Multivariate Polynomial Interpolation and the Lifting Scheme with an Application to Scattered Data Approximation

Dominik Stahl

Vom Fachbereich Mathematik der Technischen Universität Kaiserslautern zur Verleihung des akademischen Grades Doktor der Naturwissenschaften (Doctor rerum naturalium, Dr. rer. nat.) genehmigte Dissertation D 386 Datum der Disputation: 17.04.2013

Gutachter:	Prof. Dr. Tobias Damm	(TU Kaiserslautern)
	Prof. Dr. Daniel Kreßner	(EPF Lausanne)

Abstract

This thesis deals with generalized inverses, multivariate polynomial interpolation and approximation of scattered data. Moreover, it covers the lifting scheme, which basically links the aforementioned topics. For instance, determining filters for the lifting scheme is connected to multivariate polynomial interpolation. More precisely, sets of interpolation sites are required that can be interpolated by a unique polynomial of a certain degree. In this thesis a new class of such sets is introduced and elements from this class are used to construct new and computationally more efficient filters for the lifting scheme.

Furthermore, a method to approximate multidimensional scattered data is introduced which is based on the lifting scheme. A major task in this method is to solve an ordinary linear least squares problem which possesses a special structure. Exploiting this structure yields better approximations and therefore this particular least squares problem is analyzed in detail. This leads to a characterization of special generalized inverses with partially prescribed image spaces.

Contents

Preface

This thesis consists of four chapters where each chapter is devoted to one of the following subjects: *Generalized inverses*, *multivariate polynomial interpolation*, *the lifting scheme* and *approximation of scattered data*. Each chapter starts with an abstract that contains a sketch of my own contributions and an outline of the chapter. This is followed by an introductory section, where the corresponding subject is motivated.

Here, in this preface, I present a more personal view on this thesis, where I give an overview of my main contributions and comment at some points why I approached exactly those problems which are now topics in this thesis. Furthermore, it is revealed how the different subjects in this thesis are connected to each other.

Substantial parts of this thesis were motivated by the paper [KS00], which deals with the lifting scheme in arbitrary dimensions d. The lifting scheme is a filter bank structure that can be used to efficiently perform the discrete wavelet transform. Moreover, the structure of the lifting scheme gives an idea on how to construct new filters, and so new wavelets, by solving a linear system of equations

$$\sum_{k\in K} p_{-k} k^{\alpha} = \tau^{\alpha} \quad \text{for} \quad |\alpha| \leq N \;, \tag{1}$$

with $\alpha := (\alpha_1, \dots, \alpha_d)$, $|\alpha| := \alpha_1 + \cdots + \alpha_d$, $\tau \in \mathbb{R}^d$ and $N \in \mathbb{Z}_+$. The crucial point is to find a finite set $K \subset \mathbb{Z}^d$ of points $k \in K$ such that the system (1) possesses a unique solution $(p_{-k} \in \mathbb{R} : k \in K)$. As we will learn in this thesis this is the case when the set K is correct, i.e., for any $f : \mathbb{R}^d \to \mathbb{R}$ there exists a unique polynomial $q \in \Pi_N^d$ in d variables and with total degree at most N, which interpolates the values $(f(k) : k \in K)$. However, in [KS00] it is written:

> "It is not clear a priori how many interpolation points and which geometric configurations are needed to uniquely solve the interpolation problem for a space of polynomials up to a certain degree." (2)

Implementing the system of equations (1) makes it necessary to provide all multi-indices α with $|\alpha| \leq N$, which results in a set $\Gamma_{N,d} = \{\alpha \in \mathbb{Z}_+^d : |\alpha| \leq N\}$ of $\dim \Pi_N^d$ points. So without putting much thoughts on how to choose K a natural choice is to try $K = \Gamma_{N,d}$. Interestingly for any tried choice of N and d, the matrix representing the system of equations (1) was non-singular, meaning that the set $K = \Gamma_{N,d}$ is correct. My "advantage" at this point was my little knowledge on multivariate polynomial interpolation, because for $d = 2$ this configuration was already discussed in 1903 by Biermann. But having the quote (2) in mind I started thinking why this special configuration of points always led to a uniquely solvable system.

At the point when realizing why for arbitrary N and d the set $K = \Gamma_{N,d}$ is correct, it also became clear from the proof that one can even characterize a whole class of correct sets based on $\Gamma_{N,d}$. Moreover, this characterization can be used as a concrete recipe to construct correct sets. Then fortunate circumstances brought me together with Carl de Boor, who noticed that my characterization is more general than known characterizations and that for $d = 2$ my characterization is a special case of Radon's recipe. We started collaboration at which end we came up with the article [SB11], in which we present a new characterization of a class of correct sets, where we show that this characterization covers all sets that are constructible by the recursive application of Radon's recipe. These and more results on correct sets are written down in *Chapter II*.

In *Chapter III* I discuss the lifting scheme and its connection to wavelets. Moreover, I construct a new family of filters for the lifting scheme by using the new recipe on correct sets from Chapter II together with equation (1). When determining new filters for the lifting scheme one has to be aware that the coefficients $(p_{-k} : k \in K)$ of the determined filter implicitly define a Riesz basis via some refinement equation. I verified this property for all my filters in Chapter III. In comparison, not all filters which are derived in [KS00] meet this property. Additionally my filters have less filter coefficients which reflects in saving computing time when applying the lifting scheme. Furthermore, I provide a result on the geometrical configuration of K such that many filter coefficients vanish to zero. Exploiting this result yields an extension of the one-dimensional Deslaurier–Dubuc filters to the two-dimensional quincunx case.

Besides working on the aforementioned topics I developed a method to approximate scattered data by using the lifting scheme. This approach is introduced in *Chapter IV*. A major task in my approach is to solve a least squares problem

$$\min_x \|Ax - b\|_2^2 \,, \tag{3}$$

where the matrix A has the property $AE_n = E_m$, with $E_n := [1, \dots, 1]^T \in \mathbb{R}^n$. Like in similar methods, e.g., [FE98] and [NM99], I used the minimal norm solution to (3) in the beginning. But using the minimal norm solution yields bad effects near the boundary of the corresponding approximant, even for constant valued scattered data. I prove that constant valued scattered data is approximated exactly if and only if the solution to the least squares problem $\min_x \|Ax - E_m\|_2^2$ equals $x = E_n$, which is unlikely the case for the minimal norm solution within my approach. However, it is a well-known fact that all $\{1,3\}$-inverses $A^{(1,3)}$ of A give a solution to the least squares problem by $x = A^{(1,3)}b$, where the Moore–Penrose inverse $A^\dagger$ is the unique $\{1,3\}$-inverse with $\|A^\dagger b\|_2$ minimal. Hence, I started thinking of a $\{1,3\}$-inverse $A^\natural$ with the property $A^\natural E_m = E_n$. Therefore, I considered the more general case

$$A^\natural(AY) = Y \quad \text{with} \quad A \in \mathbb{C}^{m\times n}, \;\; Y \in \mathbb{C}^{n\times \ell} \;\text{ and }\; \operatorname{rank} AY = \ell \tag{4}$$

and characterized all $\{1,3\}$-inverses $A^\natural$ satisfying this condition (4). Moreover, I determined conditions such that matrices out of this subset of all $\{1,3\}$-inverses coincide with the Moore–Penrose inverse on certain subspaces of $\mathbb{C}^m$. This leads to two natural choices of $\{1,3\}$-inverses $A^\natural$ satisfying (4). All this is discussed in detail in *Chapter I* on generalized inverses where also focus is put on computational aspects and the connection to the Tikhonov regularization. Some of the results presented in there can also be found in the article [DS12], which I published together with my supervisor Tobias Damm.

The results from Chapter I are then exploited in Chapter IV on scattered data approximation, where I also compare different solutions and regularizations to the least squares problem (3) and the corresponding approximations. More precisely, I reveal why the minimal norm solution is not the method of choice and that the newly derived $\{1,3\}$-inverses from Chapter I and a regularization to (3) that restricts the roughness of the solution deliver much better results. I also compare this new method to existing methods and show that it yields similar or even better results. At the end of Chapter IV, I introduce an idea to significantly speed up the convergence of a conjugate gradient method applied on a Tikhonov regularization to (3).

Structure of the thesis

In summary, in Chapter I on generalized inverses I discuss the solution of a least squares problem with additional constraints. This is exploited in Chapter IV in the approach of scattered data approximation. Chapter II deals on how to construct correct sets for multivariate polynomial interpolation. These results are then used in Chapter III to construct a new family of Neville filters. Besides that I explain in Chapter III the lifting scheme, which is used in the method to approximate scattered data in Chapter IV. Hence:

$$\begin{tikzcd}
\text{Chapter I} \arrow[rrr, bend right] & \text{Chapter II} \arrow[r, bend left] & \text{Chapter III} \arrow[r, bend left] & \text{Chapter IV}
\end{tikzcd}$$

Notation

The j-th labeled equation in Section i within Chapter K is tagged by $(i.j)$. In Chapter $L \neq K$ this equation is referenced by $(K.i.j)$. Same, but without brackets, applies for subsections, figures, tables and the class of mathematical "environments" like definitions and theorems.

Mathematical notation used in this thesis can be found in the *glossary of notation* at page 101, where the page number behind each entry corresponds to its first appearance in this thesis.

Acknowledgments

Foremost I would like to thank my supervisor Tobias Damm for his advice and support during the last years.

Special thanks to Carl de Boor, for giving me the opportunity to collaborate with him and also for answering several questions concerning multivariate polynomial interpolation.

I also want to thank Daniel Kreßner for inviting me to EPF Lausanne to present my work and additionally for his readiness to review this PhD-thesis.

For financial support and hosting me the last years I am obliged to the whole department of *System Prognosis and Control* of the Fraunhofer ITWM in Kaiserslautern.

For (mathematical) discussions, proofreading, or any other support I want to express my gratitude to Urs Becker, Jan Hauth, Annette Krengel, Lisa Ollinger, Stefan Steidel and Ulrich Thiel.

Chapter I

Generalized Inverses

We start this chapter by discussing the *Moore–Penrose inverse* and its properties in Section 1. In Section 2, we deal with the *least squares problem* and how *generalized inverses* and the *Tikhonov regularization* are connected to it. Until this point these are all well-known results, this changes in Section 3 where we construct generalized inverses with partially prescribed image spaces to solve least squares problems. This ends in a characterization of a special subset of $\{1,3\}$-inverses. Furthermore, we discuss properties of particular $\{1,3\}$-inverses from this set, where we also deal with several computational aspects, see Section 3.2. In Section 4 we present Tikhonov regularizations that have prescribed solutions, where it is also shown that in the limit of every Tikhonov regularization its solution can also be obtained by a $\{1,2,3\}$-inverse. In the last section of this chapter we discuss the new results related to a special case.

1 Introduction

According to [BIG03] a generalized inverse of a matrix A should exist also for non-singular and even for non-square matrices while preserving properties that the usual inverse possesses. It should further coincide with the usual inverse when A is non-singular. The best-known generalized inverse that meets these conditions is the Moore–Penrose inverse. It was first discovered as *reciprocal* of a matrix by E. H. Moore in 1920, see [Moo20] and [MB35]. The problem of Moore's work was its use of a very complicated notation, which made the work accessible only to a few readers. Hence, the work of Moore was barely noticed. This made an independent rediscovery necessary. That was done by R. Penrose in [Pen55] by introducing the *generalized inverse* of a matrix. Shortly later R. Rado noticed in [Rad56] that Moore's *reciprocal* and Penrose's *generalized inverse* coincide. Hence, this generalized inverse is nowadays referred to as the Moore–Penrose inverse.

Since then the Moore–Penrose inverse found application in various fields. For instance in providing the minimal norm solution to the ordinary linear least squares problem, as we will learn in Section 2.

1.1 The Moore–Penrose inverse

The Moore–Penrose inverse $A^\dagger$ of a matrix A is the unique matrix X that satisfies the four Penrose equations

$$\begin{aligned} &1) \quad AXA = A \\ &2) \quad XAX = X \\ &3) \quad AX = (AX)^* \\ &4) \quad XA = (XA)^* \ , \end{aligned} \tag{1.1}$$

see [Pen55]. In the next lemma we state some properties of the Moore–Penrose inverse $A^\dagger$ which are immediate consequences of the above equations (1.1).

Lemma 1.1 [Pen55] *Let $A \in \mathbb{C}^{m\times n}$ and $\tau \in \mathbb{C}$, where*

$$\tau^\dagger := \begin{cases} \tau^{-1} & \textit{if} \quad \tau \neq 0 \\ 0 & \textit{if} \quad \tau = 0 \end{cases} .$$

The following hold

(a) $(A^\dagger)^\dagger = A$ *;*

(b) $(A^*)^\dagger = (A^\dagger)^*$ *;*

(c) if A is non-singular $A^\dagger = A^{-1}$ *;*

(d) $(\lambda A)^\dagger = \lambda^\dagger A^\dagger$ *;*

(e) $(A^*A)^\dagger = A^\dagger(A^\dagger)^*$ *;*

(f) $A^\dagger AA^* = A^* = A^*AA^\dagger$ *;*

(g) if U and V are unitary $(UAV)^\dagger = V^*A^\dagger U^*$ *.*

Readers that are interested in Moore's original work on the reciprocal are referred to [BI02], where Ben-Israel presents a restatement of Moore's work using modern and more simple notation.

1.2 $\{i, j, \ldots, k\}$-inverses

In this thesis we also consider generalized inverses which only satisfy some of the four Penrose equations (1.1). Consider for instance that X only satisfies Penrose equation 1) and 3), then we call such a matrix the $\{1, 3\}$-inverse of A. More generally:

Definition 1.2 [BIG03, page 40] *For any $A \in \mathbb{C}^{m\times n}$, let $A\{i, j, \ldots, k\}$ denote the set of matrices $X \in \mathbb{C}^{n\times m}$ which satisfy equations $i), j), \ldots k)$ from among (1.1). A matrix $X \in A\{i, j, \ldots, k\}$ is called an $\{i, j, \ldots, k\}$-inverse of A, and also denoted by $A^{(i,j,\ldots,k)}$.*

In this notation the Moore–Penrose inverse $A^\dagger = A^{(1,2,3,4)}$ is the $\{1, 2, 3, 4\}$-inverse of A. In Section 2.1 we will learn that $\{1, 3\}$-inverses can be used to obtain a solution to the least squares problem.

1.3 The four subspaces

To every matrix $A \in \mathbb{C}^{m\times n}$ four subspaces are connected, namely

$$\begin{array}{ccc} \mathcal{R}(A^*) & \subset & \mathbb{C}^n \\ \mathcal{N}(A) & \subset & \mathbb{C}^n \end{array} \quad \text{and} \quad \begin{array}{ccc} \mathcal{R}(A) & \subset & \mathbb{C}^m \\ \mathcal{N}(A^*) & \subset & \mathbb{C}^m \end{array} .$$

Obviously, if $A \in \mathbb{C}^{n\times n}$ is non-singular $\mathcal{R}(A) = \mathcal{R}(A^*) = \mathbb{C}^n$ and $\mathcal{N}(A) = \mathcal{N}(A^*) = \{0\}$. Hence, the inverse A^{-1} just maps $\mathcal{R}(A)$ to $\mathcal{R}(A^*)$. In case $A \in \mathbb{C}^{m\times n}$ with $m \neq n$ there is no inverse. Though an inverse from $\mathcal{R}(A)$ to $\mathcal{R}(A^*)$ still exists – the restriction of the Moore–Penrose inverse to $\mathcal{R}(A)$, i.e., $A^\dagger|_{\mathcal{R}(A)} = (A|_{\mathcal{R}(A^*)})^{-1}$. This is what we show in this section.

We start with the following relation:

Theorem 1.3 [BIG03, page 12] *For any $A \in \mathbb{C}^{m\times n}$,*

$$\begin{array}{rcl} \mathcal{R}(A^*) & = & \mathcal{N}(A)^\perp \\ \mathcal{R}(A) & = & \mathcal{N}(A^*)^\perp . \end{array} \tag{1.2}$$

Proof. Recall that

$$\langle Ax, y\rangle = \langle x, A^*y\rangle \quad \text{for all} \quad x \in \mathbb{C}^n,\ y \in \mathbb{C}^m . \tag{1.3}$$

Let $x \in \mathcal{N}(A)$. Then the left hand side of equation (1.3) vanishes for all $y \in \mathbb{C}^m$. It follows then that $x \perp A^*y$ for all $y \in \mathbb{C}^m$, or, i.e., $x \perp \mathcal{R}(A)$. This proves that $\mathcal{N}(A) \subset \mathcal{R}(A^*)^\perp$.

Conversely, let $x \in \mathcal{R}(A^*)^\perp$, so that the right hand side of equation (1.3) vanishes for all $y \in \mathbb{C}^m$. This implies that $Ax \perp y$ for all $y \in \mathbb{C}^m$. Therefore $Ax = 0$. This proves that $\mathcal{R}(A^*)^\perp \subset \mathcal{N}(A)$, and completes the proof.

The proof of relation $\mathcal{R}(A) = \mathcal{N}(A^*)^\perp$ works analogously. □

Let $\mathcal{R}(X) \subset \mathbb{C}^n$ and $\mathcal{R}(Y) \subset \mathbb{C}^m$ then we denote by $\mathcal{L}(\mathcal{R}(X), \mathcal{R}(Y))$ the set of all linear transformations from $\mathcal{R}(X)$ to $\mathcal{R}(Y)$. Since $\mathcal{L}(\mathbb{C}^n, \mathbb{C}^m)$ and the space of all matrices $\mathbb{C}^{m\times n}$ are isomorph we use the same symbol A for elements in that class. So let $A \in \mathcal{L}(\mathbb{C}^n, \mathbb{C}^m)$ and $\mathcal{R}(X) \subset \mathbb{C}^n$ then we denote the restriction of A to $\mathcal{R}(X)$ by $A|_{\mathcal{R}(X)} \in \mathcal{L}(\mathcal{R}(X), \mathbb{C}^m)$, where $A|_{\mathcal{R}(X)}x = Ax$ if $x \in \mathcal{R}(X)$.

In the next lemma we prove that for every $\{1,2\}$-inverse X of A it holds that

$$X|_{\mathcal{R}(\mathcal{A})} \in \mathcal{L}(\mathcal{R}(A), \mathcal{R}(X))$$

is a bijection. In the subsequent Theorem 1.5 we show that $\mathcal{R}(A^\dagger) = \mathcal{R}(A^*)$.

Lemma 1.4 *Let $A \in \mathbb{C}^{m\times n}$ and $X \in A\{1,2\}$, then the following hold*

(a) $\mathcal{R}(X) \cap \mathcal{N}(A) = \{0\}$ and $\mathcal{R}(A) \cap \mathcal{N}(X) = \{0\}$;

(b) $A|_{\mathcal{R}(X)} \in \mathcal{L}(\mathcal{R}(X), \mathcal{R}(A))$ and $X|_{\mathcal{R}(A)} \in \mathcal{L}(\mathcal{R}(A), \mathcal{R}(X))$ are one-to-one;

(c) $\mathcal{R}(A|_{\mathcal{R}(X)}) = \mathcal{R}(A)$ and $\mathcal{R}(X|_{\mathcal{R}(A)}) = \mathcal{R}(X)$;

(d) $X|_{\mathcal{R}(A)} = (A|_{\mathcal{R}(X)})^{-1} \in \mathcal{L}(\mathcal{R}(A), \mathcal{R}(X))$.

