the locality of the ψ_λ, the absolute value of a coefficient $d^{j_0}_\lambda$ is a measure of the spatial variability of f^{j_0} on $\operatorname{supp}\psi_\lambda$: a large value of $d^{j_0}_\lambda$ is understood to be an indicator that further resolution in this area of the domain might be required. Similarly, in order to keep control over irrelevant coefficients, if $d^{j_0}_\lambda$ is below a certain threshold, this coefficient is discarded from the approximation and the index set is modified accordingly.

This motivates to construct a refined index set Λ_{j_0+1} by including those children of the wavelets indexed by Λ_{j_0} whose coefficients are above some prescribed thresholding value *and* in whose support there are more than a fixed number of data points. Note that this strategy generates an approximation on a *tree* as index structure.

The procedure is repeated until at some dyadic highest resolution level J all the computed coefficients are smaller than the thresholding value, or none of the children whose supports shrinks with each refinement step contains enough points on their support. If one of these condition is fulfilled, the algorithm stops growing

the tree. Note that as the data set is finite, the algorithm finishes in finitely many steps, and the level J is solely determined by the data.

In the following, we will extend this strategy to include also a regularizing term in the least squares functional (A.2.1) to enforce a smooth approximation.

Multiscale data fitting which may or may not include a smoothing term has been discussed also in the following references. On structured grids, in [76] a coarse–to–fine strategy has been presented with hierarchical splines which is suited for gridded, parameterized data. In [111] scattered functional data is approximated by multilevel B–Splines.

A.3 Program features

Basically, the program offers the possibility of creating in the Matlab workspace a observation matrix with relation to a set of wavelets dynamically adapted to a data distribution, jointly with an ease-to-use indexing system, so that the user can easily integrate these objects in his own computations.

A.3.1 Call of the Program

The way between a unstructured set of points and its representation as a vector of wavelet coefficients is done in four steps:

1. Creation of an appropriately formated data file.
2. Set up of parameters.
3. Creation and use of an adaptive tree structure.

Creation of a file.

Row (x, y, z) data has to be stored in an ASCII file with the right format. This format needs the following elements:

1. An arbitrary number of lines with comments.
2. A line with at least six stars to delimit the comment area
3. The number of points to be read
4. Three columns with the x, y and z coordinates of the data.

Parameter Selection.

The type of wavelet is fixed by the selection of the working directory. Currently five options are possible.

1. Linear Wavelets with isotropic refinement.
2. Linear Wavelets with anisotropic refinement.
3. Linear Hierarchical Basis with isotropic refinement.
4. Linear Hierarchical Basis with anisotropic refinement.
5. Quadratic Wavelets with isotropic refinement.
6. Cubic Wavelets with isotropic refinement.

Once in the directory of interest, the rest of relevant parameters are fixed editing the ASCII file `driver.txt`. This edition can be done by hand or with the applications `update`, `updateite`, `updatethres`. They are called from the Matlab shell with the following syntax:

1. Input data file.

   `update('input_file',<`*file_name*`>)`

2. Maximal number of levels.

   `update('max_number_levels',<`*parameter_value*`>)`

3. Type of thresholding.

   `update('threshold_d',<`*parameter_value* `> )` ¡*parameter_value*¿ can have the following values:

   0 The thresholding is done retaining wavelets whose coefficient is above the corresponding threshold.

   1 The thresholding is done refining wavelets in whose support the l_2 error of the data w.r.t. the created function is above the corresponding threshold.

4. Thresholding value at each level.

   `updatethres('max_number_levels',<`*parameter_vector*`>)` The i-th position of the vector defined by the user sets the thresholding value for level the i. Default value is 0.

5. Number of iterations at each level.

   `updateite(<`*parameter_vector*`>)` The i-th position of the vector defined by the user sets the number of iterations that the LSQR algorithm will perform at level i. Default value is 20.

6. Tolerance. `update('tol',<`*parameter_value* `> )`

   Sets the tolerance for the LSQR algorithm. Default value is 0.

7. q parameter.

   `update('q',<`*parameter_value* `> )`

A.3.2 Creation and Use of the Tree Structure.

Once `driver.txt` contains the desired values, one just calls `fit3` from the Matlab shell.

This produces a Matlab structure called `t` (from tree) whose fields code the tree structure Λ. The order of this set $\Lambda = \{\lambda_1 \ldots, \lambda_{\#\Lambda}\}$, where a general λ is equivalent to a 4-tuple (j, k_x, k_y, e) is lexicographical with order j, e, k_x, k_y.

- `t.vj` is a vector containing the j indices of the wavelets included in the constructed set Λ.

- `t.vkx` is a vector containing the k_x indices of the wavelets included in the constructed set Λ.

- `t.vky` is a vector containing the k_y indices of the wavelets included in the constructed set Λ.

- `t.ve` is a vector containing the e indices of the wavelets included in the constructed set Λ.

That means that the indices (j, k_x, k_y, e) corresponding to the $i - th$ multiindex λ_i in Λ are represented and available to the Matlab workspace by the array - `[t.vj(i),t.vkx(i),t.vky(i),t.ve(i)]`.

In addition to the tree structure, two additional fields are provided to allow the user to make further computations from the Matlab workspace.

- `t.a` is a sparse matrix containing the computed observation matrix. Rows are ordered following the order of the data points in their original file and columns are ordered according to the indexation of the other fields of `t` .

  `t.sol` is a vector containing the coefficient values of the solution as computed by `fit3` at the top level with the number of iterations and tolerance prescribed by `driver.txt`.

A.4 Implementation Details

The bulk of the program is the creation on line of a structure that characterizes and administrates the lacunary structure of the connectivity of the elements in Λ. The representation of the structure as the user gets it on the Matlab workspace once the minimization has been operated, explained in the last section, differs notably from the one used by the computing kernel, where the principal functionality of the structure is no more user-friendliness but suitableness to efficiently reproduce the evolution of Λ.

This part of the program is performed in C++. We describe now in pseudocode the structures and procedures implemented in this area of the program, and how they correspond to mathematical objects and operations.