Proof. (a) Let $x \in \mathcal{R}(X) \cap \mathcal{N}(A)$, then there exists a y such that $Xy = x$. Then

$$x = Xy = XAXy = XAx = 0 .$$

Hence $\mathcal{R}(X) \cap \mathcal{N}(A) = \{0\}$. The case $\mathcal{R}(A) \cap \mathcal{N}(X) = \{0\}$ can be proved analogously.

(b) We start with $A|_{\mathcal{R}(X)} \in \mathcal{L}\left(\mathcal{R}(X), \mathcal{R}(A)\right)$, for $X|_{\mathcal{R}(A)}$ the proof works also analogously.
Let $x, y \in \mathcal{R}(X)$. Assume $x \neq y$ with $Ax = Ay$. Hence $A(x-y) = 0$ and thus $x - y \in \mathcal{N}(A)$, which is a contradiction to (a). Hence $x = y$ and $A|_{\mathcal{R}(X)}$ is one-to-one.

(c) $Ax = AXAx = AP_{\mathcal{R}(X)}x = A|_{\mathcal{R}(X)}x$ for all $x \in \mathbb{C}^n$. Same for the second case.

(d) Since $A|_{\mathcal{R}(X)} \in \mathcal{L}\left(\mathcal{R}(X), \mathcal{R}(A)\right)$ is one-to-one and onto there exists an inverse. Let $x \in \mathcal{R}(X)$, then there is a unique y such that $y = Ax$. Due to the first Penrose equation this is equivalent to $y = AXAx = AXy$. Since $y \in \mathcal{R}(A)$ and $X|_{\mathcal{R}(A)}$ and $A|_{\mathcal{R}(X)}$ are one-to-one and onto (b and c) it holds that $x = Xy$.

□

Theorem 1.5 *For any* $A \in \mathbb{C}^{m\times n}$,

$$\begin{array}{rcl} \mathcal{R}(A^*) & = & \mathcal{R}(A^\dagger) \\ \mathcal{N}(A^*) & = & \mathcal{N}(A^\dagger)\ . \end{array} \tag{1.4}$$

Proof. Let $y \in \mathcal{R}(A^\dagger)$ and $x \in \mathcal{N}(A)$. Then

$$x^*y = x^*A^\dagger u\ ,$$

since $A^\dagger \in A\{1,2\}$ it holds that $u \in \mathcal{R}(A)$, see Lemma 1.4 (c). Hence, there exists a $v \in \mathbb{C}^m$ such that

$$x^*y = x^*A^\dagger Av\ .$$

Because of Penrose equation 3)

$$x^*y = x^*A^*(A^\dagger)^*v = 0\ .$$

Thus $\mathcal{R}(A^\dagger) \perp \mathcal{N}(A)$. Since the orthogonal complement is unique it follows from Theorem 1.3 that $\mathcal{R}(A^\dagger) = \mathcal{R}(A^*)$.

The case $\mathcal{N}(A^*) = \mathcal{N}(A^\dagger)$ works analogously. □

2 Least squares problem

Consider the system of equations $Ax = b$ with $A \in \mathbb{C}^{m\times n}$. If $b \in \mathcal{R}(A)$ the system has at least one solution and is called *consistent*. On the other hand, the system is called *inconsistent* if $b \notin \mathcal{R}(A)$. In this case only an approximate solution can be obtained. Such an approximate solution is for instance obtainable by minimizing the Euclidean norm of the residual vector $r := Ax - b$, i.e.,

$$\min_x \|Ax - b\|_2^2\ . \tag{2.1}$$

Because (2.1) means nothing else than minimizing the sum of the squares of the absolute value of the residuals ($\sum_i |r_i|^2$), this is known as *least squares problem*. A solution to (2.1) is for instance often needed in statistical problems like regression analysis.
In [Pen56] Penrose proved that a unique solution to the least squares problem (2.1) can be obtained by the Moore–Penrose inverse.

Theorem 2.1 [Pen56] *Let* $A \in \mathbb{C}^{m\times n}$ *and* $b \in \mathbb{C}^m$. *Then* $x = A^\dagger b$ *is the unique solution to the least squares problem* $\min_x \|Ax - b\|_2^2$ *which has minimal norm* $\|x\|_2$.

Proof. See Corollary 2.5. □

Though, one can obtain more general solutions to the least squares problem, by obviously loosing the property that the solution has minimal norm. These solutions are subject of the next Section 2.1.

Penrose also dealt with the more general matrix equation $AXB = D$.

Theorem 2.2 [Pen55, Pen56] *Let $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{p\times q}$, $D \in \mathbb{C}^{m\times q}$.*

(a) A necessary and sufficient condition for the matrix equation

$$AXB = D \tag{2.2}$$

to have a solution is

$$AA^\dagger DB^\dagger B = D \ , \tag{2.3}$$

in which case the general solution is

$$X = A^\dagger DB^\dagger + K - A^\dagger AKBB^\dagger$$

for arbitrary $K \in \mathbb{C}^{n\times p}$.

(b) In general, $X = A^\dagger DB^\dagger$ is the minimum norm least squares approximate solution of equation (2.2), i.e., $A^\dagger DB^\dagger$ is the unique element of least Frobenius norm in the set

$$\{Y \ : \ \|AYB - D\|_F = \min_X \|AXB - D\|_F\} \ . \tag{2.4}$$

2.1 $\{1, 3\}$-inverses and the least squares problem

In this section we show that every $\{1, 3\}$-inverse of A yields a solution to the least squares problem (2.1) by $x = A^{(1,3)}b$. The whole set $A\{1, 3\}$ can be characterized as follows.

Theorem 2.3 [BIG03, page 55] *Let $A \in \mathbb{C}^{m\times n}$ and $A^{(1,3)} \in A\{1, 3\}$ arbitrary. Then*

(a) the set $A\{1, 3\}$ consists of all solutions X of

$$AX = AA^{(1,3)} \ . \tag{2.5}$$

(b)

$$A\{1, 3\} = \{A^{(1,3)} + (I - A^{(1,3)}A)Z : Z \in \mathbb{C}^{n\times m}\} \ . \tag{2.6}$$

Proof. (a) Multiplying equation (2.5) with A to the left side shows that the first Penrose condition is satisfied if X is a solution to (2.5). Since $A^{(1,3)}$ satisfies Penrose equation 3) $AA^{(1,3)}$ is hermitian by definition and so is AX.

Vice versa, let $X \in A\{1, 3\}$ then

$$AA^{(1,3)} = AXAA^{(1,3)} = (AX)^*AA^{(1,3)} = X^*A^*(A^{(1,3)})^*A^* = X^*A^* = AX \ .$$

(b) According to [BIG03, page 52] we can replace $A^\dagger$ and $B^\dagger$ by $A^{(1,3)}$ and $B^{(1,3)}$, respectively, in Theorem 2.2 (a). Hence, a general solution to equation (2.5) is

$$X = A^{(1,3)}AA^{(1,3)} + Y - A^{(1,3)}AY \ .$$

If we set $Y = Z + A^{(1,3)}$ for an arbitrary $Z \in \mathbb{C}^{n\times m}$ we finally obtain

$$X = A^{(1,3)} + \left(I - A^{(1,3)}A\right) Z \ .$$

□

As mentioned above, a general solution to the least squares problem can be obtained by $A^{(1,3)}b$ if and only if $A^{(1,3)} \in A\{1,3\}$.

Theorem 2.4 [BIG03, page 104] *Let $A \in \mathbb{C}^{m\times n}$ and $b \in \mathbb{C}^m$. Then $\|Ax-b\|_2$ is minimal for $x = A^{(1,3)}b$ for an $A^{(1,3)} \in A\{1,3\}$. Conversely, if $X \in \mathbb{C}^{n\times m}$ has the property that, for all b, $\|Ax-b\|_2$ is minimal when $x = Xb$, then $X \in A\{1,3\}$.*

Proof. According to Section 1.3 it holds that

$$b = (P_{\mathcal{R}(A)} + P_{\mathcal{R}(A)^\perp})b \ ,$$

which is equivalent to

$$Ax - b = (Ax - P_{\mathcal{R}(A)}b) - P_{\mathcal{N}(A^*)}b \ . \tag{2.7}$$

Let Y and Z be subspaces of $\mathbb{C}^m$. Then the Pythagorean theorem states that $\|y+z\|_2^2 = \|y\|_2^2 + \|z\|_2^2$ if and only if $Y \perp Z$, for all $y \in Y$ and $z \in Z$. Hence, applying the norm to both sides of equation (2.7) we get

$$\|Ax-b\|_2^2 = \|Ax - P_{\mathcal{R}(A)}b\|_2^2 + \|P_{\mathcal{N}(A^*)}b\|_2^2 \ . \tag{2.8}$$

This is obviously minimal if and only if

$$Ax = P_{\mathcal{R}(A)}b \ . \tag{2.9}$$

The Penrose equations (1.1) imply that $AA^\dagger = P_{\mathcal{R}(A)}$ and by Theorem 2.3 it holds that $AA^\dagger = AA^{(1,3)}$ for every $A^{(1,3)} \in A\{1,3\}$. Hence,

$$x = A^{(1,3)}b \ .$$

Vice versa, if for all $b \in \mathbb{C}^m$ it holds that $x = Xb$ minimizes $\|Ax-b\|_2$, equation (2.9) implies that $AXb = P_{\mathcal{R}(A)}b$. Hence $AX = P_{\mathcal{R}(A)}$ and thus due to Theorem 2.3 $X \in A\{1,3\}$. □

Thus, for an $A^{(1,3)} \in A\{1,3\}$ the general solution to the least squares problem (2.1) reads

$$x = A^{(1,3)}b + (I - A^{(1,3)}A)y \tag{2.10}$$

for an arbitrary $y \in \mathbb{C}^n$.

Corollary 2.5 *The solution $x = A^\dagger b$ to the least squares problem (2.1) has minimal norm $\|x\|_2$.*

Proof. In equation (2.10) set $A^{(1,3)} = A^\dagger$, then

$$\|x\|_2^2 = \|A^\dagger b\|_2^2 + \|(I - A^\dagger A)y\|_2^2 \ ,$$

by Theorem 1.5 and the Pythagorean theorem. Hence $\|x\|_2$ is minimal if $y = 0$. □

2.2 Tikhonov regularization

If the matrix A from the least squares problem (2.1) is ill-conditioned or the solution to the least squares problem should fulfill certain properties, like being smooth, the Tikhonov regularization is applied. In the standard case the Tikhonov regularization reads

$$\min_x \|Ax-b\|_2^2 + \tau^2\|x\|_2^2 \ ,$$

which for $\tau \neq 0$ has the unique solution

$$x = (A^*A + \tau^2 I)^{-1} A^* b \, .$$

We even have that

$$\lim_{\tau \to 0} (A^*A + \tau I)^{-1} A^* = A^\dagger \, , \tag{2.11}$$

which was first proved in [dBC57]. So in the standard case the Tikhonov regularization just gives the standard minimal norm solution to the least squares problem for $\tau \to 0$.

Let $T \in \mathbb{C}^{k \times n}$ for $1 \leq k \leq n$, then the general case of the Tikhonov regularization reads

$$\min_x \|Ax - b\|_2^2 + \tau^2 \|Tx\|_2^2 \quad \text{with} \quad \tau > 0 \, , \tag{2.12}$$

which is equivalent to

$$\min_x \left\| \begin{bmatrix} A \\ \tau T \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2^2 \, . \tag{2.13}$$

If $\mathcal{N}(A) \cap \mathcal{N}(T) = \{0\}$ the general Tikhonov regularization possesses the unique solution

$$x_{T,\tau} := (A^*A + \tau^2 T^*T)^{-1} A^* b \, . \tag{2.14}$$

The discrete Tikhonov regularization (2.12) can be seen as a discretization of the continuous problem

$$\min_f \|Kf - g\|_{L^2}^2 + \tau^2 S(f)^2 \, ,$$

with K being a linear operator describing some model and g representing corresponding observations. To preserve for instance physical effects in the solution f, $S(f)$ is often chosen as a so-called *smoothing norm*, see, e.g., [Cul79], [Jen06] or [Han10, Chapter 8]. One choice for smoothing norms that restrict the roughness of the solution f are weighted Sobolev norms, for example in the 2-dimensional case the so-called bending energy

$$S(f) = \left(\iint \left(\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right) \mathrm{d}x \mathrm{d}y \right)^{\frac{1}{2}} \, .$$

Another common choice for $S(f)$ which restricts the roughness of the solution is

$$S(f) = \left(\iint \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right)^2 \mathrm{d}x \mathrm{d}y \right)^{\frac{1}{2}} \, ,$$

where measures connected to the Laplacian Δf are often used in image restoration and two-dimensional smoothing, see [Jen06, Chapter 5] and the references therein.

Thus, solutions $x_{T,\tau}$ in the discrete setting (2.12) that should possess a certain smoothness can for instance be obtained by choosing T as discrete Laplace operator, see, e.g., (IV.2.11).

Choosing the regularization parameter

A natural question that arises when using the Tikhonov regularization is how to choose the regularization parameter τ. There are several methods which can be used for an automated determination of a regularization parameter τ. We briefly explain one of the most popular methods, the *generalized cross-validation*, short GCV, cf. [Wah90, Chapter 4]. For a discussion and comparison of this and other methods we refer to [Han10, Chapter 5], from which we also partially borrow the following.

Let $A \in \mathbb{R}^{m\times n}$ and $\tilde{b} \in \mathbb{R}^m$. Moreover, let $b = \tilde{b} + \delta$ for some non-zero $\delta \in \mathbb{R}^m$, where $\tilde{b}$ expresses the exact data here. Furthermore, let $x_{T,\tau}$ be the solution to the corresponding problem (2.12) for a regularization matrix T with $\mathcal{N}(A) \cap \mathcal{N}(T) = \{0\}$. Then the idea is to choose τ such that $Ax_{T,\tau}$ predicts the exact data $\tilde{b}$ as well as possible. But usually $\tilde{b}$ and δ are not known. Therefore one uses the leave-out-one strategy, i.e., one leaves one b_i out and also removes the corresponding i-th row of A and then computes the Tikhonov solution to this reduced problem. We denote its solution by $x_{T,\tau}^{(i)}$. Then we can compute an estimate of b_i by

$$A(i,:)x_{T,\tau}^{(i)} .$$

The idea of cross validation is now to choose τ such that all prediction errors for all components of b are minimized, i.e.,

$$\min_{\tau} \frac{1}{m} \sum_{i=1}^{m} \left(A(i,:)x_{T,\tau}^{(i)} - b_i\right)^2$$

which can be shown to be equivalent to the numerical more efficient form

$$\min_{\tau} \frac{1}{m} \sum_{i=1}^{m} \left(\frac{A(i,:)x_{T,\tau} - b_i}{1 - H(i,i)}\right)^2 ,$$

with $H := A(A^T A + \tau^2 T^T T)^{-1} A^T$. The problem with $H(i,i)$ is that it is dependent of the ordering of the data b, so one ends up with different τ for different orderings. This is circumvented by the generalized cross-validation, where $H(i,i)$ is replaced by the average of all diagonal elements of H. Hence in the GCV-case τ is finally obtained by minimizing the function

$$G(\tau) = \frac{\|Ax_{T,\tau} - b\|_2^2}{(m - \mathrm{trace}(H))^2} ,$$

see also [GVM97]. To efficiently evaluate the function G for large matrices A, $x_{T,\tau}$ is computed iteratively, see the next paragraph for more details. Moreover, the trace of the matrix H can be estimated by the use of Hutchinson's stochastic trace estimator $\mathrm{trace}(H) \approx u^T H u$ with u being a random vector which has values 1 and -1 with probability 0.5, see [Hut90] and [GVM97]. Hence the effort for evaluating G at a point τ can basically be reduced to two Tikhonov solutions.

Efficient computation of the Tikhonov solution

In case that the matrix A is very large and sparse, the effort to obtain a solution to the system of linear equations (2.14) is quite high when using direct methods, like QR- or singular value-decomposition. In this case it is much more appropriate to use iterative methods. Since the leading matrix of the normal equation $(A+\tau T)^*(A+\tau T)x = A^* b$ is symmetric positive definite if $\mathcal{N}(A) \cap \mathcal{N}(T) = \{0\}$ one could apply directly a conjugate gradient method to the normal equation. But numerically this also is not the choice since one has to perform a matrix multiplication on two large matrices. An option, also suggested in [Han10, page 121], is the *CGLS Algorithm* (Conjugate Gradients Least Squares). It directly works on the least squares problem (2.13) and performs in every iteration step two matrix vector multiplications, one with $[A, \tau T]$ and the other with its conjugate transpose. For a detailed discussion see [Bjö96, Section 7.4].

3 $\{1,3\}$-inverses with partially prescribed image spaces

Motivated by an application of scattered data approximation, which we will present in Chapter IV, we consider the following problem:

Problem 3.1 *Let $A \in \mathbb{C}^{m\times n}$ and $Y \in \mathbb{C}^{n\times \ell}$ so that* $\operatorname{rank} AY = \ell$.
Find a matrix $A^\natural \in \mathbb{C}^{n\times m}$ such that

(a) $x = A^\natural b$ minimizes $\|Ax-b\|_2$ for all $b \in \mathbb{C}^m$,

(b) $A^\natural(AY) = Y$.

From the previous Section 2.1 we know that a matrix $A^\natural$ satisfying Problem 3.1 (a) is necessarily a $\{1,3\}$-inverse. In this section we characterize all $A^\natural \in A\{1,3\}$ that additionally fulfill condition (b), see Theorem 3.2. Out of this set of all matrices solving Problem 3.1 we present matrices that coincide with the Moore–Penrose inverse on certain subspaces of $\mathbb{C}^m$, see Theorem 3.5 where we also give two important choices for $A^\natural$. In Section 3.2 we discuss several computational aspects of these two important candidates. We (Tobias Damm and I) published most of the results in this section in [DS12], therefore several parts here largely follow the presentation in [DS12].

We start now by presenting all matrices $A^\natural$ that satisfy Problem 3.1.

Theorem 3.2 [DS12] *Let $A \in \mathbb{C}^{m\times n}$ and $Y \in \mathbb{C}^{n\times \ell}$ so that* $\operatorname{rank} AY = \ell$. *Then a matrix $A^\natural$ solves Problem 3.1 if and only if $A^\natural = A^{(1,3)}_{Y,K}$ for an arbitrary matrix $K \in \mathbb{C}^{n\times m}$, where*

$$A^{(1,3)}_{Y,K} := A^\dagger + (I - A^\dagger A)\left(Y(AY)^\dagger + K - KAY(AY)^\dagger\right) . \tag{3.1}$$

Proof. In view of equation (2.6) we have to characterize all $Z \in \mathbb{C}^{n\times m}$ such that

$$\left(A^\dagger + (I - A^\dagger A)Z\right) AY = Y , \tag{3.2}$$

where we chose $A^{(1,3)}$ as $A^\dagger$. Equation (3.2) is equivalent to

$$(I - A^\dagger A)ZAY = (I - A^\dagger A)Y . \tag{3.3}$$

For this equation, condition (2.3) of Theorem 3.2 reads

$$(I - A^\dagger A)(I - A^\dagger A)^\dagger (I - A^\dagger A)Y(AY)^\dagger AY \overset{!}{=} (I - A^\dagger A)Y ,$$

which is obviously satisfied because of the Penrose conditions and the hypothesis $\operatorname{rank} AY = \ell$. Hence equation (3.3) is consistent, and the general solution is

$$Z = (I - A^\dagger A)Y(AY)^\dagger + K - (I - A^\dagger A)KAY(AY)^\dagger \quad \text{for arbitrary} \quad K \in \mathbb{C}^{n\times m} .$$

Inserting this in (2.6) we get the form (3.1) of all solutions to Problem 3.1. □

Corollary 3.3 *As above let $Y \in \mathbb{C}^{n\times \ell}$ so that* $\operatorname{rank} AY = \ell$. *Then for all non-singular $S \in \mathbb{C}^{\ell\times \ell}$ it holds that*

$$A^{(1,3)}_{YS,K} = A^{(1,3)}_{Y,K} .$$

Proof. By (3.1) it holds that

$$A^{(1,3)}_{YS,K} = A^\dagger + (I - A^\dagger A)\left(YS(AYS)^\dagger + K - KAYS(AYS)^\dagger\right) \ . \tag{3.4}$$

Since AYS has full rank it holds that

$$\begin{aligned}(AYS)^\dagger &= \left(S^*(AY)^*AYS\right)^{-1} S^*(AY)^* \\ &= S^{-1}\left((AY)^*AY\right)^{-1}(AY)^* \\ &= S^{-1}(AY)^\dagger \ .\end{aligned}$$

Hence, S cancels out in equation (3.4) and $A^{(1,3)}_{YS,K}$ equals $A^{(1,3)}_{Y,K}$. □

Because $K \in \mathbb{C}^{n\times m}$ is arbitrary in Theorem 3.2 there is freedom in the choice of $A^{(1,3)}_{Y,K}$. In the next lemma we show how to choose K such that $\|A^{(1,3)}_{Y,K} L\|_F$ is minimal for an $L \in \mathbb{C}^{m\times k}$.

Lemma 3.4 [DS12] *Let $A \in \mathbb{C}^{m\times n}$, $Y \in \mathbb{C}^{n\times \ell}$ so that* $\operatorname{rank} AY = \ell$ *and $L \in \mathbb{C}^{m\times k}$. Then the unique matrix $K \in \mathbb{C}^{n\times m}$ of minimal Frobenius norm that minimizes the expression $\|A^{(1,3)}_{Y,K} L\|_F$ is given by*

$$K(L) := -(I - A^\dagger A)Y(AY)^\dagger L \left(L - AY(AY)^\dagger L\right)^\dagger \ .$$

In particular $K(I) = 0$.

Proof. By definition of $A^{(1,3)}_{Y,K}$ we have

$$\min_K \|A^{(1,3)}_{Y,K} L\|_F = \min_K \left\|\left(A^\dagger + (I - A^\dagger A)Y(AY)^\dagger\right) L + (I - A^\dagger A)K(I - AY(AY)^\dagger)L\right\|_F \ .$$

By Theorem 2.2(b) we know that

$$\begin{aligned}K &= -\,(I - A^\dagger A)^\dagger \left(A^\dagger + (I - A^\dagger A)Y(AY)^\dagger\right) L \left((I - AY(AY)^\dagger)L\right)^\dagger \\ &= -\,(I - A^\dagger A)Y(AY)^\dagger L \left(L - AY(AY)^\dagger L\right)^\dagger =: K(L)\end{aligned}$$

is the minimal norm solution to $\min_K \|A^{(1,3)}_{Y,K} L\|_F$.
In particular $K(I) = -(I - A^\dagger A)Y(AY)^\dagger + (I - A^\dagger A)Y(AY)^\dagger AY(AY)^\dagger = 0$. □

In the next Theorem we show that if $\mathcal{R}(L)$ is complementary to $\mathcal{R}(AY)$, then the solution $x = A^{(1,3)}_{Y,K(L)} b$ to the least squares problem $\min_x \|Ax - b\|_2^2$ coincides with the minimal norm solution $x = A^\dagger b$ for all $b \in \mathcal{R}(L)$.

Theorem 3.5 [DS12] *Let $A \in \mathbb{C}^{m\times n}$, $Y \in \mathbb{C}^{n\times \ell}$ so that* $\operatorname{rank} AY = \ell$ *and let $L \in \mathbb{C}^{m\times(m-k)}$ with $1 \le k \le \ell$ and* $\operatorname{rank} L = m - k$. *Furthermore, let $\mathcal{R}(L) \cap \mathcal{R}(AY) = \{0\}$. Then*

(a) $A^{(1,3)}_{Y,K(L)} L = A^\dagger L$.

(b) $\mathcal{R}(L) \perp \mathcal{R}(AY)$ *implies* $A^{(1,3)}_{Y,K(L)} = A^{(1,3)}_{Y,0}$.

(c) $\mathcal{R}(Y) \perp \mathcal{R}\left(A^{(1,3)}_{Y,K(L)} L\right) \Leftrightarrow \mathcal{R}(L) \subset \mathcal{N}(Y^* A^\dagger)$.

Proof. (a) By definition of $A^{(1,3)}_{Y,K}$ and $K(L)$, we have

$$A^{(1,3)}_{Y,K(L)}L = A^\dagger L + (I - A^\dagger A)Y(AY)^\dagger L\left(I - \left((I - AY(AY)^\dagger)L\right)^\dagger (I - AY(AY)^\dagger)L\right) .$$

The second term vanishes, if $I = \left((I - AY(AY)^\dagger)L\right)^\dagger (I - AY(AY)^\dagger)L$. It thus suffices to show that $\mathcal{N}\left((I - AY(AY)^\dagger)L\right) = \{0\}$. Note that $\mathcal{N}\left(I - AY(AY)^\dagger\right) = \mathcal{R}(AY)$. Since by assumption $\mathcal{R}(L) \cap \mathcal{R}(AY) = \{0\}$ it follows that $(I - AY(AY)^\dagger)Lv = 0$ implies $v = 0$.
(b) Since AY has full column rank it holds that

$$(AY)^\dagger = (Y^*A^*AY)^{-1}(AY)^* .$$

So $\mathcal{R}(L)\perp\mathcal{R}(AY)$ implies $(AY)^\dagger L = 0$ and therefore $K(L) = 0$.
(c) Let $\mathcal{R}(L) \subset \mathcal{N}(Y^*A^\dagger)$, i.e., $0 = Y^*A^\dagger L = Y^*A^{(1,3)}_{Y,K(L)}L$ by (a). Thus $\mathcal{R}(Y)\perp\mathcal{R}\left(A^{(1,3)}_{Y,K(L)}L\right)$.
Vice versa, if $\mathcal{R}(Y)\perp\mathcal{R}\left(A^{(1,3)}_{Y,K(L)}L\right)$, then $Y^*A^\dagger L = 0$ and thus $\mathcal{R}(L) \subset \mathcal{N}(Y^*A^\dagger)$. □

Theorem 3.5 allows to choose generalized inverses $A^\natural = A^{(1,3)}_{Y,K}$ that coincide with the Moore–Penrose inverse $A^\dagger$ on an arbitrary complement $\mathcal{R}(L)$ of $\mathcal{R}(AY)$. This leads us to two natural choices for L, where we require either $\mathcal{R}(L)\perp\mathcal{R}(AY)$ or $\mathcal{R}(Y)\perp\mathcal{R}(A^{(1,3)}_{Y,K}L) = \mathcal{R}(A^\dagger L)$. In the first case

$$A^\natural = A^{(1,3)}_{Y,0} = A^\dagger + (I - A^\dagger A)Y(AY)^\dagger \tag{3.5}$$

coincides with $A^\dagger$ on the orthogonal complement of $\mathcal{R}(AY)$, i.e., $x = A^\natural b$ is the minimal norm solution of (2.1) for all $b\perp\mathcal{R}(AY)$. In the second case, the space of all $x = A^\natural b$ that are minimal norm solutions of (2.1) for some b is orthogonal to $\mathcal{R}(Y)$, i.e., $\mathcal{R}(A^\dagger)\cap\mathcal{R}(A^\natural)\perp\mathcal{R}(Y)$. This choice is realized by

$$A^\natural = A^{(1,3)}_{Y,\mathcal{K}} \quad \text{with} \quad \mathcal{K} := Y(Y^*A^\dagger AY)^{-1}Y^*A^\dagger ,$$

which equals

$$A^\natural = A^{(1,3)}_{Y,\mathcal{K}} = A^\dagger + (I - A^\dagger A)Y(Y^*A^\dagger AY)^{-1}Y^*A^\dagger . \tag{3.6}$$

It can easily be seen that $A^{(1,3)}_{Y,\mathcal{K}}L = A^\dagger L$ if $\mathcal{R}(L) = \mathcal{N}(Y^*A^\dagger)$, in which case $\mathcal{R}(Y)\perp\mathcal{R}(A^{(1,3)}_{Y,\mathcal{K}}L)$. Moreover, the two matrices $A^{(1,3)}_{Y,0}$ and $A^{(1,3)}_{Y,\mathcal{K}}$ also fulfill the second Penrose condition.

Corollary 3.6 [DS12] *Let $A \in \mathbb{C}^{m\times n}$ and $Y \in \mathbb{C}^{n\times \ell}$ so that* rank $AY = \ell$*, then $A^{(1,3)}_{Y,0}$ and $A^{(1,3)}_{Y,\mathcal{K}}$ are $\{1,2,3\}$-inverses of A.*

Proof. According to Theorem 2.3 a matrix X is a $\{1,3\}$-inverses of a matrix A if and only if $AX = AA^\dagger$. Thus we know that

$$A^{(1,3)}_{Y,\mathcal{K}}AA^{(1,3)}_{Y,\mathcal{K}} = A^{(1,3)}_{Y,\mathcal{K}}AA^\dagger = A^{(1,3)}_{Y,\mathcal{K}} .$$

Since $(AY)^\dagger = (Y^*A^*AY)^{-1}Y^*A^*$ and $A^*AA^\dagger = A^*$ (cf. Lemma 1.1 (f)) we have

$$A^{(1,3)}_{Y,0}AA^{(1,3)}_{Y,0} = A^{(1,3)}_{Y,0}AA^\dagger = A^{(1,3)}_{Y,0} .$$

□

In the next paragraph we briefly discuss the advantage of $\{1,2,3\}$-inverses and also characterize all $\{1,2,3\}$-inverses fulfilling Problem 3.1.

$\{1,2,3\}$-inverses solving Problem 3.1

Let $X \in A\{1,2\}$ then we know from Lemma 1.4 that $X|_{\mathcal{R}(A)}$ is the inverse of $A|_{\mathcal{R}(X)}$. Furthermore, it is clear that for every $X \in A\{1,2\}$ it holds that $\operatorname{rank} X = \operatorname{rank} A$, this immediately follows from Penrose equation 1) and 2). Moreover, Bierhammer proved in [Bje58] that also the reverse direction holds if $X \in A\{1\}$,

Theorem 3.7 [Bje58] *Let $A \in \mathbb{C}^{m\times n}$ and $X \in A\{1\}$. Then*

$$X \in A\{1,2\} \Leftrightarrow \operatorname{rank} X = \operatorname{rank} A \,.$$

Thus, if $X \in A\{1,2,3\}\backslash A\{2\}$ is a pure $\{1,3\}$-inverse, Theorem 3.7 implies together with Penrose equation 1) that $\operatorname{rank} X > \operatorname{rank} A$. Hence $X|_{\mathcal{R}(A)} \notin \mathcal{L}(\mathcal{R}(A), \mathcal{R}(X))$ is not an inverse of $A|_{\mathcal{R}(X)}$.

In the next proposition we show that $A^{(1,3)}_{Y,K(L)}$ additionally satisfies the second Penrose equation if $\mathcal{R}(L) \subset \mathcal{R}(A)$.

Proposition 3.8 *Let $A \in \mathbb{C}^{m\times n}$ with $\operatorname{rank} A = r$ and $Y \in \mathbb{C}^{n\times \ell}$ such that $\operatorname{rank} AY = \ell$. Furthermore, let $L \in \mathbb{C}^{m\times(m-k)}$ with $1 \le k \le \ell$ and $\operatorname{rank} L = m-k$ such that $\mathcal{R}(L) \subset \mathcal{R}(A)$ and $\mathcal{R}(L) \cap \mathcal{R}(AY) = \{0\}$. Then*

$$A^{(1,3)}_{Y,K(L)} \in A\{1,2,3\} \,.$$

Proof. Recall from Lemma 3.4 that

$$A^{(1,3)}_{Y,K(L)} = A^\dagger + (I - A^\dagger A)Y(AY)^\dagger \left(I - L\left((I - AY(AY)^\dagger)L\right)^\dagger\right) \,.$$

Since $A^{(1,3)}_{Y,K(L)} \in A\{1,3\}$ it holds that $A^{(1,3)}_{Y,K(L)}AA^{(1,3)}_{Y,K(L)} = A^{(1,3)}_{Y,K(L)}AA^\dagger$.

To prove that $A^{(1,3)}_{Y,K(L)}AA^\dagger = A^{(1,3)}_{Y,K(L)}$ we first show

$$\left((I - AY(AY)^\dagger)L\right)^\dagger AA^\dagger = \left((I - AY(AY)^\dagger)L\right)^\dagger \,,$$

which is equivalent to

$$\left((I - AY(AY)^\dagger)L\right)^\dagger (I - AA^\dagger) = 0 \,. \tag{3.7}$$

Since $\mathcal{R}(L) \subset \mathcal{R}(A)$ there exists a V such that $L = AV$. This together with the fact that $(I - AY(AY)^\dagger)L$ has full rank (see proof of Theorem 3.5a) yields that the left hand side of equation (3.7) is equivalent to

$$(L^*(I - AY(AY)^\dagger)L)^{-1}V^*A^*(I - AY(AY)^\dagger)(I - AA^\dagger) \,. \tag{3.8}$$

It is obvious that

$$(I - AA^\dagger)\big(I - AY(AY)^\dagger\big)A = 0 \,.$$

Taking the conjugate transpose of the left hand side of the latter equation makes clear that (3.8) is equal to 0. □

We present now the class of all $\{1,2,3\}$-inverses that satisfy also condition (b) of Problem 3.1. According to [BIG03] all $\{1,2,3\}$-inverses X of a matrix A are characterized by the set

$$A\{1,2,3\} = \{A^\dagger + (I - A^\dagger A)ZA^\dagger : Z \in \mathbb{C}^{n\times m}\} \,. \tag{3.9}$$

As in Theorem 3.2 we have to characterize all Z such that

$$(I - A^\dagger A)ZA^\dagger AY = (I - A^\dagger A)Y \ .$$

Since we already know that there are $\{1,2,3\}$-inverses satisfying Problem 3.1 (e.g., $A_{Y,0}^{(1,3)}$), we do not have to explicitly verify the latter equation. Hence, by Theorem 2.2

$$A_{Y,K}^{(1,2,3)} := A^\dagger + (I - A^\dagger A)\left(Y(A^\dagger AY)^\dagger + K - K(A^\dagger AY)(A^\dagger AY)^\dagger\right)$$

satisfies condition (b) of Problem 3.1 for all $K \in \mathbb{C}^{n\times m}$.

Remark 3.9 *In [DS12, Theorem 2.1(b)] there is a citation error which was pointed out by Qingxiang Xu. Therefore, in [DS12, Theorem 2.1(b)] and [DS12, Lemma 2.3] the 2-norm has to be replaced by the Frobenius norm. This is already corrected in the corresponding Theorem 2.2(b) and Lemma 3.4 in this thesis.*

3.1 Geometrical ansatz

In this section we present a geometrical ansatz to solve Problem 3.1. Since we already characterized all solutions to Problem 3.1, this section does not bring new results but it might be useful for a better understanding.

We start by introducing the singular value decomposition of A. Let $A = U\Sigma V^*$ have rank r, and write the singular value decomposition of A as

$$A = U\Sigma V^* = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} = U_1\Sigma_1 V_1^* \ , \tag{3.10}$$

where $\Sigma_1 = \operatorname{diag}(\sigma_1, \ldots, \sigma_r) > 0$ is nonsingular and U_1 and V_1 have r linearly independent columns. Then Lemma 1.1 (g) implies that

$$A^\dagger = V_1\Sigma_1^{-1}U_1^* \ .$$

It is also a known fact that U_1 and V_1 span $\mathcal{R}(A)$ and $\mathcal{R}(A^*)$, respectively. Moreover, U_2 and V_2 span $\mathcal{N}(A^*)$ and $\mathcal{N}(A)$.

As we stated in Theorem 2.1 the Moore–Penrose inverse gives by $A^\dagger b$ the minimal norm solution to the least squares problem (2.1). Furthermore, it holds that $\mathcal{R}(A^*) = \mathcal{R}(A^\dagger)$, see Theorem 1.5. Assume that $\mathcal{R}(Y)$ is not a subset of $\mathcal{R}(A^*)$ and $\mathcal{N}(A)$, then the Moore–Penrose inverse does not satisfy Problem 3.1, i.e., $A^\dagger AY \neq Y$. But, we can obviously add any $y \in \mathcal{N}(A)$ to $A^\dagger b$ and still get a least squares solution, see equation (2.10). Hence an ansatz for an $A^\natural$ satisfying Problem 3.1 is

$$A^\natural = (V_1 + V_2L)\Sigma^{-1}U_1^* \quad \text{for an} \quad L \in \mathbb{C}^{(n-r)\times n} \ .$$

This means nothing else than rotating $\mathcal{R}(A^\dagger)$ by adding linear combinations of the basis vectors of $\mathcal{N}(A)$ to the basis vectors of $\mathcal{R}(A^\dagger)$. Hence, the only thing left is to determine L such that $\mathcal{R}(Y) \subset \mathcal{R}(A^\natural)$. Before doing this we state in the next Lemma 3.10 that for all L the corresponding matrix $A^\natural$ is a $\{1,2,3\}$-inverse. Then we characterize some L such that $A^\natural$ satisfies Problem 3.1 in the subsequent Proposition 3.11.

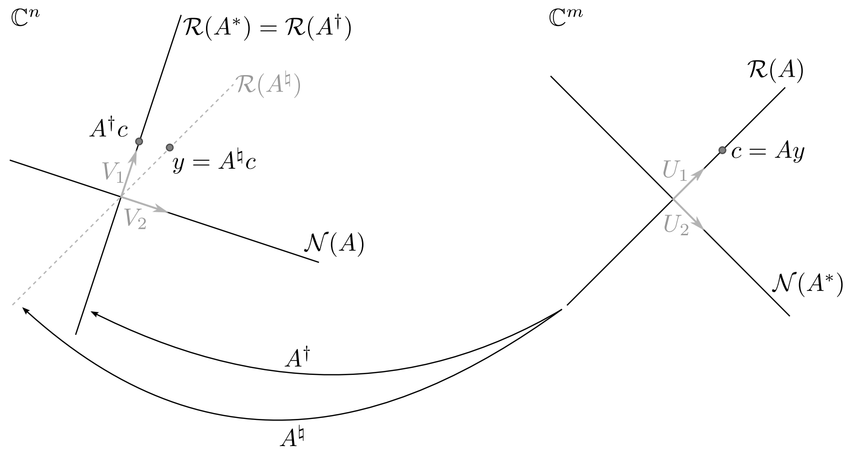


Figure 3.1: Geometric interpretation where $A \in \mathbb{C}^{m\times n}$ can be decomposed as in eq. (3.10)

Lemma 3.10 *Let A be as above and let $L \in \mathbb{C}^{(n-r)\times r}$. Then the matrix*

$$A^\natural = \Big(V_1 + V_2 L\Big)\Sigma_1^{-1} U_1^*$$

is a $\{1, 2, 3\}$-inverse of A.