A.4.1 Class Hierarchy

The `tree` class

Λ is represented with a structure `tree`. We will denote by `tree`(Λ) the representation of a particular tree of wavelet-indices Λ. The `tree` structure comprises fundamentally a list of pointers to `level` structures.

The `level` class

This structure quite naturally represents a dyadic level of wavelets included in Λ. that is

$$\texttt{tree.level[j]} \leftrightarrow \Lambda_{[j]} := \{\lambda : \lambda \in \Lambda, |\lambda| = j\}. \tag{A.4.1}$$

Remark: we use the notation $\Lambda_{j'}$ for the *whole* configuration of a tree when it has grown up to level j', and reserve the notation $\Lambda_{[j']}$ for the members of Λ whose scale location index is j', as explicited in (A.4.1). A `level` contains a list of pointers to the `node` structures that represent the wavelets indices included at the level $\Lambda_{[j]}$:

$$\texttt{tree.level[j].node[l]} \leftrightarrow \lambda_l,\ \lambda_l \in \Lambda_{[j]},\ l = 1, \ldots, \#\Lambda_{[j]}. \tag{A.4.2}$$

The order of the elements in $\Lambda_{[j]}$ is lexicographical. Note that $\Lambda_{[j]}$ can be lacunary if the minimization algorithm causes but the `node` list in the corresponding `level` is not.

The `node` class

Each element of the basis present in Λ is represented by an element of the class `node` . Each instance of this class is provided with fields serving three different purposes:

1. **Wavelet identification:** Wavelet indices: `j,kx,ky,e`.

2. **Data storage:** Dynamically allocated array containing primarily the index of the points found to lye in the support of the wavelet.

3. **Tree connectivity:** Pointers to those children determined to belong to Λ.

A.4.2 Creation of the Tree

Horizontal Thresholding

Accommodating unstructured data into a structured set of basis functions constitutes an important problem with regard to the implementation efficiency, as the set of data points that lyes on each wavelet in the tree has to be determined simultaneously to the set of wavelets that actually belong to the tree.

In our implementation this simultaneity is simulated by a bootstrapping construction, so that the two aspects of the problem are treated sequentially at each level.

We distinguish between the first level and all the subsequent ones.

Creation of the first level

- **Binning**

  In this first part, a loop on the data points creates the nodes of wavelets that overlap the data and stores locally the indices of the corresponding data points.

- **Closure of `level`**

  The binning creates correspondences between points and *all* the wavelets at the basis level $\Lambda_{[j_0]}$. Part of this information is useless as not all these wavelets are to be included in Λ, as far as both vertical and horizontal thresholding may exclude some (or all) of them.

  After the whole data has been explored, the q criterion is used to rule not densely populated wavelets out of `tree`.

The flow of this program is done following the Algorithm 1.

Extending to further levels At subsequent steps, the binning step of the creation can be done locally, only on the points lying at each of the wavelets included in the last current level of the tree, according to Algorithm 2.

For the implementation of this scheme one needs a definition of the index set Λ_λ of the children of a given wavelet ψ_λ. A valid definition has to include the two following properties:

$$\psi_{\lambda'} \in \Lambda_\lambda \Rightarrow \text{supp } \psi_{\lambda'} \subset \psi_\lambda \tag{A.4.3}$$

and

$$\forall \psi_{\lambda'} \in \Lambda_j, \exists | \lambda \in \Lambda_{[j+1]} \text{ so that } \psi_{\lambda'} \in \Lambda_\lambda. \tag{A.4.4}$$

In the section (A.4.3) we specify how this is done in the two different refinement schemes we work with.

Algorithm 1 Creation of basic level

Create a `level` instance for Λ_{j_0}
{Global Binning}
for $i = 1, \ldots, N$ **do**
  Determine $\Lambda_{x_i}^{j_0} := \{\lambda : x_i \in \text{supp}\psi_\lambda, , |\lambda| = j_0\}$
  for $\forall \lambda \in \Lambda_{x_i}$ **do**
    if $\lambda \notin \Lambda$ **then**
      create a `node` instance `node`(λ) for ψ_λ.
    end if
    store i in the data area of `node`(λ)
  end for
end for
{Closure of the level}
for $\lambda \in \Lambda$ **do**
  if #indices stored in `node`$(\lambda) > q$ **then**
    insert `node`(λ) in `tree.level[`j_0`]`
  end if
end for

Vertical thresholding

There are two types of vertical thresholding policies to be operated at each level, chosen by the user as explained in section A.3.1. Both of them need the construction of an approximation $f^j(x) = \sum_{\lambda \in \Lambda_j} d_\lambda^j \psi_\lambda(x)$ to the data points with the maximum available resolution level.

Approximation on level j. The construction of this temporary reconstruction is performed by an implementation of the LSQR method, whose tolerance and maximal number of iterations are fed into the procedures as explained in the previous section.

The basis of this method are the matrix-vector multiplications involving the observation matrix A and its transpose.

As `tree` (Λ) contains exactly the sparsity pattern of A, these operations are constructed as procedures of the `tree` class and can be performed with no further overhead.

Once the numerical values of the coefficients $\{d_\lambda^j\}_{\lambda \in \Lambda}$ are available, the chosen thresholding policy can be implemented. The simplest one is the thresholding according to size of the wavelet coefficients, which is just as shrinking of the last level of the `tree` .

Algorithm 2 Level expansion

Create a `level` instance for $\Lambda_{[j+1]}$ {Local Binning}
for $\forall$ `node`(λ) in `level`$[j]$ **do**
  Create a `node` instance for each chil of ψ_λ
  for $i \in$ data area of `node`(λ) **do**
    for $\lambda' \in \Lambda_\lambda$, $x_i \in \text{supp}\psi_{\lambda'}$ **do**
      store i in data area of `node`(λ')
    end for
  end for
  {Closure of the level}
  if #indices stored in `node`$(\lambda') < q$ **then**
    insert `node`(λ') in `tree.level`$[j+1]$
  end if
end for

Algorithm 3 Vertical Thresholding of level j, first policy

Given ϵ_j
for $\forall$`node`(λ) in `level`$[j]$ **do**
  if $|d_\lambda| \leq \epsilon_j$ **then**
    eliminate `tree.level[j].node`(λ)
  end if
end for