Proof. Since $AV_2 = 0$ it is clear that $AA^\natural = AA^\dagger$. Hence 1) and 3) hold. Moreover, from $A^\natural = (I + V_2 L V_1^*)A^\dagger$ it follows

$$A^\natural A A^\natural = \Big(I + V_2 L V_1^*\Big) A^\dagger A A^\dagger = A^\natural \ .$$

Thus also condition 2) holds.

□

One could now determine all L such that Problem 3.1 is satisfied by applying Theorem 2.2 to the equation

$$V_2 L \Sigma_1^{-1} U_1^* AY = (I - V_1 V_1^*)Y \ .$$

This would end in cumbersome representations for L and since we already determined the class of all $\{1, 2, 3\}$-inverses that satisfy Problem 3.1, we present here only a subset of that class. This subset has an easy representation and will also cover the two most important choices $A^{(1,3)}_{Y,0}$ and $A^{(1,3)}_{Y,\mathcal{K}}$, as we will see below.

In fact for any matrix M with $M(AY) = Y$ the matrix $L = V_2^* M U_1 \Sigma_1$ results in an $A^\natural$ that solves Problem 3.1, as we show in the following proposition. Since for different M the matrix $A^\natural$ can differ we introduce the notation $A^\natural_M$ to distinguish.

Proposition 3.11 *Let A be as above and consider a matrix $Y \in \mathbb{C}^{n\times \ell}$ with* $\operatorname{rank} AY = \ell$. *Let M be any matrix that fulfills $M(AY) = Y$. Then the matrix*

$$A^\natural_M := A^\dagger + V_2 V_2^* M U_1 U_1^*$$

satisfies Problem 3.1.

Proof. Firstly, $A^\natural_M$ is a $\{1,2,3\}$-inverse of A. This is clear by Lemma 3.10 choosing $L = V_2^* M U_1 \Sigma_1$.
Secondly, it holds that

$$A^\natural_M AY = A^\dagger AY + V_2 V_2^* M A A^\dagger AY \ .$$

Now we use Penrose condition 1) and the fact that $A^\dagger AY = V_1 V_1^* Y$ and obtain

$$\begin{aligned} A^\natural_M AY &= V_1 V_1^* Y + V_2 V_2^* MAY \\ &= V_1 V_1^* Y + V_2 V_2^* Y \\ &= Y \ . \end{aligned}$$

In the last two steps we have used $MAY = Y$ and the identity $I = V_1 V_1^* + V_2 V_2^*$. □

In the next proposition we show the existence of such a matrix M, by giving two possible choices.

Proposition 3.12 *Let $AY =: C$ with $Y \in \mathbb{C}^{n \times \ell}$ and $\operatorname{rank} AY = \ell$. Then*

(a) $M_1 := Y(C^*C)^{-1}C^*$ and $M_2 := Y(Y^* V_1 V_1^* Y)^{-1} Y^* A^\dagger$ fulfill $M_1(AY) = Y$ and $M_2(AY) = Y$, respectively.

(b) $A^\natural_{M_1} = A^\dagger \Big(I - C(C^*C)^{-1}C^* \Big) + Y(C^*C)^{-1}C^*$.

(c) $A^\natural_{M_2} = A^\dagger + V_2 V_2^* Y (Y^* V_1 V_1^* Y)^{-1} Y^* A^\dagger$.

Proof. (a) Note that $\operatorname{rank} AY = \ell$ implies $\operatorname{rank} V_1^* Y = \ell$ whence $Y^* V_1 V_1^* Y \in \mathbb{C}^{\ell \times \ell}$ is regular. For M_1 it is trivial to see. For M_2 we have

$$\begin{aligned} M_2 AY &= Y(Y^* V_1 V_1^* Y)^{-1} Y^* A^\dagger AY \\ &= Y(Y^* V_1 V_1^* Y)^{-1} Y^* V_1 V_1^* Y \\ &= Y \ . \end{aligned}$$

(b) We will use the following facts:

(i) $V_1 V_1^* Y = A^\dagger AY = A^\dagger C$

(ii) $C^* U_1 U_1^* = C^*$

By definition we have that

$$A^\natural_{M_1} = A^\dagger + V_2 V_2^* Y (C^*C)^{-1} C^* U_1 U_1^* \ .$$

Using fact (ii) implies

$$A^\natural_{M_1} = A^\dagger + V_2 V_2^* Y (C^*C)^{-1} C^* \ .$$

Now we use the identity $I = V_1 V_1^* + V_2 V_2^*$ and get

$$A^\natural_{M_1} = A^\dagger + Y(C^*C)^{-1}C^* - V_1 V_1^* Y (C^*C)^{-1} C^* \ .$$

With (i) we have

$$A^\natural_{M_1} = A^\dagger \Big(I - C(C^*C)^{-1}C^* \Big) + Y(C^*C)^{-1}C^* \ .$$

(c) This holds because $A^\dagger U_1 U_1^* = A^\dagger$.

□

We will see in the next section (Remark 3.14) that $A^\natural_{M_1} = A^{(1,3)}_{Y,0}$ and $A^\natural_{M_2} = A^{(1,3)}_{Y,\mathcal{K}}$.

3.2 Computational aspects

In this section we derive representations of $A^{(1,3)}_{Y,0}$ and $A^{(1,3)}_{Y,\mathcal{K}}$ in terms of the singular value decomposition. Then we show in the case $AY = C$ that $A^{(1,3)}_{Y,0}$ and $A^{(1,3)}_{Y,\mathcal{K}}$ are robust in the sense that cropping of singular values typically does not influence their property of mapping Y back to C. In Theorem 3.16, we derive a limit representation for $A^{(1,3)}_{Y,\mathcal{K}}$ which can be used for a Tikhonov regularization. Finally, we discuss on how to obtain the solutions $A^{(1,3)}_{Y,0}b$ and $A^{(1,3)}_{Y,\mathcal{K}}b$ to the least squares problem efficiently.

Representation in terms of the SVD

As above, let $A = U\Sigma V^*$ have rank r, and write the singular value decomposition of A as

$$A = U\Sigma V = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} = U_1\Sigma_1 V_1^* \ , \tag{3.11}$$

where $\Sigma_1 = \operatorname{diag}(\sigma_1, \dots, \sigma_r) > 0$ is nonsingular and U_1 and V_1 have r columns. Then the Moore–Penrose inverse $A^\dagger$ is equal to $V_1\Sigma_1^{-1}U_1^*$. Thus the matrix $A^{(1,3)}_{Y,\mathcal{K}}$ can be written as

$$A^{(1,3)}_{Y,\mathcal{K}} = A^\dagger + V_2V_2^*Y(Y^*V_1V_1^*Y)^{-1}Y^*A^\dagger \ .$$

Before we continue we introduce another representation for $A^{(1,3)}_{Y,0}$.

Lemma 3.13 *Let $Y \in \mathbb{C}^{n\times\ell}$ satisfy* $\operatorname{rank} AY = \ell$ *and set $AY = C$. Then*

$$A^{(1,3)}_{Y,0} = A^\dagger\Big(I - C(C^*C)^{-1}C^*\Big) + Y(C^*C)^{-1}C^* \ .$$

Proof. This follows if we replace $(AY)^\dagger$ by $(C^*C)^{-1}C^*$ and AY by C in equation (3.5). □

Remark 3.14 *Hence, $A^\natural_{M_1} = A^{(1,3)}_{Y,0}$ and $A^\natural_{M_2} = A^{(1,3)}_{Y,\mathcal{K}}$, cf. Proposition 3.12.*

We show now that if we crop singular values of $A^{(1,3)}_{Y,0}$ and $A^{(1,3)}_{Y,\mathcal{K}}$ condition b) of Problem 3.1 stays valid.

Proposition 3.15 [DS12] *Let $Y \in \mathbb{C}^{n\times\ell}$ and $AY = C$ as above with* $\operatorname{rank} AY = \ell$ *and consider the matrix*

$$\tilde{A} := \tilde{U}_1\tilde{\Sigma}_1\tilde{V}_1^* \ ,$$

where $\tilde{\Sigma}_1 := \operatorname{diag}(\sigma_1, \dots, \sigma_k)$ for a $k < r$. The matrices $\tilde{U}_1$ and $\tilde{V}_1$ consist of the first k-columns of U_1 and V_1, respectively. Furthermore, let

$$\tilde{A}^{(1,3)}_{Y,0} := \tilde{A}^\dagger\Big(I - C(C^*C)^{-1}C^*\Big) + Y(C^*C)^{-1}C^*$$

and

$$\tilde{A}^{(1,3)}_{Y,\mathcal{K}} := \tilde{A}^\dagger + \tilde{V}_2\tilde{V}_2^*Y(Y^*\tilde{V}_1\tilde{V}_1^*Y)^{-1}Y^*\tilde{A}^\dagger \ .$$

Then it still holds that

(a) $\tilde{A}^{(1,3)}_{Y,0}C = Y$.

(b) $\tilde{A}^{(1,3)}_{Y,\mathcal{K}}C = Y$, *if* $\operatorname{rank} \tilde{A}Y = \ell$.

Proof. (a) Simple calculation implies

$$\tilde{A}_{Y,0}^{(1,3)} C = \tilde{A}^{\dagger}\Big(I - C(C^*C)^{-1}C^*\Big)C + Y(C^*C)^{-1}C^*C = Y\ .$$

(b) By construction, we have $\tilde{U}_1^* U_1 = [I_\ell, 0]$, so that

$$\tilde{A}^{\dagger} A = \tilde{V}_1 \tilde{\Sigma}_1^{-1} \tilde{U}_1^* U_1 \Sigma_1 V_1^* = \tilde{V}_1 \tilde{V}_1^*\ .$$

Hence

$$\tilde{A}_{Y,\mathcal{K}}^{(1,3)} AY = \tilde{V}_1 \tilde{V}_1^* Y + \tilde{V}_2 \tilde{V}_2^* Y (Y^* \tilde{V}_1 \tilde{V}_1^* Y)^{-1} Y^* \tilde{V}_1 \tilde{V}_1^* Y = (\tilde{V}_1 \tilde{V}_1^* + \tilde{V}_2 \tilde{V}_2^*) Y = Y\ ,$$

if $Y^* \tilde{V}_1 \tilde{V}_1^* Y \in \mathbb{C}^{\ell \times \ell}$ is non-singular, which is the case if $\operatorname{rank} \tilde{A} Y = \ell$. □

Thus, if the matrix A is ill-conditioned we can crop the smallest singular values without losing the desired property that C is still mapped to Y by $\tilde{A}_{Y,0}^{(1,3)}$. For $\tilde{A}_{Y,\mathcal{K}}^{(1,3)}$ we just need that $\operatorname{rank} \tilde{A} Y = \ell$ to preserve this property.

Efficient solution of Problem 3.1 with $A_{Y,0}^{(1,3)}$ and $A_{Y,\mathcal{K}}^{(1,3)}$

The computational effort to obtain a solution $A_{Y,K}^{(1,3)} b$ to the least squares problem (2.1) is quite high if one uses the singular value decomposition. Especially when A is sparse an iterative approach is more appropriate. Since

$$A^{\dagger} = \lim_{\tau \to 0} \left(A^* A + \tau^2 I\right)^{-1} A^*$$

and $A_{Y,0}^{(1,3)}$ can be written as in Lemma 3.13, we can obtain an approximation to the solution $A_{Y,0}^{(1,3)} b$ by $\hat{A}_{Y,0}^{(1,3)} b$, with

$$\hat{A}_{Y,0}^{(1,3)} := \left(A^* A + \tau^2 I\right)^{-1} A^* \left(I - C(C^*C)^{-1}C^*\right) + Y(C^*C)^{-1}C^* \quad \text{for a} \quad \tau > 0\ . \tag{3.12}$$

The solution $\hat{A}_{Y,0}^{(1,3)} b$ can efficiently be determined by first applying the CGLS-algorithm (cf. Section 2.2) to the least squares problem

$$\min_x \left\| \begin{bmatrix} A \\ \tau I \end{bmatrix} x - \begin{bmatrix} (I - C(C^*C)^{-1}C^*) b \\ 0 \end{bmatrix} \right\|_2^2 \quad \text{for a} \quad \tau > 0\ . \tag{3.13}$$

Secondly, $\hat{A}_{Y,0}^{(1,3)} b$ is obtained by adding $Y(C^*C)^{-1}C^* b$ to the minimal norm solution of (3.13). Note that $\hat{A}_{Y,0}^{(1,3)}(AY) = Y$ and hence condition (b) of Problem 3.1 is still preserved.

In the following theorem we derive a limit representation for $A_{Y,\mathcal{K}}^{(1,3)}$, which then also can be used for an iterative ansatz to obtain the solution $A_{Y,\mathcal{K}}^{(1,3)} b$.

Theorem 3.16 [DS12] *Let $A \in \mathbb{C}^{m \times n}$ and $Y \in \mathbb{C}^{n \times \ell}$ with $\operatorname{rank} AY = \ell$. Then if $Y^* Y = I$ it holds that*

$$A_{Y,\mathcal{K}}^{(1,3)} = \lim_{\tau \to 0} \left(A^* A + \tau^2 (I - YY^*)\right)^{-1} A^*\ .$$

Proof. We know that the matrix $A^*A + \tau^2(I - YY^*)$ is non-singular if

$$\mathcal{N}(A^*A) \cap \mathcal{N}(\tau^2(I - YY^*)) = \{0\} .$$

This is obviously satisfied because of the hypothesis $\operatorname{rank} AY = \operatorname{rank} Y = \ell$.

By the Sherman–Morrison–Woodbury-formula (cf. Lemma 3.19), it holds

$$\begin{aligned}\left(A^*A + \tau^2 I - \tau^2 YY^*\right)^{-1} A^* &= \left(A^*A + \tau^2 I\right)^{-1} A^* \\ &+ \left(A^*A + \tau^2 I\right)^{-1} \tau^2 Y \left(I - Y^* \left(A^*A + \tau^2 I\right)^{-1} \tau^2 Y\right)^{-1} Y^* \left(A^*A + \tau^2 I\right)^{-1} A^* ,\end{aligned}$$

where

$$\begin{aligned}\left(A^*A + \tau^2 I\right)^{-1} A^* &= V_1(\Sigma_1^2 + \tau^2 I)^{-1}\Sigma_1 U_1^* &\overset{\tau \to 0}{\to} A^\dagger , \qquad \text{and} \\ \left(A^*A + \tau^2 I\right)^{-1} \tau^2 &= V_1(\Sigma_1^2 + \tau^2 I)^{-1} V_1^* \tau^2 + V_2 V_2^* &\overset{\tau \to 0}{\to} V_2 V_2^* .\end{aligned}$$

Exploiting the identity $I = Y^*Y = Y^*(V_1V_1^* + V_2V_2^*)Y$, we obtain

$$\left(A^*A + \tau^2 I - \tau^2 YY^*\right)^{-1} A^* \overset{\tau \to 0}{\to} A^\dagger + V_2V_2^*Y(I - Y^*V_2V_2^*Y)^{-1}Y^*A^\dagger = A^{(1,3)}_{Y,\mathcal{K}} ,$$

which we wanted to show. □

Note that we can assume without loss of generality that $Y^*Y = I$. According to Corollary 3.3 $A^{(1,3)}_{Y,\mathcal{K}} = A^{(1,3)}_{YS,\mathcal{K}}$ for an arbitrary but regular $S \in \mathbb{C}^{\ell \times \ell}$. Since Y has full rank, Y^*Y is symmetric positive definite. Hence there exists a unitary matrix $V \in \mathbb{C}^{\ell \times \ell}$ such that $V^*(Y^*Y)V = D$, where D is a diagonal matrix containing the eigenvalues. Now, let E also be a diagonal matrix, with $E(i,i) = 1/\sqrt{D(i,i)}$. Then $S := VE$ yields $(YS)^*(YS) = I$.

Corollary 3.17 *Consider the Tikhonov regularization from Section 2.2 with $T = (I - YY^*)$ and w.l.o.g. $Y^*Y = I$. Then*

$$A^{(1,3)}_{Y,\mathcal{K}} b = \lim_{\tau \to 0} x_{T,\tau} .$$

So an approximation to the solution $A^{(1,3)}_{Y,\mathcal{K}} b$, of the least squares problem (2.1), can efficiently be obtained by applying the CGLS-algorithm to the least squares problem

$$\min_x \left\| \begin{bmatrix} A \\ \tau(I - YY^*) \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2^2 \quad \text{with} \quad \tau > 0 . \tag{3.14}$$

Remark 3.18 *By Proposition 4.2 below, it holds that $\left(A^*A + \tau^2(I - YY^*)\right)^{-1} A^*(AY) = Y$ for all $\tau > 0$. Thus condition (b) of Problem 3.1 is also preserved in this approximative case.*

Special case: Rank Y small

Assume that A is sparse, and that $Y \in \mathbb{R}^{n \times \ell}$ with $\ell \ll n$ has few zero entries. Then in view of equation (3.14) the matrix $[A, \tau(I - YY^T)]^T$ is not sparse anymore, because of the nearly fully occupied matrix $\operatorname{rank} \ell$ matrix YY^T. In this paragraph we show that we can solve $\ell + 1$ sparse systems instead. To achieve this we make use of the Sherman–Morrison–Woodbury-formula:

Lemma 3.19 [GVL96, page 50] *Let $A \in \mathbb{R}^{n \times n}$ be non-singular and $U, V \in \mathbb{R}^{n \times \ell}$ with $\operatorname{rank} UV^T = \ell$. Furthermore, let $(I + V^TA^{-1}U)$ be non-singular. Then it holds that*

$$(A - UV^T)^{-1} = A^{-1} + A^{-1}U(I - V^TA^{-1}U)^{-1}V^TA^{-1} . \tag{3.15}$$

Furthermore, we exploit that a solution to the least squares problem (3.14) is equal to

$$x = (\overbrace{A^TA + \tau^2 I}^{=:\hat{A}} - \tau^2 YY^T))^{-1} A^T b \;, \tag{3.16}$$

where we assume, as in Theorem 3.16, that without loss of generality $Y^TY = I$. Thus, by the Sherman–Morrison–Woodbury-formula (3.15) equation (3.16) is equivalent to

$$x = \Big(\hat{A}^{-1} + \tau^2 \hat{A}^{-1} Y (I - \tau^2 Y^T \hat{A}^{-1} Y)^{-1} Y^T \hat{A}^{-1}\Big) A^T b \;,$$

where $I - \tau^2 Y^T \hat{A}^{-1} Y$ is as explained in the proof of Theorem 3.16 non-singular. So we can obtain the solution x by solving the $\ell + 1$ systems:

$$\hat{A} x_0 = A^T b \quad \text{and} \quad \hat{A} x_i = Y(:, i) \text{ for } \; i = 1{:}\ell \;.$$

Again, this can efficiently be done by applying the CGLS-algorithm to the $\ell + 1$ sparse least squares problems

$$\min_{x_0} \left\| \begin{bmatrix} A \\ \tau I \end{bmatrix} x_0 - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2^2 \quad \text{and} \quad \min_{x_i} \left\| \begin{bmatrix} A \\ \tau I \end{bmatrix} x_i - \begin{bmatrix} Y(:, i) \\ 0 \end{bmatrix} \right\|_2^2 \quad \text{for } \; i = 1{:}\ell \; \text{ and a } \; \tau > 0 \;.$$

Finally, let $X := [x_1, \ldots, x_\ell]$, then the solution x to the least squares problem (3.14) is equal to

$$x = x_0 + \tau^2 X (I - \tau^2 Y^T X)^{-1} Y^T x_0 \;.$$

Rank $Y = 1$

If $Y \in \mathbb{R}^{n \times 1}$ the above procedure to obtain the solution x reduces to the solution of the two systems

$$\hat{A} x_0 = A^T b \quad \text{and} \quad \hat{A} x_1 = Y \;,$$

and finally

$$x = x_0 + x_1 \frac{\tau^2 Y^T x_0}{1 - \tau^2 Y^T x_1} \;.$$

4 Tikhonov regularizations and Problem 3.1

In Section 3 we searched for generalized inverses $A^\natural$ that satisfy Problem 3.1, i.e., that solve the least squares problem (2.1) by $A^\natural b$ and additionally fulfill $A^\natural(AY) = Y$. We can formulate something similar for the Tikhonov regularization

$$\min_x \|Ax - b\|_2^2 + \tau^2 \|Tx\|_2^2 \;,$$

which we presented in Section 2.2.

Problem 4.1 *Find a matrix T with $\mathcal{N}(A) \cap \mathcal{N}(T) = \{0\}$ such that*

$$\left(A^*A + \tau^2 T^*T\right)^{-1} A^*(AY) = Y \quad \textit{for} \quad \tau > 0 \;. \tag{4.1}$$

It is quite easy to see that all matrices T that satisfy this problem need the property that $\mathcal{R}(Y) \subset \mathcal{N}(T)$, as we show in the following proposition.

Proposition 4.2 *Consider $A \in \mathbb{C}^{m\times n}$ and $Y \in \mathbb{C}^{n\times \ell}$ with* $\operatorname{rank} AY = \ell$. *Furthermore, let $\tau \neq 0$ and $T \in \mathbb{C}^{k\times n}$ for $1 \le k \le n$ with $\mathcal{N}(A) \cap \mathcal{N}(T) = \{0\}$. Then*

$$\left(A^*A + \tau^2 T^*T\right)^{-1} A^*(AY) = Y \quad \Leftrightarrow \quad \mathcal{R}(Y) \subset \mathcal{N}(T) \; .$$

Proof. Let $\mathcal{R}(Y) \subset \mathcal{N}(T)$ then $A^*AY = (A^*A + \tau^2T^*T)Y$ which is equivalent to

$$\left(A^*A + \tau^2 T^*T\right)^{-1} A^*AY = Y \; .$$

Vice versa, let $\left(A^*A + \tau^2 T^*T\right)^{-1} A^*(AY) = Y$. This is again equivalent to

$$(A^*A + \tau^2T^*T)Y = A^*AY \; ,$$

which in turn is equivalent to $T^*TY = 0$. Hence $\mathcal{R}(Y) \subset \mathcal{N}(T)$. □

Moreover, we can show that, in the limit, all solutions to the Tikhonov regularization can also be obtained by a $\{1,2,3\}$-inverse of A.

Theorem 4.3 *Let $A \in \mathbb{C}^{m\times n}$ and $T \in \mathbb{C}^{k\times n}$ for $1 \le k \le n$ with $\mathcal{N}(A) \cap \mathcal{N}(T) = \{0\}$, then there exists an $X \in A\{1,2,3\}$ such that*

$$X = \lim_{\tau\to 0} \left(A^*A + \tau^2 T^*T\right)^{-1} A^*,$$

which implies that $Xb = \lim_{\tau\to 0} x_{T,\tau}$.

Proof. Recall that by the Sherman–Morrison–Woodbury-formula it holds that

$$(A - UV^*)^{-1} = A^{-1} + A^{-1}U(I - V^*A^{-1}U)^{-1}V^*A^{-1} \; ,$$

see Lemma 3.19. Adding 0 to $A^*A + \tau^2T^*T$ yields

$$\underbrace{(A^*A + \tau^2 I)}_{=:\hat{A}} - \tau^2 \underbrace{(I - T^*T)}_{=:U} \cdot \underbrace{(I)}_{=:V^*} \; .$$

Hence by the Sherman–Morrison–Woodbury-formula

$$\left(A^*A + \tau^2 T^*T\right)^{-1} A^* = \hat{A}^{-1}A^* + \hat{A}^{-1}\tau^2 U(I - \hat{A}^{-1}\tau^2 U)^{-1}\hat{A}^{-1}A^* \; .$$

In the proof of Theorem 3.16 we already showed that

$$\lim_{\tau\to 0} \hat{A}^{-1}\tau^2 = I - A^\dagger A \; .$$

Furthermore, it holds that

$$\lim_{\tau\to 0} \hat{A}^{-1}A^* = A^\dagger \; .$$

So altogether we have

$$\lim_{\tau\to 0} \left(A^*A + \tau^2 T^*T\right)^{-1} A^* = A^\dagger + (I - A^\dagger A)ZA^\dagger =: X \; ,$$

for $Z = U(I - (I - A^\dagger A)U)^{-1}$. Hence, X indeed is a $\{1,2,3\}$-inverse of A, see equation (3.9). □

But not for every $\{1,2,3\}$-inverse there exists a Tikhonov regularization.

Corollary 4.4 *Let $A \in \mathbb{C}^{m\times n}$ and $T \in \mathbb{C}^{k\times n}$ for $1 \le k \le n$ with $\mathcal{N}(A) \cap \mathcal{N}(T) = \{0\}$, then there exist $X \in A\{1,2,3\}$ such that*

$$X \neq \lim_{\tau\to 0} \left(A^*A + \tau^2 T^*T\right)^{-1} A^* \, .$$

Proof. As above let $\hat{A} := A^*A + \tau^2 I$ and choose $Z = U(I - V^*(I - A^\dagger A)U)^{-1}V^*$ with arbitrary but feasible U and V, then

$$\begin{aligned} A^\dagger + (I - A^\dagger A)ZA^\dagger &= \lim_{\tau\to 0} \hat{A}^{-1}A^* + \hat{A}^{-1}\tau^2 U(I - V^*\hat{A}^{-1}\tau^2 U)^{-1}V^*\hat{A}^{-1}A^* \\ &= \lim_{\tau\to 0} (\hat{A} - \tau^2 UV^*)^{-1}A^* \\ &= \lim_{\tau\to 0} \left(A^*A + \tau^2(I - UV^*)\right)^{-1} A^* \end{aligned}$$

Obviously, one can choose U and V such that $T^*T \neq I - UV^*$ for all feasible T. □

So Theorem 4.3 and Proposition 4.2 imply that if $\mathcal{R}(Y) \subset \mathcal{N}(T)$ all $\{1,2,3\}$-inverses which result from a Tikhonov regularization, i.e., $A^\natural = \lim_{\tau\to 0}(A^*A + \tau^2 T^*T)^{-1}A^*$, satisfy Problem 3.1 in the limit $\tau \to 0$.

5 Special case $AE_n = E_m$

Later, in Section IV.2, we present a method to approximate scattered data. There we have to solve an ordinary least squares problem, where the matrix A has the property $AE_n = E_m$, with $E_n := [1,\ldots,1]^T \in \mathbb{R}^n$. In our method to approximate scattered data we figured out that the minimal norm solution to the least squares problem is not the right choice. But solutions to Problem 3.1 or 4.1, with $Y = E_n$, delivered much better results. This will be more enlightened in Section IV.2.3. In this section here we interpret the solutions $A^{(1,3)}_{E_n,0}b$ and $A^{(1,3)}_{E_n,\mathcal{K}}b$ to Problem 3.1 and for Problem 4.1 we present choices for T.

Interpretation of $A_{E_n,0}b$

From Lemma 3.13 we know that

$$\begin{aligned} A^{(1,3)}_{E_n,0}b &= A^\dagger\Big(I - E_m(E_m^TE_m)^{-1}E_m^T\Big)b + E_n(E_m^TE_m)^{-1}E_m^Tb \\ &= A^\dagger\Big(b - E_m\frac{E_m^Tb}{m}\Big) + E_n\frac{E_m^Tb}{m} \, . \end{aligned}$$

Since E_m^Tb/m is the mean of b, the solution $A^{(1,3)}_{E_n,0}b$ to the least squares problem (2.1) is obtained by first subtracting the mean of b from b, i.e., $\hat{b} := b - E_mE_m^Tb/m$. Then the minimal norm solution to $\hat{b}$ is computed and finally the mean of b is added again.

Furthermore, the solution $A^{(1,3)}_{Y,0}b$ to the least squares problem is as close as possible to the mean of b.

Proposition 5.1 [DS12] *Let $AY = C$ with $Y = E_n$, $C = E_m$. Then $x = A^{(1,3)}_{Y,0}b$ is the solution of the least squares problem (2.1) which minimizes*

$$\left\| A^{(1,3)}_{Y,0}b - \frac{E_nE_m^Tb}{m} \right\|_2 \, .$$

Proof. First notice that $A_{Y,0}^{(1,3)} E_m = E_n$, then a short calculation yields

$$\left\| A_{Y,0}^{(1,3)} b - \frac{E_n E_m^T b}{m} \right\|_2 = \left\| A_{Y,0}^{(1,3)} \left(b - \frac{E_m E_m^T}{m} \right) \right\|_2 = \left\| A^\dagger \left(b - \frac{E_m E_m^T b}{m} \right) \right\|_2 .$$

□

Interpretation of $A_{E_n,\mathcal{K}} b$

To interpret the solution $A_{E_n,\mathcal{K}} b$ we take a look on the regularization derived at Theorem 3.16

$$A_{E_n,\mathcal{K}}^{(1,3)} b = \lim_{\tau \to 0} \left(A^T A + \tau^2 \left(I - \frac{E_n E_n^T}{n} \right) \right)^{-1} A^T b .$$

As stated in Corollary 3.17 this is the same as the solution to the Tikhonov regularization

$$\min_x \|Ax - b\|_2^2 + \tau^2 \|Tx\|_2^2 \quad \text{for} \quad \tau \to 0 ,$$

with

$$T = I - \frac{E_n E_n^T}{n} = \begin{bmatrix} 1 - \frac{1}{n} & -\frac{1}{n} & -\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n} & 1 - \frac{1}{n} & -\frac{1}{n} & \cdots & -\frac{1}{n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -\frac{1}{n} & \cdots & -\frac{1}{n} & 1 - \frac{1}{n} & -\frac{1}{n} \\ -\frac{1}{n} & -\frac{1}{n} & \cdots & -\frac{1}{n} & 1 - \frac{1}{n} \end{bmatrix} . \tag{5.1}$$

Since $\frac{1}{n}\|Tx\|_2^2$ equals the variance of the components of x, this regularization balances the solution in a way that the variance of its components is kept small.

Choices for Problem 4.1

Considering Proposition 4.2 the regularization matrix T of the Tikhonov regularization has to be chosen such that $E_n \in \mathcal{N}(T)$ and $\mathcal{N}(A) \cap \mathcal{N}(T) = \{0\}$. Clearly, one choice is $T = I - (E_n E_n^T)/n$ from the paragraph above. Also the discrete Laplace operator with homogenous Neumann boundary conditions, which is applied in Chapter IV, meets this property, see (IV.2.11).

Chapter II

Multivariate Polynomial Interpolation

This chapter deals with multivariate polynomial interpolation, where the main focus lies on correct sets, i.e., sets of interpolation sites that can be interpolated by a unique polynomial. In Section 2 we start by presenting several classes of sets known to be correct, like generalized principal lattices, whereas in Section 2.4 we characterize a class of correct sets, which is shown to be more general than existing classes of correct sets. Moreover, we present a new and concrete recipe, which yields elements in that newly characterized class.

1 Introduction

Univariate polynomial interpolation has a rather long history and its theory is already well settled. Compared to that, the multivariate counterpart, i.e., polynomial interpolation in several variables, is more complex, as we will learn in Section 2. Moreover, the multivariate counterpart is fairly new, according to [CG10] it systematically started developing in the second half of the 20th century and is still an active research area. Though there are no books which solely treat multivariate polynomial interpolation, there are several survey articles which do. We emphasize in particular [GS00a], [GS00b], [Sau06] and [CG10].

Before we continue and start putting our focus on correct sets we present some standard notation, which is mostly borrowed from the mentioned surveys. In this chapter let $\mathbb{F}$ be either $\mathbb{R}$ or $\mathbb{C}$. Denote by

$$\Gamma_{n,d} := \{\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{Z}_+^d : |\alpha| \leq n\}$$

a set of multi-indices, where

$$|\alpha| := \sum_{j=1}^{d} \alpha_j \ .$$

Let $\xi := (\xi_1, \ldots, \xi_d)$ be a set of indeterminates, then $\xi^\alpha := \xi_1^{\alpha_1} \cdots \xi_d^{\alpha_d}$. The polynomial ring in d variables is denoted by

$$\Pi^d := \Big\{ \sum_{\alpha \in \mathbb{Z}_+^d} a_\alpha \xi^\alpha : a_\alpha \in \mathbb{F} \quad \text{with} \quad a_\alpha = 0 \quad \text{for almost all} \quad \alpha \in \mathbb{Z}_+^d \Big\} \ .$$

The degree of a polynomial $p \in \Pi^d$ is defined by

$$\deg p := \max\{|\alpha| : a_\alpha \neq 0\} ,$$

if $p \neq 0$ and $\deg p := -1$ for $p = 0$. Furthermore, we denote by

$$\Pi_n^d := \{p \in \Pi^d : \deg p \leq n\}$$

the subspace of all d-variate polynomials with degree at most n. The basic problem in multivariate polynomial interpolation is:

Problem 1.1 *Let $S \subset \mathbb{Z}_+^d$ be finite and consider a set of $\#S$ distinct points $\{x_\alpha \in \mathbb{F}^d : \alpha \in S\}$, some constants $\{y_\alpha \in \mathbb{F} : \alpha \in S\}$ and a subspace $V \subset \Pi^d$. Then find a polynomial $p \in V$ such that*

$$p(x_\alpha) = y_\alpha \quad \text{for all} \quad \alpha \in S . \tag{1.1}$$

This problem is also referred to as Lagrange interpolation problem, see, e.g., [GS00b] where also the formulation of Problem 1.1 is partially borrowed from. The points x_α are also called *nodes* or *sites*.

The research area in the context of Lagrange interpolation mainly consists of two parts. One is to find an interpolation space $V \subset \Pi_n^d$ for a given set of interpolation nodes $X = \{x_\alpha : \alpha \in S\}$ such that for any choice of constants $\{y_\alpha : \alpha \in S\}$ there exists a unique polynomial $p \in V$ that fulfills equation (1.1), see, e.g., [BR90] where it is also shown that such a polynomial subspace V always exists. The other part, which is considered the mainstream, is to find a set of interpolation sites $X = \{x_\alpha : \alpha \in S = \Gamma_{n,d}\}$ such that for any constants $\{y_\alpha : \alpha \in \Gamma_{n,d}\}$ there exists a unique $p \in V = \Pi_n^d$ satisfying equation (1.1). Note that necessarily

$$\dim \Pi_n^d = \#X = \#\Gamma_{n,d} = \binom{n+d}{n} .$$

Sets of interpolation nodes X are called *correct* if the Lagrange interpolation problem 1.1 always has a unique solution. Some authors also use the terms *poised*, *unisolvent* or *regular*. The task of finding correct sets is for instance crucial in constructing filters for the lifting scheme, as we will learn in Chapter III. According to [GS00a] correct sets are also needed in finite element analysis, see [CR72] for one of the most important papers.

In this thesis we follow the mainstream and look for distributions of points that are correct in Π_n^d. We get more concrete about correct sets in the next section, where we start by presenting known classes of correct sets. Furthermore, in Section 2.4 we introduce a new characterization of a class of correct sets, which will be shown to be more general.

2 Sets correct for multivariate polynomial interpolation

From now on we call a set of distinct nodes $X := \{x_\alpha \in \mathbb{F}^d : \alpha \in \Gamma_{n,d}\}$ n-correct, or (n,d)-correct, for the interpolation space Π_n^d if the Lagrange interpolation problem 1.1 on X has always a unique solution in Π_n^d, i.e., for every $f : \mathbb{F}^d \to \mathbb{F}$ there exists a unique $p \in \Pi_n^d$ such that $p(x) = f(x)$ for all $x \in X$. A more formal definition of an n-correct set is for instance given in [Boo09]:

Definition 2.1 *A set X of $\dim \Pi_n^d$ distinct nodes $x \in \mathbb{F}^d$ is n-correct if the restriction map*

$$\Pi_n^d \longrightarrow \mathbb{F}^X \ : \ p \mapsto p|_X := (p(x) : x \in X) \tag{2.1}$$

is invertible.

In the univariate case ($d = 1$) finding nodes that form an n-correct set is an easy task. One just needs $n + 1$ pairwise distinct nodes. For $d > 1$ this is more complex, since here the nodes need some geometric structure. Consider for instance the linear case for $d = 2$ and $\mathbb{F} = \mathbb{R}$. Hence $X = \{x_\alpha \in \mathbb{R}^2 : \alpha \in \Gamma_{1,2}\}$ consists of three nodes. Assume that these three points lie on a single straight line. Then there are either infinitely many or no solutions to the Lagrange interpolation problem 1.1, dependent on the values $y_\alpha \in \mathbb{R}$ with $\alpha \in \Gamma_{1,2}$.

In fact a set X of $\#\Gamma_{n,d}$ distinct points in $\mathbb{F}^d$ is (n, d)-correct if and only if X is not a subset of any hypersurface of degree n, see, e.g., [Coa66]. But usually it is very hard to state whether a given set of points lies on such a hypersurface or not, especially for high dimension d or degree n. Therefore, in the classical paper [CY77], Chang and Yao give a simpler sufficient geometric condition for nodes to form an n-correct set, which we present in the next section.