Algorithm 4 Level expansion with thresholding

Given ϵ_j
Compute $\{d_\lambda^j\}$ and $\{f^j(x_i)\}_{i=1,\dots,N}$
Create a `level` instance for $\Lambda_{[j+1]}$ {Local Binning}
for $\forall$ `node`(λ) in `level`$[j]$ **do**
  Create a `node` instance for each children $\psi_{\lambda'}$ of ψ_λ
  Initialize `error` counter on each $\lambda' \in \Lambda_\lambda$
  for $i \in$ data area of `node` $.(\lambda)$ **do**
    for $\forall \lambda' \in \Lambda_\lambda$, $x_i \in \text{supp}\psi_{\lambda'}$ **do**
      store i in data area of `node`(λ')
      `node` (λ')`.error`$+= (f^j(x_i) - z_i)^2$
    end for
  end for
  {Insertion of the children $\lambda' \in \Lambda'_\lambda$ in level}
  if #indices stored in `node`$(\lambda') < q$ **then**
    if `node`(λ')`.error` $\geq \epsilon_j$ **then**
      insert `node`(λ') in `tree.level`$[j+1]$
    end if
  end if
end for

In this case, `node` instances already inserted in a `level` are eliminated. This involves the deallocation of the memory of the `node` and the restructuring of the pointer list of the `level` .

The thresholding according to the l^2 approximation error is performed simultaneously with the horizontal thresholding, following the scheme presented in Algorithm 4. This policy avoids the elimination of `node` instances already inserted in the last `level` of the `tree` .

A.4.3 Implementation of Refinement Schemes

There are two main refinement schemes, corresponding to the two standard ways to create a bivariate basis with tensor operations on elements of a basis.

Anisotropic refinement

The supports of the elements of the bivariate basis with anisotropic refinement are as represented in Figure A.4.1.

The refinement Schema for this basis is as follows:

- nodes with $j_x \neq j_y$ refine in two children, see Figure A.4.3.
- nodes with $j_x = j_y$ refine in three groups of children.
    1. Four isotropic children, like in the left of Figure A.4.2.
    2. Two anisotropic children, elongated in the vertical direction like in the central plot of Figure A.4.2.
    3. Two anisotropic children, elongated in the horizontal direction like in the right plot of Figure A.4.2.

The arising structure is not a -lacunary- quad-tree.

Isotropic Refinement

The nodes representing each type of wavelet are refined independently. The arising structure is a vector with three lacunary quad-trees, one for each type of wavelets.

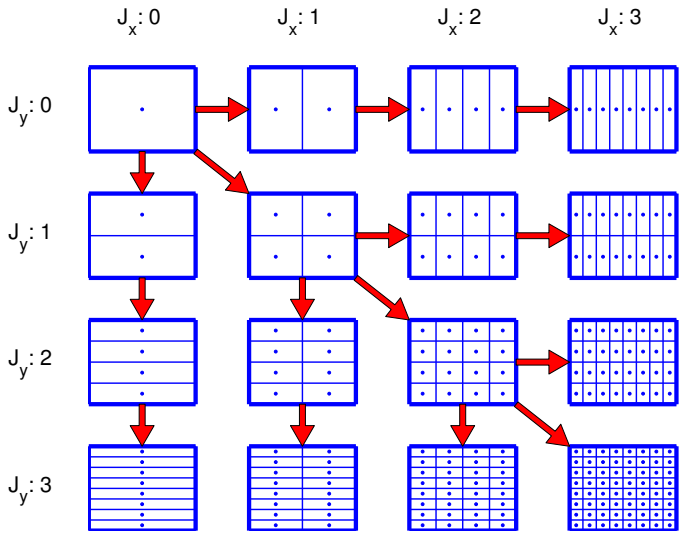
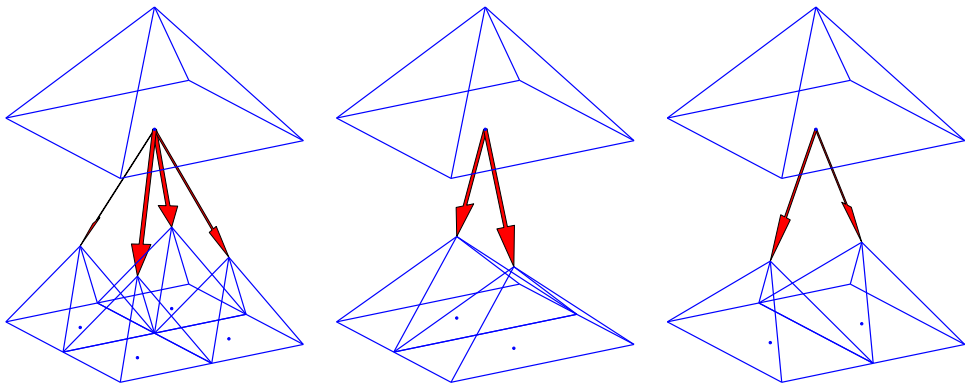
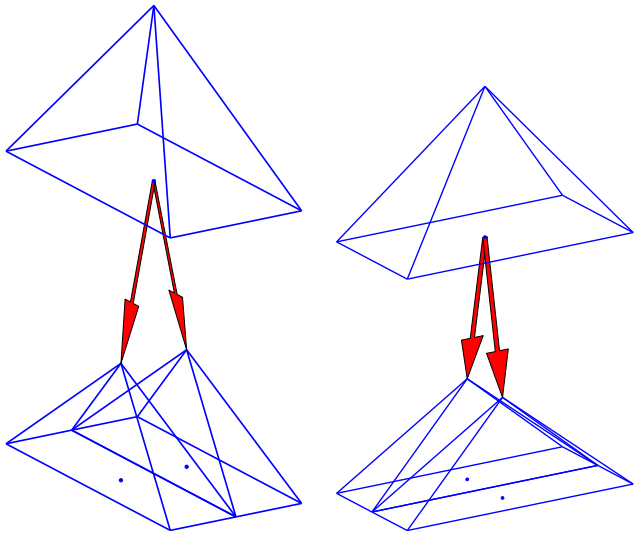


Figure A.4.1: Supports of the functions of an anisotropic linear hierarchical basis. The points are placed on the function's pinnacle. The arrows represent the refinement schema as the program operates it.