Most of the sets, correct for interpolation, that we present in the following are obtained by the intersection of several hyperplanes, where a hyperplane H is defined as

$$H := \{x \in \mathbb{F}^d : h(x) := \langle a, x\rangle + c = 0,\ a \in \mathbb{F}^d\backslash\{0\},\ c \in \mathbb{F}\}\ .$$

2.1 The geometric characterization of Chung and Yao

Definition 2.2 [CY77] *As above let $X := \{x_\alpha : \alpha \in \Gamma_{n,d}\}$ be a set of $\#\Gamma_{n,d}$ distinct points. Then the set X is said to fulfill the geometric characterization, short* GC*, if for every node x_α there exist n distinct hyperplanes whose union contain $X\backslash x_\alpha$ but not x_α itself. Such a set X is also referred to as a* GC$_n$*-set.*

Theorem 2.3 [CY77] *Every* GC$_n$*-set is n-correct.*

Proof. Let X be a GC$_n$-set and let

$$H_\alpha^j \quad \text{for} \quad j = 1, \dots, n$$

be the n hyperplanes that contain $X\backslash x_\alpha$. Since any hyperplane is the zero set of a polynomial of degree 1, there exists a unique polynomial, up to a constant multiple, $h_\alpha^j \in \Pi_1^d$ defining H_α^j by $\{x \in \mathbb{F}^d : h_\alpha^j(x) = 0\}$ for $j = 1, \dots, n$.

Let

$$p_\alpha := \prod_{j=1}^{n} h_\alpha^j\ .$$

Then due to the geometric characterization $p_\alpha(x_\alpha) \neq 0$ and $p_\alpha(x_\beta) = 0$ for all $\beta \in \Gamma_{n,d}\backslash\alpha$. Thus the polynomial

$$p := \sum_{\alpha\in\Gamma_{n,d}} \frac{p_\alpha}{p_\alpha(x_\alpha)} f(x_\alpha)$$

satisfies $p(x) = f(x)$ for all $x \in X$ and any $f : \mathbb{F}^d \to \mathbb{F}$. Hence the restriction map (2.1) is onto, which implies that $\dim \Pi_n^d \geq \#X$. Then because of the hypothesis that $\#X = \dim \Pi_n^d$ it holds that the map (2.1) is invertible and hence X is n-correct. □

In general, the problem with the geometric characterization of Chang and Yao is that it still does not provide a general recipe to construct GC$_n$-sets. In the following Sections 2.2 and 2.3 we present important classes of GC$_n$-sets which give suggestions on how to construct GC$_n$-sets, whereas in Section 2.4 we characterize a correct class. Moreover, we present a new and concrete recipe that produces elements in that class.

2.2 Natural lattices

Chung and Yao provided, also in [CY77], the first specific method to construct correct sets that fulfill their geometric characterization, namely the approach of *natural lattices*:

Definition 2.4 [CY77] *Let $m = n+d$, then X is called a natural lattice of degree n if there exists a collection of m hyperplanes $\mathcal{H}$ such that any choice of d distinct hyperplanes of $\mathcal{H}$ intersects in exactly one $x \in X$. Moreover, different choices yield different x.*

Theorem 2.5 [CY77] *Every natural lattice X of degree n is a* GC_n*-set.*

Proof. Let X be a natural lattice of degree n and denote by $\mathcal{H} = \{H_1, \dots, H_m\}$ the collection of its $m = n + d$ constructing hyperplanes. Let $x \in X$ be arbitrary. By definition of the natural lattice and since $\#X = \binom{n+d}{n}$ there exist d hyperplanes, without loss of generality $H_1, \dots, H_d$, which intersect in $x \in X$. Hence, by definition, x cannot belong to one of the n hyperplanes $H_{d+1}, \dots, H_m$.

Let $y \in X$ be arbitrary with $y \neq x$. Then it is left to show that there exists a hyperplane in $\{H_{d+1}, \dots, H_m\}$ which contains y. This is again clear by definition, because there exists a collection of d hyperplanes, different from $\{H_1, \dots, H_d\}$, which intersect in y. Hence there must obviously be a hyperplane in $\{H_{d+1}, \dots, H_m\}$ containing y. □

In Figure 2.1 we present an example of a natural lattice of degree 2 for $d = 2$. On the left the natural lattice with its $m = 4$ constructing hyperplanes $\mathcal{H}$ is depicted, whereas the six figures on the right show all possible choices of 2 hyperplanes out of $\mathcal{H}$ having exactly one point in common.

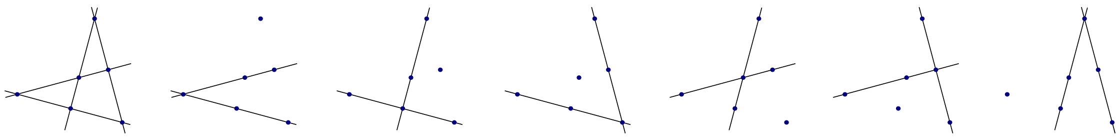


Figure 2.1: Natural lattice for $d = n = 2$

2.3 (Fully) generalized principal lattices

Generalized principal lattices were introduced in [CGS06], whereas in [Boo09] a different but equivalent definition is given. In what follows let

$$\tilde{\Gamma}_{n,d} := \{(n - |\alpha|, \alpha) : \alpha \in \Gamma_{n,d}\} \subset \mathbb{Z}^{d+1}$$

denote the set of homogenized multi-indices from $\Gamma_{n,d}$.

Definition 2.6 [Boo09] *A set X is called generalized principal lattice of degree n (GPL_n) if it can be indexed as*

$$X = \{x_\alpha : \alpha \in \tilde{\Gamma}_{n,d}\}$$

so that there exists a collection of hyperplanes

$$\mathcal{H} := \left(H_i^j : i \in 0{:}(n-1), j \in 0{:}d\right)$$

such that it holds that for all applicable $\alpha \in \tilde{\Gamma}_{n,d}$, r and i

$$\bigcap_{j \neq r} H_{\alpha_j}^j = \{x_\alpha\} \subset H_{\alpha_r}^r \ , \tag{2.2}$$

while

$$x_\alpha \in H_i^j \quad \Longrightarrow \quad \alpha_j = i\ . \tag{2.3}$$

Corollary 2.7 [Boo09] *Let X be a generalized principal lattice of degree n and $\mathcal{H}$ the collection of hyperplanes constructing it. Then $\#\mathcal{H} = n(d+1)$, which means that all hyperplanes $H_i^j \in \mathcal{H}$ are pairwise distinct.*

Proof. Assume that $H_i^j = H_s^r$ for some $i, s < n$. Then by (2.2)

$$x_\alpha \in H_i^j = H_s^r \quad \text{for all} \quad \alpha \in \tilde{\Gamma}_{n,d} \quad \text{with} \quad \alpha_j = i\ .$$

But (2.3) implies that

$$\alpha_r = s \quad \text{for all} \quad \alpha \in \tilde{\Gamma}_{n,d} \quad \text{with} \quad \alpha_j = i\ ,$$

which can only hold if $j = r$ and $i = s$. □

Theorem 2.8 [CGS06] *Every* GPL$_n$*-set is a* GC$_n$*-set.*

Proof. [Boo09] Let X be a GPL$_n$-set and let $\mathcal{H} := (H_i^j : i \in 0{:}(n-1), j \in 0{:}d)$ be the collection of its $n(d+1)$ constructing hyperplanes. To show that X is a GC$_n$-set we have to prove that for every $x_\alpha \in X$ the set $X \backslash x_\alpha$ is contained in the union of n hyperplanes that does not contain x_α.

So let $x_\alpha \in X$ be fixed and define

$$\tilde{\mathcal{H}} := \{H_i^j \in \mathcal{H} : i < \alpha_j\}\ .$$

Then $\tilde{\mathcal{H}}$ consists of n hyperplanes. Obviously for every $\beta \in \tilde{\Gamma}_{n,d} \backslash \alpha$ there exists a j with $\beta_j < \alpha_j$. Hence by (2.2) $\tilde{\mathcal{H}}$ contains all x_β with $\beta \in \tilde{\Gamma}_{n,d} \backslash \alpha$, i.e., $\mathcal{H}$ contains $X \backslash x_\alpha$, but not x_α itself because of condition (2.3). □

In [Boo09] it was noticed that in this proof only

$$\alpha_r < n \quad \Longrightarrow \quad x_\alpha \in H_{\alpha_r}^r$$

and

$$x_\alpha \in H_i^j \quad \Longrightarrow \quad \alpha_j \leq i\ ,$$

is used. Thus, not the full power of (2.2) and (2.3) is needed. This results in the definition of fully generalized principal lattices.

Definition 2.9 [Boo09] *A fully generalized principal lattice of degree n (or,* FGPL$_n$*-set for short) is a set X in $\mathbb{F}^d$ that can be so indexed as $X = \{x_\alpha : \alpha \in \tilde{\Gamma}_{n,d}\}$ that*

$$\alpha_r < n \implies x_\alpha \in H_{\alpha_r}^r \tag{2.4}$$

and

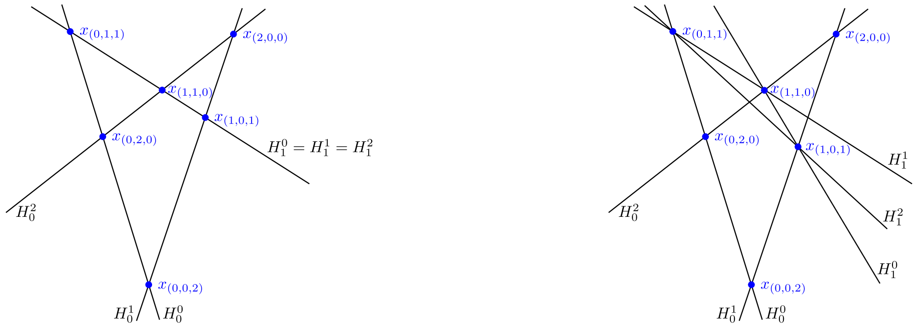
$$x_\alpha \in H_i^j \implies \alpha_j \leq i \tag{2.5}$$

hold for some collection $\mathcal{H} := (H_i^j : i \in 0{:}(n-1), j \in 0{:}d)$ of hyperplanes and all applicable α, r, and i.

Theorem 2.10 [Boo09] *Any* FGPL$_n$*-set is a* GC$_n$*-set.*

Proof. See the proof of Theorem 2.8 and the subsequent lines. □

By a result of [CGS09] we know that each natural lattice of degree 2 is an FGPL$_2$-set, see [Boo09]. Hence the class of FGPL$_n$-sets is strictly larger than the class of GPL$_n$-sets. This can also be seen in the following figure, where we depict a FGPL$_2$-set and its perturbation into a GPL$_2$-set. Note that the FGPL$_2$-set evidently is also a natural lattice of degree 2. The main idea of this figure is borrowed from [Boo09, Figure 1].



(a) Example of an FGPL$_2$-set

(b) Example of a GPL$_2$-set

Figure 2.2: FGPL$_2$-set and its perturbation into a GPL$_2$-set for $d = 2$

2.4 On Radon's recipe

In [Rad48] Radon proposed a recipe to construct n-correct sets for the bivariate ($d = 2$) case. More precisely, given an $(n-1, 2)$-correct set Y and a set Z of $n+1$ distinct points on a straight line H in $\mathbb{R}^2$. Then if $Y \cap H = \emptyset$ the set $X = Y \cup Z$ is $(n, 2)$-correct. By an observation from [GR70] this recipe works also for $d \geq 2$. Given an $(n-1, d)$-correct set Y and a $(n, d-1)$-correct set Z which has no intersection with the hyperplane spanned by the affine hull of Y, then $X = Y \cap Z$ is (n, d)-correct. If we speak in the following of sets constructible by *Radon's recipe* we mean sets that can be constructed as mentioned above, also for $d \geq 2$. Radon's recipe even works for $n = 1$ and arbitrary d as long as we interpret $(0, d)$-correctness to mean $(0, 0)$-correctness as we will do from now on.

In the following we introduce a characterization of a class of (n, d)-correct sets, see Definition 2.11. Furthermore, we show that this class coincides with the collection of all sets that are constructible by the recursive application of Radon's recipe. Moreover, we show that this class is a superset of the class of FGPL$_n$-sets. We (Carl de Boor and I) published most of the following together in [SB11].

Before we continue we present the definition of the *affine hull* $\flat(X)$ of $X \subset \mathbb{F}^s$ for some natural number s:

$$\flat(X) := \Big\{ \sum_{x \in X} x w(x) : \sum_{x \in X} w(x) = 1, \quad \# \operatorname{supp} w < \infty \Big\}$$

The affine hull of X is also called flat spanned by X and its dimension,

$$d_X := \dim \flat(X) \ ,$$

is the *affine dimension* of X and equals the dimension of the subspace $\flat(X) - x$ for any $x \in \flat(X)$.

Definition 2.11 [SB11] *Denote by*

$$R_{n,d}$$

the collection of all $X \subset \mathbb{F}^s$ whose affine dimension is bounded by d and for which there is a map

$$\Gamma_{n,d} \longrightarrow X : \quad \alpha \mapsto x_\alpha$$

such that, for each $j \in 1{:}d$ and each $\gamma \in \Gamma_{n-1,j}$, $X = \{x_\alpha : \alpha \in \Gamma_{n,d}\}$ satisfies the following condition.

Condition(γ, j): The affine hull of

$$Y_\gamma^j := \{x_\alpha \in X : \alpha_i = \gamma_i \text{ for } 0 < i \leq j\} \tag{2.6}$$

has only Y_γ^j in common with

$$X_\gamma^j := \{x_\alpha \in X : \alpha_i = \gamma_i \text{ for } 0 < i < j;\ \alpha_j \geq \gamma_j\}\ . \tag{2.7}$$

Note that Condition(γ, j) is satisfied in case there is a hyperplane containing Y_γ^j whose intersection with X_γ^j is Y_γ^j. Note also that there is no assumption that the map $\alpha \mapsto x_\alpha$ be 1-1. Though, this readily follows directly from the Condition(γ, j). Indeed, if $\alpha, \beta \in \Gamma_{n,d}$ with $\alpha \neq \beta$, then there is a smallest j for which $\alpha_j \neq \beta_j$. Let, without loss of generality, $\alpha_j < \beta_j$ then $\gamma := (\alpha_1, \ldots, \alpha_j)$ satisfies $|\gamma| < n$. Hence, by Condition(γ, j), x_α must lie in some flat that does not contain x_β. Therefore $x_\alpha \neq x_\beta$.

Note finally that in Definition 2.11 we have chosen $X \subset \mathbb{F}^s$ for some natural number s with $\dim \flat(X) \leq d$ and not $X \subset \mathbb{F}^d$. Yet it will follow from the definition that, for $n > 0$, necessarily $d_X = d$. In fact, the definition of $R_{n,d}$ is tailor-made for an inductive proof of the following claim.

Theorem 2.12 [SB11] *For $n, d > 0$, $X \subset \mathbb{F}^s$ is in $R_{n,d}$ if and only if X is constructible by recursive application of the Radon recipe. In particular, $X \in R_{n,d}$ is (n, d)-correct for $n, d \geq 0$ and $d_X = d$ for $n > 0$.*

Proof. The proof is by induction on n and d. For $n = 0$ or $d = 0$, any $X \in R_{n,d}$ consists of exactly one point, hence is evidently (n, d)-correct.

Now assume $n, d > 0$ and let $X \in R_{n,d}$. Then X is the disjoint union of the two sets

$$Y := \{x_\alpha : \alpha_1 = 0, \alpha \in \Gamma_{n,d}\} \tag{2.8}$$

and

$$Z := \{x_\alpha : \alpha_1 > 0, \alpha \in \Gamma_{n,d}\}$$

with

$$\flat(Y) \cap X = Y\ ,$$

hence $d_Y \leq d_X - 1 \leq d - 1$, while $d_Z \leq d_X \leq d$. Thus we know that X is obtainable by the recursive application of the Radon recipe once we know that each of Y and Z is so obtainable (or, else, contains just one point), and this we know by induction hypothesis once we show that $Y \in R_{n,d-1}$ and $Z \in R_{n-1,d}$.

For this, we observe that Y satisfies the other requirements of being an $R_{n,d-1}$-set with the assignment

$$y_\alpha \leftarrow x_{(0,\alpha)}\ , \quad \alpha \in \Gamma_{n,d-1}\ ,$$

while Z satisfies the other requirements for being in $R_{n-1,d}$ with the assignment

$$z_\alpha \leftarrow x_{\alpha+\epsilon} \ , \quad \alpha \in \Gamma_{n-1,d} \ ,$$

with $\epsilon := (1, 0, 0, \ldots)$ of the appropriate length. Hence, by induction hypothesis, Y is $(n, d-1)$-correct, and $d_Y = d - 1$. Thus $d_X = d$ and hence $\dim \Pi_n(\flat(X)) = \#\Gamma_{n,d} = \dim \mathbb{F}^X$, where $\Pi_n(\flat(X))$ denotes the space of all polynomials on $\flat(X)$ with degree at most n. Therefore, we know that X is n-correct as soon as we have shown that the linear map

$$\Pi_n(\flat(X)) \longrightarrow \mathbb{F}^X : \quad p \mapsto p|_X \tag{2.9}$$

is 1-1 on $\Pi_n(\flat(X))$, i.e., $p \in \Pi_n(\flat(X))$ and $p|_X = 0$ implies $p = 0$. For this, let $p \in \Pi_n(\flat(X))$ vanish on X. Hence p also vanishes on Y. Therefore, by induction hypothesis, p must vanish on all of $\flat(Y)$. Now, let h be any polynomial of degree 1 on $\flat(X)$ which vanishes on $\flat(Y)$. Then h must be a factor of p, i.e., $p = hq$ for some $q \in \Pi_{<n}(\flat(X))$, cf. Lemma III.4.3. But, by assumption, h fails to vanish anywhere on Z. Hence q must vanish on Z and by induction hypothesis must be identically 0. Thus, $p = 0$ and therefore the linear map (2.9) is 1-1, and because of $\dim \Pi_n(\flat(X)) = \dim \mathbb{F}^X$ it is also invertible. Hence, X is indeed (n, d)-correct, and obtainable by the recursive application of Radon's recipe, thus advancing the induction hypothesis.

Now, let X be an (n, d)-correct set which is obtainable by the recursive application of Radon's recipe. Then $d_X = d$ and $X = Y \cup Z$ must be the disjoint union of two sets, with Y an $(n, d-1)$-correct set and Z an $(n-1, d)$-correct set. Both Y and Z are obtainable by the recursive application of Radon's recipe or else a 1-point set, and $\flat(Y) \cap Z = \emptyset$. By induction hypothesis $Y \in R_{n,d-1}$ and $Z \in R_{n-1,d}$. Thus we can index the elements of X as

$$x_\alpha := \begin{cases} y_{\alpha_{2:d}} \in Y, & \alpha_1 = 0 \\ z_{\alpha-\epsilon} \in Z, & \alpha_1 > 0 \end{cases} , \quad \alpha \in \Gamma_{n,d} \ .$$

Then X, so indexed, satisfies

(a) Condition$(0, 1)$ by the Radon recipe;

(b) Condition$((0, \gamma), j)$ for $1 < j \leq d$ and $\gamma \in \Gamma_{n-1,j-1}$ since that corresponds to the Condition$(\gamma, j-1)$ satisfied by Y;

(c) Condition$(\gamma + \epsilon, j)$ for $1 \leq j \leq d$ and $\gamma \in \Gamma_{n-2,j}$ since that corresponds to the Condition(γ, j) satisfied by Z.

In short, then $X \in R_{n,d}$, thus advancing the induction hypothesis. □

Since any affine map carries flats to flats, the set $R_{n,d}$ is closed under invertible affine maps of $\mathbb{F}^s$. The index set $\Gamma_{n,d}$ as a subset of $\mathbb{F}^d$ is evidently in $R_{n,d}$. Because for any $j \in 1{:}d$ and $\gamma \in \Gamma_{n,j}$ the set

$$\{\alpha \in \Gamma_{n,d} : \alpha_i = \gamma_i, i \in 1{:}j\}$$

lies in the hyperplane $\{x \in \mathbb{F}^d : x_j = \gamma_j\}$ which does not contain any $\beta \in \Gamma_{n,d}$ with $\beta_j > \gamma_j$. Incidentally, the sets $\Gamma_{n,d}$ are the most natural approach for an n-correct set. For $d = 2$ this was already discussed in [Bie03].

Furthermore, any fully generalized principal lattice, see again Definition 2.9, is in $R_{n,d}$.

Theorem 2.13 [SB11] *Any* FGPL$_n$*-set is an* $R_{n,d}$*-set.*

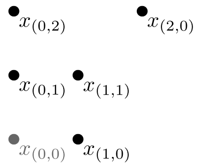
Proof. Let X be an FGPL$_n$-set and let $x_\alpha := x_{(n-|\alpha|,\alpha)}$ for $\alpha \in \Gamma_{n,d}$. Then, as a subset of $\mathbb{F}^d$, its affine dimension is bounded by d. Further, for any $j \in 1{:}d$ and $\gamma \in \Gamma_{n-1,j}$, the hyperplane $H^j_{\gamma_j}$ from Definition 2.9 contains, according to condition (2.4), the set Y^j_γ defined in (2.6). Since $H^j_{\gamma_j}$ contains every x_β with $\beta_j = \gamma_j$, hence also contains $\flat(Y^j_\gamma)$, but, according to (2.5), fails to contain any x_β with $\beta_j > \gamma_j$. Therefore, Condition(γ, j) holds and thus $X \in R_{n,d}$. □

But not every $R_{n,d}$ satisfies the geometric characterization.

Proposition 2.14 [SB11] *There exist* $X \in R_{n,d}$ *with* $X \notin$ GC$_n$.

Proof. Here is a simple $R_{2,2}$-set X that fails to be a GC$_2$-set:

$$x_\alpha = \begin{cases} \alpha & \alpha_1 < 2 \\ (2,2) & \alpha_1 = 2 \end{cases}, \quad \alpha \in \Gamma_{2,2}$$



Indeed, $X \backslash \{x_{(0,0)}\}$ fails to be contained in the union of two straight lines. □

Moreover, not every GC$_n$-set is a $R_{n,d}$-set. This is because for every $R_{n,d}$-set X there exists a hyperplane $\flat(Y)$, with Y equal to (2.8), that contains $\dim \Pi^{d-1}_n = \binom{n+d-1}{d-1}$ points from X. But in [Apo11], an example of a GC$_2$-set in $\mathbb{R}^6$ is given which has no hyperplane containing $\dim \Pi^{d-1}_n$ points. Hence cannot be a $R_{2,6}$-set. This GC$_2$-set was constructed to disprove the multivariate extension [Boo07] of the Gasca–Maeztu conjecture from [GM82], which states that every GC$_n$-set possesses a hyperplane containing $\dim \Pi^{d-1}_n$ points. Hence the conjecture could have been proven by showing that every GC$_n$-set is a $R_{n,d}$-set, but due to the mentioned counter-example of [Apo11] this has become obsolete now. Nevertheless, the conjecture for $d = 2$ is so far proven for $n \leq 4$, see [Bus90].

In the next paragraph we present a concrete recipe for (n, d)-correct sets, which are contained in $R_{n,d}$. Moreover, in Figure 2.3 an example of an $R_{2,3}$-set is depicted.

Concrete recipe for sets contained in $R_{n,d}$

Definition 2.15 [SB11] *Denote by*

$$S_{n,d}$$

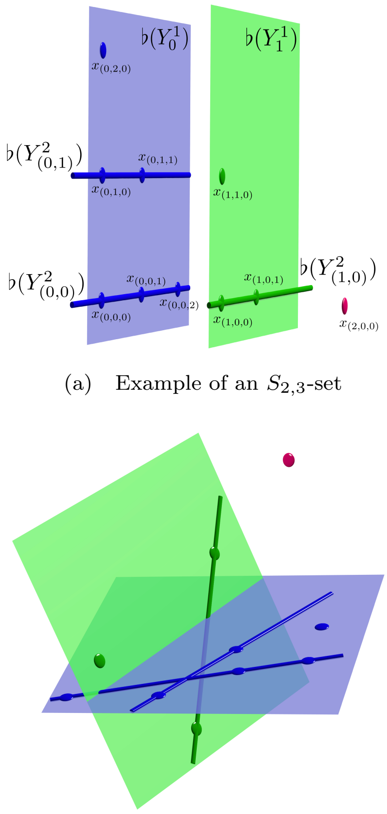
the collection of all subsets X *of* $\mathbb{F}^d$ *that can be so indexed by* $\Gamma_{n,d}$ *that, for every* $j \in 1{:}d$ *and every* $\alpha, \beta \in \Gamma_{n,d}$*, if* $\alpha_i = \beta_i$ *for* $i < j$*, then* $(x_\alpha)_j = (x_\beta)_j$ *if and only if* $\alpha_j = \beta_j$.

Corollary 2.16 [SB11] $S_{n,d} \subset R_{n,d}$. *In particular, any* $S_{n,d}$*-set is* (n, d)*-correct.*

Proof. Let $X \in S_{n,d}$. Since $X \subset \mathbb{F}^d$, $d_X \leq d$. Also, for $j \in 1{:}d$ and $\gamma \in \Gamma_{n-1,j}$, the hyperplane $\{x \in \mathbb{F}^d : x_j = (x_{(\gamma,\beta)})_j\}$ with $\beta := 0 \in \mathbb{F}^{d-j}$ contains $x_\alpha \in X$ with $\alpha_i = \gamma_i$ for $i < j$ and $\alpha_j \geq \gamma_j$ if and only if $\alpha_j = \gamma_j$. Hence Condition(γ, j) holds and thus X is an $R_{n,d}$-set. □

Example for an $S_{2,3}$-set and an $R_{2,3}$-set

In Figure 2.3a we give an example for an $S_{2,3}$-set by depicting $\Gamma_{2,3}$, which we label as an $R_{2,3}$-set. In Figure 2.3b we present an $R_{2,3}$-set, which can be seen as a perturbation of the $S_{2,3}$-set from Figure 2.3a. For a better clarity Figure 2.3b is not labeled, though the labels from Figure 2.3a are still valid here. Next to these two figures we present the conditions for all $X \in R_{2,3}$.



(a) Example of an $S_{2,3}$-set

(b) Example of an $R_{2,3}$-set

$j = 1$

Condition$(0, 1)$:
$\flat(Y_0^1) \cap X_0^1 = Y_0^1$
$Y_0^1 = \{x_{(0,0,0)}, x_{(0,0,1)}, x_{(0,0,2)}, x_{(0,1,0)}, x_{(0,1,1)}, x_{(0,2,0)}\}$
$X_0^1 = \{x_\alpha : \alpha \in \Gamma_{2,3}\}$

Condition$(1, 1)$:
$\flat(Y_1^1) \cap X_1^1 = Y_1^1$
$Y_1^1 = \{x_{(1,0,0)}, x_{(1,0,1)}, x_{(1,1,0)}\}$
$X_1^1 = \{x_{(1,0,0)}, x_{(1,0,1)}, x_{(1,1,0)}, x_{(2,0,0)}\}$

$j = 2$

Condition$((0,0), 2)$:
$\flat(Y_{(0,0)}^2) \cap X_{(0,0)}^2 = Y_{(0,0)}^2$
$Y_{(0,0)}^2 = \{x_{(0,0,0)}, x_{(0,0,1)}, x_{(0,0,2)}\}$
$X_{(0,0)}^2 = \{x_{(0,0,0)}, x_{(0,0,1)}, x_{(0,0,2)}, x_{(0,1,0)}, x_{(0,1,1)}, x_{(0,2,0)}\}$

Condition$((0,1), 2)$:
$\flat(Y_{(0,1)}^2) \cap X_{(0,1)}^2 = Y_{(0,1)}^2$
$Y_{(0,1)}^2 = \{x_{(0,1,0)}, x_{(0,1,1)}\}$
$X_{(0,1)}^2 = \{x_{(0,1,0)}, x_{(0,1,1)}, x_{(0,2,0)}\}$

Condition$((1,0), 2)$:
$\flat(Y_{(1,0)}^2) \cap X_{(1,0)}^2 = Y_{(1,0)}^2$
$Y_{(1,0)}^2 = \{x_{(1,0,0)}, x_{(1,0,1)}\}$
$X_{(1,0)}^2 = \{x_{(1,0,0)}, x_{(1,0,1)}, x_{(1,1,0)}\}$

$j = 3$

Condition$((0,0,0), 3)$:
$\flat(Y_{(0,0,0)}^3) \cap X_{(0,0,0)}^3 = Y_{(0,0,0)}^3$
$Y_{(0,0,0)}^3 = \{x_{(0,0,0)}\}$
$X_{(0,0,0)}^3 = \{x_{(0,0,0)}, x_{(0,0,1)}, x_{(0,0,2)}\}$

Condition$((0,0,1), 3)$:
$\flat(Y_{(0,0,1)}^3) \cap X_{(0,0,1)}^3 = Y_{(0,0,1)}^3$
$Y_{(0,0,1)}^3 = \{x_{(0,0,1)}\}$
$X_{(0,0,1)}^3 = \{x_{(0,0,1)}, x_{(0,0,2)}\}$

Condition$((0,1,0), 3)$:
$\flat(Y_{(0,1,0)}^3) \cap X_{(0,1,0)}^3 = Y_{(0,1,0)}^3$
$Y_{(0,1,0)}^3 = \{x_{(0,1,0)}\}$
$X_{(0,1,0)}^3 = \{x_{(0,1,0)}, x_{(0,1,1)}\}$

Condition$((1,0,0), 3)$:
$\flat(Y_{(1,0,0)}^3) \cap X_{(1,0,0)}^3 = Y_{(1,0,0)}^3$
$Y_{(1,0,0)}^3 = \{x_{(1,0,0)}\}$
$X_{(1,0,0)}^3 = \{x_{(1,0,0)}, x_{(1,0,1)}\}$

Figure 2.3: $S_{2,3}$-set and its perturbation into a $R_{2,3}$-set

In the next paragraph we present a different proof showing that $S_{n,d}$ is n-correct.

Alternative proof for $S_{n,d}$ being n-correct

We give here an alternative proof, showing that $S_{n,d}$ is n-correct. In this paragraph we assume for arbitrary n and d that the index sets $\Gamma_{n,d}$ are lexicographically ordered, where $\boldsymbol{\alpha}_i$ denotes the i-th element of the so ordered set $\Gamma_{n,d}$. Hence, we can express any polynomial $p \in \Pi_n^d$ as

$$p = \sum_{i=1}^{\#\Gamma_{n,d}} a_{\boldsymbol{\alpha}_i} \xi^{\boldsymbol{\alpha}_i} \quad \text{with} \quad a_{\boldsymbol{\alpha}_i} \in \mathbb{F} \,. \tag{2.10}$$

Recall that the set $X = \{x_{\boldsymbol{\alpha}_i} : i \in 1{:}\#\Gamma_{n,d}\}$ is n-correct if for every $f : \mathbb{F}^d \to \mathbb{F}$ there exists a unique polynomial $p \in \Pi_n^d$ such that

$$p(x_{\boldsymbol{\alpha}_i}) = f(x_{\boldsymbol{\alpha}_i}) \tag{2.11}$$

for all $i \in 1{:}\#\Gamma_{n,d}$. Thus, by (2.10) and (2.11) we can set up a system of $\#\Gamma_{n,d}$ equations and know that $X \in S_{n,d}$ is n-correct if the matrix $M_{n,d}^T$ resulting from this system of equations is non-singular, in particular

$$M_{n,d} := \begin{bmatrix} x_{\boldsymbol{\alpha}_1}^{\boldsymbol{\alpha}_1} & x_{\boldsymbol{\alpha}_2}^{\boldsymbol{\alpha}_1} & \cdots & x_{\boldsymbol{\alpha}_{\#\Gamma_{n,d}}}^{\boldsymbol{\alpha}_1} \\ x_{\boldsymbol{\alpha}_1}^{\boldsymbol{\alpha}_2} & x_{\boldsymbol{\alpha}_2}^{\boldsymbol{\alpha}_2} & \cdots & x_{\boldsymbol{\alpha}_{\#\Gamma_{n,d}}}^{\boldsymbol{\alpha}_2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\boldsymbol{\alpha}_1}^{\boldsymbol{\alpha}_{\#\Gamma_{n,d}}} & x_{\boldsymbol{\alpha}_2}^{\boldsymbol{\alpha}_{\#\Gamma_{n,d}}} & \cdots & x_{\boldsymbol{\alpha}_{\#\Gamma_{n,d}}}^{\boldsymbol{\alpha}_{\#\Gamma_{n,d}}} \end{bmatrix}. \tag{2.12}$$

Theorem 2.17 *The matrix $M_{n,d}$ is non-singular if the set $X = \{x_{\boldsymbol{\alpha}_i} : i \in 1{:}\#\Gamma_{n,d}\}$ is indexed as in Definition 2.15. In particular $X \in S_{n,d}$ is n-correct.*

Proof. The proof is by induction on d. For $d = 1$ the assumption obviously is true since we get a Vandermonde matrix with disjoint nodes, which is non-singular. Let $n \geq 0$ be arbitrary and $d \geq 1$. Assume that the assumption holds for d.
For $d + 1$ the matrix $M_{n,d+1}$ can be written as block matrix

$$M_{n,d+1} = \left[\begin{array}{c|c|c} \mathbf{m}_{1,1} & \cdots & \mathbf{m}_{1,n+1} \\ \hline \vdots & \ddots & \vdots \\ \hline \mathbf{m}_{n+1,1} & \cdots & \mathbf{m}_{n+1,n+1} \end{array}\right],$$

with

$$\mathbf{m}_{i,j} = \begin{bmatrix} x_{(j-1,\boldsymbol{\beta}_1)}^{(i-1,\boldsymbol{\alpha}_1)} & \cdots & x_{(j-1,\boldsymbol{\beta}_{\#\Gamma_{n-j+1,d}})}^{(i-1,\boldsymbol{\alpha}_1)} \\ \vdots & & \vdots \\ x_{(j-1,\boldsymbol{\beta}_1)}^{(i-1,\boldsymbol{\alpha}_{\#\Gamma_{n-i+1,d}})} & \cdots & x_{(j-1,\boldsymbol{\beta}_{\#\Gamma_{n-j+1,d}})}^{(i-1,\boldsymbol{\alpha}_{\#\Gamma_{n-i+1,d}})} \end{bmatrix} \quad \text{where} \quad \begin{array}{l} \boldsymbol{\alpha}_k \in \Gamma_{n-i+1,d} \\ \boldsymbol{\beta}_l \in \Gamma_{n-j+1,d} \end{array}.$$

By the recipe of Definition 2.15 $(x_\alpha)_1 = (x_\beta)_1$ if $\alpha_1 = \beta_1$, with $\alpha, \beta \in \Gamma_{n,d+1}$. Thus the submatrices $\mathbf{m}_{i,j}$ can be written as

$$\mathbf{m}_{i,j} = c_j^{i-1} \begin{bmatrix} (x_{(j-1,\boldsymbol{\beta}_1)})_{2:d+1}^{\boldsymbol{\alpha}_1} & \cdots & (x_{(j-1,\boldsymbol{\beta}_{\#\Gamma_{n-j+1,d}})})_{2:d+1}^{\boldsymbol{\alpha}_1} \\ \vdots & & \vdots \\ (x_{(j-1,\boldsymbol{\beta}_1)})_{2:d+1}^{\boldsymbol{\alpha}_{\#\Gamma_{n-i+1,d}}} & \cdots & (x_{(j-1,\boldsymbol{\beta}_{\#\Gamma_{n-j+1,d}})})_{2:d+1}^{\boldsymbol{\alpha}_{\#\Gamma_{n-i+1,d}}} \end{bmatrix} =: c_j^{i-1}\tilde{\mathbf{m}}_{i,j}$$

with $c_j \in \mathbb{C}$ pairwise different. Thus the block matrix $M_{n,d+1}$ reads

$$M_{n,d+1} = \left[\begin{array}{c|c|c} c_1^0\tilde{\mathbf{m}}_{1,1} & \cdots & c_{n+1}^0\tilde{\mathbf{m}}_{1,n+1} \\ \hline \vdots & \ddots & \vdots \\ \hline c_1^n\tilde{\mathbf{m}}_{n+1,1} & \cdots & c_{n+1}^n\tilde{\mathbf{m}}_{n+1,n+1} \end{array}\right].$$

Since for fixed j the blocks $\tilde{\mathbf{m}}_{i,j}$ are linearly dependent on $\tilde{\mathbf{m}}_{r,j}$ for all $r < i$ and the diagonal blocks $\tilde{\mathbf{m}}_{i,i}$ are equal to $M_{n-i+1,d}$, we can transform $M_{n,d+1}$ by Gaussian elimination to the block upper triangular matrix

$$M_{n,d+1} = \left[\begin{array}{c:c:c:c} b_1 M_{n,d} & * & \cdots & * \\ \hdashline 0 & b_2 M_{n-1,d} & * & * \\ \hdashline \vdots & & \ddots & * \\ \hdashline 0 & \cdots & 0 & b_{n+1} M_{0,d} \end{array}\right] .$$

The matrix consisting only of the leading coefficients c_j^{i-1} of the block matrices $\tilde{\mathbf{m}}_{\mathbf{ij}}$ forms a non-singular Vandermonde matrix

$$\begin{bmatrix} c_1^0 & \cdots & c_{n+1}^0 \\ \vdots & & \vdots \\ c_1^n & \cdots & c_{n+1}^n \end{bmatrix} .$$

Therefore, the coefficients b_i for $i = 1, \ldots, n+1$ are unequal to zero. This and the fact that due to our assumption $M_{n,d}$ is non-singular for arbitrary n, yields that the matrix $M_{n,d+1}$ is non-singular, thus advancing the induction hypothesis. □

Chapter III

The Lifting Scheme

This chapter presents the lifting scheme and shows how to construct appropriate filters for it. We start with a brief introduction on wavelets, and explain the connection between the fast wavelet transform and the standard two-channel filter banks. Then we show that the usual two-channel filter bank can be transformed into a more efficient structure – the lifting scheme, see Section 3.3. Own contributions are found in Section 4 where we construct new and shorter filters for the lifting scheme in the two-dimensional case. We also provide a result which yields an extension of the one-dimensional Deslauriers–Dubuc filters to the two-dimensional case.

1 Introduction

In this section we present the wavelet transform. This introduction is just thought as a brief motivation for the rest of this chapter, therefore we refer the interested reader for more details to the books [Dau92], [LMR94], [VK95] and [SN96], where also the main inspiration of what follows is taken from.