Isotropic. anisotropic along x anisotropic along y

Figure A.4.2: Refinements of an isotropic node.



Refinement for $j_x > j_y$ nodes.

Refinement for $j_y > j_x$ nodes.

Figure A.4.3: Refinements of different anisotropic nodes

Bibliography

[1] A. Averbuch, G. Beylkin, R. Coifman, M. Israeli, *Multiscale inversion of elliptic operators*, in Signal and Image Representation in Combined Spaces, J. Zeevi, R. Coifman (eds.), Academic Press, Boston, 1995, pp. 1–16.

[2] J. Ahlberg, E. Nilson, J. Walsh, *The Theory of Splines and Their Applications*. Academic Press, New York 1967.

[3] A. Arge, A. Kunoth, *An efficient ADI-solver for scattered data problems with global smoothing*, J. Comput. Phys. 139, 1998, pp. 343–358.

[4] T. A. Arias, *Multiresolution analysis of electronic structure: semicardinal and orthogonal wavelet bases*, Reviews of Modern Physics 71(1), 1999, pp. 267–311.

[5] A. Barinka, *Fast Evaluation Tools for Adaptive Wavelet Schemes*, PhD Thesis, RWTH Aachen, 2004, in preparation.

[6] A. Barinka, T. Barsch, P.Charton, A. Cohen, S. Dahlke, W. Dahmen, K. Urban, *Adaptive wavelet schemes for elliptic problems - Implementation and numerical experiments*, SIAM J. Sci. Comp. 23(3), 2001, pp. 910–939.

[7] A. Barinka, W. Dahmen, R. Schneider, *Fast Computation of Adaptive Wavelet Expansions*, in preparation, 2004.

[8] S. Bertoluzza, *A-posteriori error estimates for wavelet Galerkin methods*, Appl Math. Lett. 8, 1995, pp. 1–6.

[9] G. Beylkin, J. M. Keiser, *An adaptive pseudo-wavelet approach for solving nonlinear partial differential equations*, in Multiscale Wavelet Methods for PDEs, W. Dahmen, A. J. Kurdila, P. Oswald (eds.), Academic Press, 1997, pp. 137–197.

[10] Ch. Bernard, *Wavelets and ill-posed Problems: Optic Flow Estimation and Scattered Data Interpolation*, PhD Thesis, École Polytechnique, Paris, 1999.

[11] K. Bittner, K. Urban, *Adaptive wavelet methods using semi–orthogonal spline wavelets: Sparse evaluation of nonlinear functions*, University of Ulm, Preprint, 2004.

[12] G. P. Bonneau, *Multiresolution analysis on irregular surface meshes*, IEEE Transactions on Visualization and Computer Graphics 4(4), 1998, pp. 365–378.

[13] D. Braess, *Finite Elements: Theory, Fast Solvers and Applications in Solid Mechanics*, Cambridge University Press, Cambridge, 1997.

[14] J. H. Bramble, A. Cohen, W. Dahmen, *Multiscale Problems and Methods in Numerical Simulations : Lectures given at the C.I.M.E. Summer School held in Martina Franca, Italy, September 9-15, 2001*, C. Canuto (ed.), Springer, New York, 2003.

[15] A. G. Bruce, D. L. Donoho, H. Gao, R. D. Martin, *Denoising and robust nonlinear wavelet analysis*, IEEE Proceedings SPIE: Wavelet Applications, Orlando, 2242, 1994, pp. 325-336.

[16] M. D. Buhmann, *Radial basis functions*, Acta Numerica 9, 2000, pp. 1–38.

[17] J. M. Burgers, *The Nonlinear Diffusion Equation. Asymptotic Solutions and Statistical Problems*, D. Reidel Publishing, Dordrecht, 1974.

[18] P. L. Butzer, R. L. Stens, *Linear prediction by samples from the past*, in Advanced Topics in Shannon Sampling and Interpolation Theory, R. J. Marks II (ed.), Springer-Verlag, New York, 1993, pp. 157–183.

[19] M. Calhoun–Lopez, *Numerical Solutions of Hyperbolic Conservation Laws: Incorporating Multi–resolution Viscosity Methods into the Finite Element Framework*, Ph.D. Thesis, Iowa State University, 2003.

[20] M. Calhoun–Lopez, M. D. Gunzburger, *A finite element, multi–resolution viscosity method for hyperbolic conservation laws*, Preprint, 2003.

[21] C. Canuto, I. Cravero, *Wavelet-based adaptive methods for advection-diffusion problems*, Math. Mod. Meths. Appl. Sci. 7, 1997, pp. 265–289.

[22] C. Canuto, A. Tabacco, K. Urban, *The wavelet element method, part I: Construction and analysis*, Appl. Comp. Harm. Anal. 6, 1999, pp. 1–52.

[23] J. M. Carnicer, W. Dahmen, J. M. Peña, *Local decomposition of refinable spaces*, Appl. Comp. Harm. Anal. 8, 2000, pp. 127–153.

[24] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, T. R. Evans, *Reconstruction and representation of 3D objects with radial basis functions*, in Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM Press, New York, 2001, pp. 67–76.

[25] P. J. Mc Carthy, *Direct analytic model of the L-curve for Tikhonov regularization parameter selection.* Inverse Problems 19, 2003, pp. 643–663.

[26] D. Castaño, A. Kunoth, *Adaptive fitting of scattered data by spline–wavelets*, in: "Curves and Surfaces", L. L. Schumaker et. al. (eds.), Vanderbilt University Press, Nashville, 2003, pp. 65–78.

[27] D. Castaño, A. Kunoth, *Multilevel regularization of wavelet based fitting of scattered data - Some experiments*, 2004, to appear in Numer. Algor.

[28] L. Y. Chen, J. T. Chen, H. K. Hong, C. H. Chen, *Application of Cesáro mean and the L-curve for the deconvolution problem*, Soil Dynamics and Earthquake Engineering 14, 1995, pp. 361–373.

[29] G. Chiavassa, M. Guichaoua, J. Liandrat *Two adaptive wavelet algorithms for non-linear parabolic partial differential equations*, Computers and Fluids 31, 2002, pp. 467–480.

[30] K. Cho, T. A. Arias, J. D. Joannopoulos, P. K. Lam, *Wavelets in electronic structure calculations*, Physical Review Letters 71, 1993, pp. 1808–1811.

[31] C. K. Chui, E. G. Quak, *Wavelets on a bounded interval*, in Numerical Methods of Approximation Theory, Volume 9, D. Braess and L. L. Schumaker (eds.), International Series of Numerical Mathematics, Volume 105, Birkhäuser, Basel, 1992, pp. 53-75.

[32] P. G. Ciarlet, *Basic Error estimates for the finite element method*, in Handbook of Numerical Analysis, II, P. G. Ciarlet and J.-L. Lions (eds.), Elsevier, Amsterdam, 1991.

[33] A. Cohen, *Wavelet Methods in Numerical Analysis*, in The Handbook of Numerical Analysis, VII, P. G. Ciarlet and J.-L. Lions (eds.), Elsevier, Amsterdam, 2000.

[34] A. Cohen, W. Dahmen, R. DeVore, *Adaptive wavelet methods for elliptic operator equations – convergence rates*, Math. Comp. 70, 2001, pp. 27–75.

[35] A. Cohen, W. Dahmen, R. DeVore, *Adaptive wavelet schemes for nonlinear variational problems*, SIAM J. Numer. Anal. 5(41), 2003, pp. 1785–1823.

[36] A. Cohen, W. Dahmen, R. DeVore, *Sparse evaluation of compositions of functions using multiscale expansions*, SIAM J. Math. Anal. 35, 2003, pp. 279–303.

[37] A. Cohen, W. Dahmen, R. Devore, *Adaptive wavelet techniques in numerical simulation*, in Encyclopedia of Computational Mathematics, Stein, de Borst, Hughes (eds.), John Wiley & Sons, Chichester, 2004.

[38] A. Cohen, I. Daubechies, J. C. Feauveau, *Biorthogonal bases of compactly supported wavelets*, Comm. Pure Appl. Math. 45(5), 1992, pp. 485–500.

[39] A. Cohen, R. Masson, *Wavelet adaptive methods for second-order elliptic problems, boundary conditions and domain decomposition*, Numer. Math. 8, 1997, pp. 21–47.

[40] A. Cohen, R. Masson, *Wavelet methods for second-order elliptic problems, preconditioning and adaptivity*, SIAM J. Sci. Comput. 21, 1999, pp. 1006–1026.

[41] P. Craven, G. Wahba, *Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation*, Numer. Math. 31, 1979, pp. 377–403.

[42] S. Dahlke, *Besov regularity for elliptic boundary value problems on polygonal domains*, Appl. Math. Lett. 12, 1999, pp. 31–36.

[43] S. Dahlke, W. Dahmen, R. DeVore, *Nonlinear approximation and adaptive techniques for solving elliptic operator equations*, in: Multiscale Wavelet Methods for PDEs, W. Dahmen, A. Kurdila, P. Oswald (eds.), Academic Press, San Diego, 1997, 237-283.

[44] S. Dahlke, R. DeVore, *Besov regularity for elliptic boundary value problems*, Comm. Partial Differential Equations 22, 1997, pp. 1–16.

[45] W. Dahmen, *Some remarks on multiscale transformations, stability and biorthogonality*, in Wavelets, Images and Surface Fitting, P. J. Laurent, A. Le Mehaute, L. L. Schumaker (eds.), AK Peters, Wellesley, 1994, pp. 157–188.

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

[47] W. Dahmen, A. Kunoth, *Multilevel preconditioning*, Numer. Math. 63, 1992, pp. 315–344.

[48] W. Dahmen, A. Kunoth, K. Urban, *Biorthogonal spline-wavelets on the interval – Stability and moment conditions*, Appl. Comp. Harm. Anal. 6, 1999, pp. 132–196.

[49] W. Dahmen, S. Müller, T. Schlinkmann, *Multigrid and multiscale decompositions*, in Large-Scale Scientific Computations of Engineering and Environmental Problems, M. Griebel, O.P. Iliev, S.D. Margenov, P.S. Vassilevski (eds.), Notes on Numerical Fluid Mechanics 62, Vieweg, 1998, pp. 18–41.

[50] W. Dahmen, R. Schneider, *Composite wavelet bases for operator equations*, Math. Comp 68, 1999, pp. 184–230.

[51] W. Dahmen, R. Schneider, Y. Xu, *Nonlinear functionals of wavelet expansions — Adaptive reconstruction and fast evaluation*, Numer. Math. 86, 2000, pp. 49–101.

[52] W. Dahmen, R. Stevenson, *Element-by-element construction of wavelets – stability and moment conditions*, SIAM J. Numer. Anal. 37, 1999, pp. 319–325.

[53] I. Daubechies, *Orthonormal bases of compactly supported wavelets*, Comm. Pur. Appl. Math. 41(7), 1998, pp. 909–996.

[54] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992.

[55] I. Daubechies, I. Guskov, P. Schröder, W. Sweldens, *Wavelets on irregular point sets*, Phil. Trans. R. Soc. Lon. A, 357(1760), 1999, pp. 2397–2413.

[56] I. P. Daykov, T. D. Engeness, T. A. Arias, *Robust ab initio calculation of condensed matter: transparent convergence through semicardinal multiresolution analysis*, Physical Review Letters, 90(21), 2003, pp. 216402.

[57] C. de Boor, *Bicubic spline interpolation*, J. Math. Phys. 41, 1962, pp. 212–218.

[58] R. A. DeVore, *Nonlinear approximation*, Acta Numerica 7, 1998, pp. 51–150.

[59] R. A. DeVore, B. Jawerth, V. Popov, *Compression of wavelet decompositions*, Amer. J. Math. 114, 1992, pp. 737–785.

[60] R. A. DeVore, B. Lucier, *Wavelets*, Acta Numerica 1, 1991, pp. 1–56.

[61] R. A. DeVore, V. A. Popov, *Interpolation of Besov spaces*, Trans. Amer. Math. Soc. 305, 1988, pp. 397–414.

[62] P. Dierckx, *Curve and Surface Fitting with Splines*, Clarendon Press, Oxford, 1993.

[63] P. L. Dragotti, M. Vetterli, *Wavelet footprints: theory, algorithms and applications*, IEEE Trans. on Signal Processing, 51(5), 2003, pp. 1306–1323.

[64] S. Durand, J. Froment, *Artifact free signal denoising with wavelets*, in Proc. of ICASSP'01, 6, 2001.

[65] S. Durand, J. Froment, *Reconstruction of wavelet coefficients using total variation minimization*, SIAM Journal of Scientific Computing 24(5), 2003, pp. 1754–1767.

[66] N. Dyn, M. S. Floater, A. Iske, *Adaptive thinning for bivariate scattered data*, Journal of Computational and Applied Mathematics 145(2), 2002, pp. 505–517.

[67] T. D. Engeness, T. A. Arias, *Multiresolution analysis for efficient, high precision all-electron density-functional calculations*, Physical Review B, 65, 2002, article number 165106.

[68] H. W. Engl, M. Hanke, A. Neubauer, *Regularization of Inverse Problems*, Mathematics and Its Applications, 375, Kluwer Academics Publishers, 2000.

[69] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, Boston, third edition, 1993.

[70] H. G. Feichtinger, K. Gröchenig, *Theory and practice of irregular sampling*, in Wavelets: Mathematics and Applications, J. Benedetto and M. Frazier (eds.), CRC Press, Boca Raton, 1993, pp. 305–363.

[71] H. G. Feichtinger, K. Gröchenig, T. Strohmer, *Efficient numerical methods in non-uniform sampling theory*, Numer. Math. 69, 1995, pp. 423–440.

[72] A. Finkelstein, D. H. Salesin, *Multiresolution curves*, in *Proceedings of SIGGRAPH '94*, ACM, New York, 1994, pp. 261–268.

[73] M. S. Floater, A. Iske, *Thinning, inserting and swapping scattered data*, in Surface Fitting and Multiresolution Methods, A. Le Mehaute, C. Rabut and L. L. Schumaker (eds.), Vanderbilt University Press, Nashville, 1996, pp. 139–144.

[74] M. S. Floater, A. Iske, *Multistep scattered data interpolation using compactly supported radial basis functions*, Journal of Computational and Applied Mathematics 73(5), 1996, pp. 65–78.

[75] M. S. Floater, A. Iske, *Thinning algorithms for scattered data interpolation*, BIT 38(4), 1998, pp. 705–720.

[76] D. R. Forsey, R. H. Bartels, *Surface fitting with hierarchical splines*, ACM Transactions on Graphics (TOG), 14(2), 1995, pp. 134–161.

[77] R. Franke, *Scattered data interpolation: Tests of some methods*, Math. Comput. 38, 157, 1982, pp. 181-200.

[78] Data available through `http://www.math.nps.navy.mil/~rfranke/README`.

[79] A. Gelb, E. Tadmor, *Enhanced spectral viscosity approximations for conservation laws*, Appl. Num. Math. 33, 2000, pp. 3–21.

[80] T. Gerstner, *Adaptive hierarchical methods for landscape visualization and analysis*, in Process Modeling and Landform Evolution, S. Hergarten, H.J. Neugebauer (eds.), Lecture Notes in Earth Sciences 78, Springer, Berlin, 1999, pp. 75–92.

[81] G. H. Golub, C. F. van Loan, *Matrix Computations*, John Hopkins, Baltimore, 1996.

[82] S. J. Gortler, M. Cohen, *Hierarchical and variational geometric modeling with wavelets*, in Proc. 1995 Symposium on Interactive 3D Graphics, pp. 35–ff.

[83] G. Greiner, K. Hormann, *Interpolating and approximating scattered 3D-data with hierarchical tensor product splines*, in Surface Fitting and Multiresolution Methods, A. Le Mehaute and C. Rabut and L. L. Schumaker (eds.), Vanderbilt University Press, Nashville, 1996, pp. 163–172.

[84] M. H. Gross, O. G. Staadt, R. Gatti, *Efficient triangular surface approximations using wavelets and quadtree data structure*, IEEE Trans. Vis. Comput. Graph. 2(2), 1996, pp. 130–143.

[85] U.S. Geological Survey's GTOPO30 Digital Elevation Model. Website location: `http://edcdaac.usgs.gov/gtopo30/gtopo30.asp`.

[86] A. Haar, *Zur Theorie der orthogonalen Funktionen-System*, Mathematische Annalen 69, 1910, pp. 331–371.

[87] M. Hanke, *Conjugate Gradient Type Methods for Ill-Posed Problems*, Longman Scientific & Technical, Harlow, 1995.

[88] M. Hanke, P. Ch. Hansen, *Regularization methods for large-scale problems*, Surveys on Mathematics for Industry, 3, 1993, pp. 253–315.

[89] P. Ch. Hansen, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Review, 34(4), 1992, pp. 561–580.

[90] P. Ch. Hansen, *Regularization Tools: A Matlab package for analysis and solution of discrete ill-posed problems*, Numerical Algorithms, 6, 1994, pp. 1–35.

[91] P. Ch. Hansen, *The L-curve and its use in the numerical treatment of inverse problems*, in Computational Inverse Problems in Electrocardiology, P. Johnston (ed.) Computational Inverse Problems in Electrocardiology, WIT Press, Southampton, 2001, pp. 119–142.

[92] W. Härdle, G. Kerkyacharian, D. Picard, A. Tsybakov, *Wavelets, Approximation and Statistical Applications*, Springer, New York, 1998.

[93] J. G. Hayes, J. Halliday, *The least squares fitting of cubic spline surfaces to general data sets*, J. Inst. Math. Appl. 14, 1974, pp. 89–106.

[94] M. Hegland, S. Roberts, I. Altas, *Finite element thin plate splines for surface fitting,* in Computational Techniques and Applications: CTAC97, B. J. Noye, M. D. Teubner and A. W. Gill (eds.), World Scientific, 1997, pp. 289–296.

[95] F. Heurtaux, F. Planchon, M. V. Wickerhauser, *Scale decomposition in Burgers' equation*, Wavelets: Mathematics and Applications, John Benedetto, Michael Frazier (eds.), CRC Press, 1994, pp. 505-523.

[96] M. Holschneider, *Wavelets: An Analysis Tool*. Oxford Mathematical Monographs, Clarendon Press, Oxford, 1995.

[97] P. J. Huber, *Robust Statistics*, Wiley, New York, 1981.

[98] M. F. Hutchinson, F. R. de Hoog, *Smoothing noisy data with spline functions*, Numer. Math. 47, 1985, pp. 99–106.

[99] A. Iske, J. Levesley, *Multilevel scattered data approximation by adaptive domain decomposition*, Technical Report No. 2002/17, University of Leicester, 2002.

[100] S. Jaffard, *Exposants de Hölder en points donnés et coefficients d'ondelettes,* C. R. Acad. Sci. Paris, 308, 1989, pp. 79–81.

[101] S. Jaffard, *Pointwise smoothness, two-microlocalization and wavelet coefficients*, Publicaciones Matemáticas 1, 1991, pp. 316–328.

[102] S. Jaffard, *Wavelet methods for fast resolution of elliptic equations*, SIAM J. Numer. Anal. 29, 1992, pp. 965–986.

[103] S. Jaffard, Y. Meyer, *Wavelet Methods for Pointwise Regularity and Local Oscillations of Functions*, 123, American Mathematical Society, 1996.

[104] P. Joly, Y. Maday, V. Perrier, *A dynamical adaptive concept based on wavelet packet best bases: application to convection diffusion partial differential equations*, in Multiscale Wavelet Methods for PDEs, W. Dahmen, A. Kurdila, P. Oswald (eds.), Academic Press, 1997.

[105] J. Kamm, *Singular Value Decomposition-based Methods for Signal and Image Restoration*, PhD Thesis, Southern Methodist University, Dallas, 1998.

[106] L. Kaufman, A. Neumaier, *PET regularization by envelope guided conjugate gradients*, IEEE Trans. Medical Imaging 15, 1996, pp. 385–389.

[107] A. Kovac, *Wavelet Thresholding for Unequally Spaced Data*, PhD Thesis, University of Bristol, 1998.

[108] R. Kraft, *Adaptive and linearly independent multilevel B-splines*, in Surface Fitting and Multiresolution Methods, A. Le Mehaute, C. Rabut and L. L. Schumaker (eds.), Vanderbilt Univ. Press, 1997, pp. 209–218.

[109] A. Kunoth, *Adaptive wavelet schemes for an elliptic control problem with Dirichlet boundary control*, Preprint, Universität Bonn, 2003, to appear in Numer. Algor.

[110] P. Lancaster, K. Šalkauskas, *Curve and Surface Fitting: An Introduction*, Academic Press, London, 1986.

[111] S. Lee, G. Wolberg, S. Y. Shin, *Scattered data interpolation with multilevel B-splines,* IEEE Trans. Visualization and Computer Graphics 3(3), 1997, pp. 228–244.

[112] J. Liandrat, Ph. Tchamitchian,*Resolution of the 1D regularized Burgers equation using a spatial wavelet approximation: Algorithm and Numerical results*, ICASE Report No 90-83, 1990.

[113] J. Lin, Z. Huang, *Adaptive scattered data interpolation with multilevel nonuniform B-splines*, in Eurographics 99 Short Papers and Demos, M. A. Alberti, G. Gallo and I. Jelinek (eds.), Eurographics Association, 1999, pp. 220–222.

[114] R. A. Lippert, T. A. Arias, A. Edelman, *Multiscale computation with interpolating wavelets*, J. Comput. Phys. 140, 1998, pp. 278–310.

[115] Y. Maday, V. Perrier, J. C. Perrier, *Adaptivitée dynamique sur bases d'ondelettes pour l'approximation d'equations aux derivées partielles*, CRAS Paris I, 1990, pp. 405–410.

[116] S. G. Mallat, *A theory for multiresolution signal decomposition: The wavelet representation*, IEEE Transactions on Pattern Analysis and Machine Intelligence 11(7), 1989, pp. 674- 693.

[117] S. G. Mallat, *A Wavelet Tour on Signal Processing*, Academic Press, San Diego, 1997.

[118] S. Mallat, W. L. Hwang, *Singularity detection and processing with wavelets*, IEEE Trans. Inform. Theory 38(2), 1992, pp. 617–643.

[119] Y. Meyer, *Ondelettes et opérateurs 1-3: Ondelettes*, Hermann, Paris, 1990.

[120] Y. Meyer, *Ondelettes sur l'intervalle*, Revista Matemática Iberoamericana 7(2), 1991, pp. 115–143.

[121] S. Müller, *Adaptive Multiscale Schemes for Conservation Laws*, Springer, Berlin, 2003.

[122] G. M. Nielson, R. Franke, *Scattered data interpolation and applications: A tutorial and survey*, in Geometric Modeling: Methods and Their Applications, H. Hagen and D. Roller (eds.), Springer, Berlin, 1990, pp. 131–160.

[123] P. Oswald, *Multilevel Finite Element Approximation: Theory and Applications*, Teubner Skripten zur Numerik, Teubner, Stuttgart, 1994.

[124] C. C. Paige, *LSQR. An algorithm for sparse linear equations and sparse least squares*, ACM Trans, Math. Software 8, 1982, pp. 43–71.

[125] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, J. D. Joannopoulos, *Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients*, Reviews of Modern Physics, 64(4), 1992, pp. 1045–1097.

[126] R. Pfeifle, H. Seidel, *Fitting triangular B-splines to functional scattered data*, Computer Graphics Forum 15(1), 1996, pp. 15–23.

[127] M. J. D. Powell, *The theory of radial basis function approximation in 1990*, in Advances in Numerical Analysis II: Wavelets, Subdivision and Radial Basis Functions, W. A. Light (ed.), Clarendon Press, Oxford, 1992, pp. 105–210.

[128] The Puerto Rico Tsunami Warning and Mitigation Program. Data obtainable at `http://poseidon.uprm.edu`.

[129] M. Rauth, *Gridding of Geophysical Potentials from Noisy Scattered Data*, PhD Thesis, University of Vienna, May 1998.

[130] V. Scheib, J. Haber, M. C. Lin, H. P. Seidel, *Efficient fitting and rendering of large scattered data sets using subdivision surfaces*, Computer Graphics Forum 21(3), 2002, pp. 353–362.

[131] K. Schittkowski, *Data fitting in partial differential algebraic equations: some academic and industrial applications*, Journal of Computational and Applied Mathematics, 163(1), 2004, pp. 29–54.

[132] I. J. Schoenberg, A. Whitney, *On Pólya frequency functions III*, Trans. Am. Math. 74, 1953, pp. 246–259.

[133] L. L. Schumaker, *Fitting surfaces to scattered data*, in Approximation Theory II, G. G. Lorentz, C. K. Chui and L. L. Schumaker (eds.), Academic Press, New York, 1976, pp. 203–268.

[134] H. Schwetlick, T. Schütze, *Least squares approximation by splines with free knots*, BIT, 35(3), 1995, pp. 361–384.

[135] G. W. Stewart, *Matrix Algorithms: Basic Decompositions*, Society for Industrial and Applied Mathematics, Philadelphia, 1998.

[136] E. J. Stollnitz, T. D. DeRose, D. H. Salesin, *Wavelets for Computer Graphics: Theory and Applications*, Morgan Kaufmann, San Francisco, 1996.

[137] W. Sweldens, *The lifting scheme: A custom-design construction of biorthogonal wavelets*, Appl. Comput. Harmon. Anal 3(2), 1996, pp. 186-200.

[138] E. Tadmor, *Convergence of spectral methods for nonlinear conservation laws*, SIAM J. Numer. Anal. 26, 1989, pp. 30–44.

[139] H. Triebel, *Interpolation Theory, Function Spaces and Differential Operators*, North Holland, Amsterdam, 1978.

[140] F. Utreras, *Optimal smoothing of noisy data using spline functions*, SIAM J. Sci. Statist. Comput. 2, 1981, pp. 349–362.

[141] C. R. Vogel, *Computational Methods for Inverse Problems*, Frontiers in Applied Mathematics, SIAM, Philadelphia, 2002.

[142] G. Wahba, *Splines Models for Observational Data*, Series in Applied Mathematics, 59, SIAM, Philadelphia, 1990.

[143] G. Wahba, *(Smoothing) Splines in Nonparametric Regression*, in Encyclopedia of Environmetrics, A. El-Shaarawi and W. Piegorsch (eds.), Wiley, 4, 2001, pp. 2099–2112.

[144] G. Wahba, D. R. Johnson, F. Gao, J. Gong, *Adaptive tuning of numerical weather prediction models: randomized GCV in three- and four-dimensional data assimilation*, Monthly Weather Review, 123, 1995, pp. 3358–3369.

[145] D. C. Wan, G. W. Wei, *The study of quasi wavelets based numerical method applied to Burgers' equations*, Appl. Math. Mech.-Engl. 21, 2000, pp. 1099-1110.

[146] H. Weimer, J. Warren, *Fast approximating triangulation of large scattered datasets*, Advances in Engineering Software, 30(6), 1999, pp. 389–400.

[147] A. Williams, K. Burrage, *Surface fitting using GCV smoothing splines on supercomputers*, in Proceedings of the 1995 ACM/IEEE Conference on Supercomputing, ACM Press, San Diego USA, 1995, article number 11.

[148] Z. Wu, R. Schaback, *Local error estimates for radial basis function interpolation of scattered data*, IMA J. Numer. Anal. 13, 1993, pp. 13–27.

[149] H. Yserentant, *On the multi-level splitting of finite element spaces*, Numer. Math. 49, 1986, pp. 379–412.

[150] O. C. Zienkiewicz, D. W. Kelley, S. R. Gago, I. Babuška, *Hierarchical finite element approaches, error estimates and adaptive refinement*, in The Mathematics of Finite Elements and Applications IV, Academic Press, New York, 1982, pp. 313–346.

Acknowledgments

First of all, I wish to thank my supervisor Angela Kunoth, who made this thesis possible in different ways: offering me the position as a research assistant, giving me freedom, advice and encouragement to pursuit different research lines, and occasionaly filling my reserves of optimism.

I am especially grateful to Arestis Antoniadis for his visit to our group, taking the effort of reading my work and my software sources and for his comments about the meaning of regularization and robust regression. I am also indebted to Max Gunzburger for motivating and encouraging my work on the Burgers' equation.

Sincere thanks are extendend to Carsten Burstedde for being always available for discussions about wavelets, numerics, condition numbers, philosophy of science, and beer. The same holds for Mario Mommer.

Thanks also to David Fernández and Jorge Iñiguez for his magistral lessons about *ab initio* methods for electronic density computations.

Finally, I want to express my gratitude towards my coworkers at the Institute for Applied Mathematics in the University of Bonn for the friendly atmosphere.

Eskerrik asko

Aunque me tiren el puente
y tambien la pasarela
me verás cruzar el Ebro
en un barquito de vela.

"Si me quieres escribir", canción popular, 1938

La vieja canción republicana y sus circunstancias históricas nos recuerdan algo en el fondo muy cotidiano: que uno puede poner todo su ahínco en algo (y encima tener razón), y sin embargo fracasar miserablemente. Y que aún así, tras una de éstas toca intentar tirar por algún lado.

Y esto se hace más fácil si sabes que te van a echar una mano. Este es el momento de agradecer esa ayuda que me ha sido brindada por Montse, Loli, César y Mari Luz, con su experiencia y su ejemplo; por los X-Patata (besteak beste Convenioren baikortasuna, Ethilosen txiste txarrak, Viperisen jakinduria, Esthetosen ideia sutsuak, Armoniaren eta Balsamoren etengabeko sostengua, Trebolen ikuspuntu aberasgarriak eta Asilaboren asilabotasuna direla medio) y los f97*, en especial la anárquica, isoespínica e imprescindible intersección de ambos grupos; por la solidaridad internacional de las jitxas madrileñas, de Lavinia, de Marian...*

Y por Regine, sin la cual esta tesis se habría escrito, pero no habría valido la pena.