The term wavelet as it is known today goes back to Goupilland, Morlet and Grossmann and has its origin in the analysis of seismic signals, see, e.g., the classical paper [GGM84]. The need for wavelets was due to the missing time localization property of the standard Fourier transform, which for $f \in L^2(\mathbb{R})$ is defined as

$$(\mathcal{F}f)(\omega) := \frac{1}{\sqrt{(2\pi)}} \int e^{-i\omega t} f(t) dt \; .$$

The standard Fourier transform suffers from the infinite extent of its basis functions, so spreading the information of f over the whole frequency axis. One ansatz to circumvent this is the windowed Fourier transform

$$(\mathcal{F}_{\mathrm{win}}f)(\omega, \tau) := \int f(t) w(t-\tau) e^{-i\omega t} dt \; ,$$

which uses a window function w that is usually compactly supported or has a fast decay for $|t| \to \infty$ and is of a certain smoothness, for instance a Gaussian. Thus, the windowed Fourier transform has a better time localization than the standard Fourier transform but has the drawback that the size of the window function is constant, so providing only one resolution. This is resolved by the wavelet transform, defined as

$$(\mathcal{W}f)(a,b) := |a|^{-1/2} \int f(t)\psi\left(\frac{t-b}{a}\right) dt \ ,$$

with $a \in \mathbb{R}_+$, $b \in \mathbb{R}$ and $\int \psi = 0$. As can be seen from the formula it is based on the translates and dilates

$$|a|^{-1/2}\psi\left(\frac{t-b}{a}\right) \tag{1.1}$$

of one function ψ, which is called the mother wavelet. Assume for a moment that b is fixed, then for a large a the dilates (1.1) correspond to a very wide window which in turn corresponds to low frequencies, vice versa for small a. Thus when a changes, the dilates (1.1) cover different frequency ranges, whereas changing b yields a different center of localization in time.

Because the wavelet transform is highly redundant for continuous a and b, it is usually only evaluated at the discrete grid

$$(2^j, k2^j) \quad \text{for} \quad j,k \in \mathbb{Z} \ ,$$

so yielding the discrete wavelet transform

$$(\mathcal{W}f)(2^j, k2^j) = 2^{-j/2} \int f(t)\psi(2^{-j}t-k)dt = \langle f, \psi_{j,k}\rangle \ ,$$

with $\psi_{j,k}(t) := 2^{-j/2}\psi(2^{-j}t-k)$. The final breakthrough of the discrete wavelet transform was then provided by the work of Mallat and Meyer, see [Mal89] and [Mey90], by introducing the multiresolution analysis (see Section 2) which makes a fast computation of the wavelet coefficients $\langle f, \psi_{j,k}\rangle$ possible (see Section 2.1). Moreover, it connected the discrete wavelet transform to filter banks, which developed separately from wavelets (see Section 3). In [Swe96] Sweldens presented the lifting scheme filter bank, which on the one hand allows a more efficient implementation of the discrete wavelet transform and additionally gives an idea on how to construct new wavelets (see Section 3.3). We exploit this idea in Section 4 where we construct and verify new filters for the lifting scheme.

2 Multiresolution analysis

Multiresolution analysis goes back to Mallat and Meyer, see [Mal89] and [Mey90]. It provided the key to fast implementations for the discrete wavelet transform and thereby also connected wavelets to filter banks, as we see below. So far we only considered the one-dimensional case. From now on the dimension d is arbitrary. Therefore we need to present the *dilation matrix.*

Definition 2.1 *A matrix $D \in \mathbb{Z}^{d\times d}$ is called dilation matrix if all its eigenvalues have absolute value greater than* 1.

Thus a dilation matrix D is expanding and the subgroup $D\mathbb{Z}^d$ possesses $|\det(D)|$ distinct cosets $D\mathbb{Z}^d + t_i$ for $i \in \{0,\dots,|\det(D)|-1\}$ with $t_i \in \mathbb{Z}^d$ and $t_0 = 0$, see [GM92]. Since in this thesis we only deal with two-channel filter banks it is sufficient to consider $|\det(D)| = 2$, which we will do from now on. For example, the most considered dilation matrices for $d = 2$ with $|\det(D)| = 2$ are

$$D_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ with } t_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad D_2 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \text{ with } t_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \ . \tag{2.1}$$

Their corresponding cosets or so called sublattices are referred to as *Quincunx lattices*, see for instance [VBU05].

We continue with the definition of the multivariate multiresolution analysis, where we mainly follow [LMR94].

Definition 2.2 *A multiresolution analysis of $L^2(\mathbb{R}^d)$ is an ascending sequence of closed subspaces $(V_j)_{j\in\mathbb{Z}}$ of $L^2(\mathbb{R}^d)$*

$$\{0\} \subset \ldots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \ldots \subset L^2(\mathbb{R}^d)$$

with the following properties:

(a) $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R}^d)$,

(b) $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$,

(c) $f(\cdot) \in V_j \Leftrightarrow f(D^j\cdot) \in V_0$,

(d) there exists a function $\phi \in L^2(\mathbb{R}^d)$, called scaling function, whose translates form a Riesz basis of V_0, i.e.,

$$V_0 = \overline{\operatorname{span}\{\phi_{0,k}(\cdot) := \phi(\cdot - k) : k \in \mathbb{Z}^d\}}\ .$$

and

$$A \sum_{k\in\mathbb{Z}^d} c_k^2 \leq \Big\| \sum_{k\in\mathbb{Z}^d} c_k \phi(\cdot - k)\Big\|_{L^2}^2 \leq B \sum_{k\in\mathbb{Z}^d} c_k^2\ ,$$

for all $(c_k)_{k\in\mathbb{Z}^d} \in l^2(\mathbb{Z}^d)$ and with A, B being positive real constants.

By the definition of the multiresolution analysis, more explicitly by condition (c) and (d), it holds that

$$V_j = \overline{\operatorname{span}\{\phi_{j,k}(\cdot) := 2^{-j/2}\phi(D^{-j}\cdot - k) : k \in \mathbb{Z}^d\}}\ .$$

This is why the function ϕ is called *scaling function*, because its scaled versions are needed to generate the space V_j. Moreover, the inclusion $V_0 \subset V_{-1}$ implies the existence of a real valued sequence $(h_k)_{k\in\mathbb{Z}^d}$ such that

$$\phi(x) = \sqrt{2} \sum_{k\in\mathbb{Z}^d} h_k \phi(Dx - k)\ . \tag{2.2}$$

This equation, also referred to as *refinement equation*, is the most important equation in the multiresolution analysis. Firstly, it is the key to the fast wavelet transform and secondly it connects wavelets to filter banks as we see in Section 3. Moreover, we want to point out that the function ϕ is later not explicitly available, but is implicitly given by the sequence $(h_k)_{k\in\mathbb{Z}^d}$. Conditions for the coefficient sequence $(h_k)_{k\in\mathbb{Z}^d}$ which are necessary to induce via the refinement equation (2.2) a ϕ whose translates generate a Riesz basis are discussed in Section 2.2.

Because the ascending sequence $(V_j)_{j\in\mathbb{Z}}$ is nested we can define corresponding complements W_j such that

$$V_j \oplus W_j = V_{j-1}\ , \tag{2.3}$$

where $\oplus$ denotes the direct sum. Note that W_j is not necessarily the orthogonal complement of V_j. The space W_j is also generated by translates and dilates of a function $\psi \in L^2(\mathbb{R}^d)$, called wavelet:

$$W_j = \overline{\operatorname{span}\{\psi_{j,k}(\cdot) := 2^{-j/2}\psi(D^{-j}\cdot - k) : k \in \mathbb{Z}^d\}}\ .$$

Also for the wavelet there exists an analog to the refinement equation (2.2). Because of the relation $W_0 \subset V_{-1}$ there also exists a real valued sequence $(g_k)_{k\in\mathbb{Z}^d}$ such that

$$\psi(x) = \sqrt{2} \sum_{k\in\mathbb{Z}^d} g_k \phi(Dx - k) \ . \tag{2.4}$$

Now, we have the ingredients to present the fast wavelet transform.

2.1 Fast wavelet transform

Let $(V_j)_{j\in\mathbb{Z}}$ be a multiresolution analysis. Moreover, let $(\tilde{V}_j)_{j\in\mathbb{Z}}$ be a second multiresolution analysis with scaling function $\tilde{\phi}$, such that the two multiresolution analyses and their corresponding wavelet spaces W_j and $\tilde{W}_j$ are linked by the biorthogonality condition

$$\langle \phi_{j,l}, \tilde{\phi}_{j,k} \rangle = \delta_{l,k} \ , \qquad \langle \psi_{j,l}, \tilde{\psi}_{j,k} \rangle = \delta_{l,k} \ , \tag{2.5}$$

$$\langle \phi_{j,l}, \tilde{\psi}_{j,k} \rangle = 0 \ , \qquad \langle \psi_{j,l}, \tilde{\phi}_{j,k} \rangle = 0 \ , \tag{2.6}$$

for all $j \in \mathbb{Z}$ and $k, l \in \mathbb{Z}^d$. In this setting the scaling function ϕ is called primal and $\tilde{\phi}$ dual, same applies for the wavelets. It is also possible to choose the primal scaling function ϕ such that its translates generate an orthogonal basis, so implying $\phi = \tilde{\phi}$. But the orthogonal case has some limitations in the design of new sequences $(h_k)_{k\in\mathbb{Z}^d}$, which we enlight in Section 3. For the one-dimensional case the biorthogonal setup (2.5)-(2.6) goes back to [CDF92]. A generalization to the multidimensional case is done, e.g., in [KV99], where we also refer to for more details.

We present now the fast wavelet transform, where we will see that one multiresolution analysis is used to decompose a function and the other to reconstruct it. We start with a function f in V_0. Thus, f can be expressed by the linear combination

$$f = \sum_{k\in\mathbb{Z}^d} \underbrace{\langle f, \tilde{\phi}_{0,k} \rangle}_{=:c_{0,k}} \phi_{0,k} \ .$$

Because of equation (2.3) we can decompose f further into

$$f = \sum_{k\in\mathbb{Z}^d} \underbrace{\langle f, \tilde{\phi}_{J,k} \rangle}_{=c_{J,k}} \phi_{J,k} + \sum_{j=1}^{J} \sum_{k\in\mathbb{Z}^d} \underbrace{\langle f, \tilde{\psi}_{j,k} \rangle}_{=:d_{j,k}} \psi_{j,k} \ . \tag{2.7}$$

Exploiting the dual of equation (2.2) we obtain

$$\tilde{\phi}_{j,k}(x) = \sum_{l\in\mathbb{Z}^d} \tilde{h}_l \tilde{\phi}_{j-1,Dk+l}$$

and therefore it holds that

$$c_{j,k} = \sum_{l\in\mathbb{Z}^d} \tilde{h}_{l-Dk} c_{j-1,k} \ , \tag{2.8}$$

for $j \in 1{:}J$ and $k \in \mathbb{Z}^d$. Similarly, by the dual of equation (2.4) we obtain

$$\tilde{\psi}_{j,k}(x) = \sum_{l\in\mathbb{Z}^d} \tilde{g}_l \tilde{\phi}_{j-1,Dk+l}$$

and hence

$$d_{j,k} = \sum_{l\in\mathbb{Z}^d} \tilde{g}_{l-Dk} c_{j-1,k} \ , \tag{2.9}$$

for $j \in 1{:}J$ and $k \in \mathbb{Z}^d$. So we can recursively decompose f starting with $(c_{0,k} : k \in \mathbb{Z}^d)$ without explicitly determining the inner products $\langle f, \tilde{\psi}_{j,k} \rangle$. This recursive procedure is referred to as the fast wavelet transform. Choosing an *interpolating scaling function*, i.e., a scaling function satisfying

$$\phi(k) = \delta_{0,k} \quad \text{for all} \quad k \in \mathbb{Z}^d \ ,$$

yields

$$c_{0,k} = f(k)$$

for all $k \in \mathbb{Z}^d$.

To reconstruct the function f from the data $\{c_{J,k}, d_{j,k} : j \in 1{:}J,\ k \in \mathbb{Z}^d\}$ we need the primal scaling functions and primal wavelets and recursively obtain $(c_{0,k})_{k\in\mathbb{Z}^d}$ by

$$c_{j,k} = \sum_{l\in\mathbb{Z}^d} h_{k-Dl} c_{j+1,l} + \sum_{l\in\mathbb{Z}^d} g_{k-Dl} d_{j+1,k} \ , \tag{2.10}$$

for $j = J-1, J-2, \ldots, 0$ and $k \in \mathbb{Z}^d$. This can be seen by equating the coefficients of

$$\begin{aligned} \sum_{k\in\mathbb{Z}^d} c_{0,k} &= \sum_{k\in\mathbb{Z}^d} c_{1,k}\phi_{1,k} + \sum_{k\in\mathbb{Z}^d} d_{1,k}\psi_{1,k} \\ &= \sum_{k\in\mathbb{Z}^d} c_{1,k} \sum_{l\in\mathbb{Z}^d} h_l c_{0,l+Dk} + \sum_{k\in\mathbb{Z}^d} d_{1,k} \sum_{l\in\mathbb{Z}^d} g_l c_{0,l+Dk} \ , \end{aligned}$$

where we again exploited the equations (2.2) and (2.4).

In Section 3 we learn that the decomposition or analysis step from $j-1$ to j is nothing else than applying $(c_{j-1,k})_{k\in\mathbb{Z}^d}$ to filters and subsample the result afterwards. Similarly, but the other way round, for the synthesis step from j to $j-1$.

Vanishing moments

In applications, like data compression, it is important that the spaces V_j of the multiresolution analysis contain polynomials of a certain degree $\tilde{N}$. This means that the coefficients $\langle f, \tilde{\psi}_{j,k} \rangle$ from the decomposition (2.7) have to be zero for all $f \in \Pi^d_{\tilde{N}}$, $j \in 1{:}J$ and $k \in \mathbb{Z}^d$. One says a dual wavelet $\tilde{\psi}$ has $\tilde{N}$ dual vanishing moments if

$$\int x^\alpha \tilde{\psi}(x) \mathrm{d}x = 0 \quad \text{for all } \alpha \text{ with } |\alpha| < \tilde{N} \ .$$

This implies that the primal scaling function ϕ is able to reproduce polynomials up to degree $\tilde{N}-1$. In that case ϕ is said to be of order $\tilde{N}$. Similarly choosing a primal wavelet ψ with N primal vanishing moments, the dual scaling function $\tilde{\phi}$ can reproduce polynomials up to degree $N-1$. Moreover, the number of primal vanishing moments is connected to the smoothness of the dual wavelet and vice versa, see, e.g., [JS94, page 20]. In Section 4 we show how to construct primal and dual wavelets with a certain number of vanishing moments.

2.2 Stability and regularity of multivariate scaling functions

In Section 2.1 we saw that in the fast wavelet transform the scaling functions and wavelets were not explicitly used. Instead, it was sufficient to know the corresponding coefficient sequences of the refinement equations, which implicitly defined these functions. But not every sequence

$(h_k)_{k\in\mathbb{Z}^d}$ results in a scaling function ϕ that generates a multiresolution analysis, or in other words its translates do not generate a Riesz basis. If they do then ϕ is called *stable*. Recall the refinement equation

$$\phi(x) = 2\sum_{k\in\mathbb{Z}^d} h_k\phi(Dx-k)\ , \tag{2.11}$$

where in this section we choose for convenience, without loss of generality, a different factor as in equation (2.2). In Section 4 we are going to construct new sequences $(h_k)_{k\in\mathbb{Z}^d}$. Therefore we provide in this section conditions for a sequence $(h_k)_{k\in\mathbb{Z}^d}$ to define via the refinement equation (2.11) a stable scaling function ϕ.

From now on we assume that the sequence $(h_k)_{k\in\mathbb{Z}^d}$ has only finitely many non-zero coefficients h_k and that $\sum_k h_k = 1$. To check stability and other properties like regularity of a scaling function ϕ one has to investigate eigenvalues of a linear operator. We are going to motivate this operator by the *cascade algorithm*, where we mainly follow [LLS98].

The cascade algorithm

The cascade algorithm can be used to iteratively compute the scaling function ϕ by the coefficient sequence $(h_k)_{k\in\mathbb{Z}^d}$ via the following iteration over $j\in\mathbb{Z}_+$:

$$\phi_j(x) := \sum_{k\in\mathbb{Z}^d} 2h_k\phi_{j-1}(Dx-k)\ , \tag{2.12}$$

where one starts with a compactly supported function ϕ_0. To show the L^2-convergence of the cascade algorithm one uses the *autocorrelation* of the scaling function ϕ, which for any $\phi\in L^2(\mathbb{R}^d)$ is defined as

$$\phi^{(\text{au})}(k) := \int \phi(x)\phi(x-k)\mathrm{d}x \quad \text{for} \quad k\in\mathbb{Z}^d\ . \tag{2.13}$$

Then equation (2.12) together with equation (2.13) yields

$$\phi_j^{(\text{au})}(k) = \sum_{l\in\mathbb{Z}^d} 2h_{Dk-l}^{(\text{au})}\phi_{j-1}^{(\text{au})}(l)\ , \tag{2.14}$$

with

$$h_k^{(\text{au})} := \sum_{l\in\mathbb{Z}^d} h_{k-l}h_{-l} \quad \text{for} \quad k\in\mathbb{Z}^d$$

being the autocorrelation of the sequence $(h_k)_{k\in\mathbb{Z}^d}$. Moreover, if ϕ satisfies equation (2.11) then $\phi^{(\text{au})}$ satisfies the refinement equation

$$\phi^{(\text{au})}(x) = \sum_k h_k^{(\text{au})}\phi^{(\text{au})}(Dx-k)\ .$$

Defining the linear transformation $T_{h^{(\text{au})}} : l^2(\mathbb{Z}^d) \to l^2(\mathbb{Z}^d)$ as

$$(T_{h^{(\text{au})}}b)_k := \sum_{l\in\mathbb{Z}^d} 2h_{Dk-l}^{(\text{au})}b_l \quad \text{for} \quad b\in l^2(\mathbb{Z}^d) \ \text{ and } \ k\in\mathbb{Z}^d\ ,$$

we can write equation (2.14) as

$$\phi_j^{(\text{au})} = T_{h^{(\text{au})}}\phi_{j-1}^{(\text{au})}\ . \tag{2.15}$$

It can be shown that the iteration (2.15) and so the cascade algorithm converges if $\lambda = 1$ is a simple eigenvalue of $T_{h^{(\text{au})}}$ and $|\lambda| < 1$ for all the other eigenvalues, see [LLS98, Theorem 2.2].

Hence convergence of the cascade algorithm becomes the convergence of the power method in (2.15), see also [SN96, page 234]. To actually compute the eigenvalues, one restricts the operator $T_{h^{(\mathrm{au})}}$ to an invariant support set Ω:

Definition 2.3 [LLS97] *Let $D \in \mathbb{Z}^{d\times d}$ be a dilation matrix. Then $\Omega \subset \mathbb{Z}^d$ is called an invariant support set for the transition operator $T_{h^{(au)}}$ if*

(a) Ω is finite,

(b) for all sequences b with support in Ω, the support of $T_{h^{(au)}}b$ is also in Ω,

(c) the support of every finitely supported eigenvector of $T_{h^{(au)}}$ corresponding to a nonzero eigenvalue is contained in Ω.

For $T_{h^{(\mathrm{au})}}$ such an invariant support set Ω always exists, see [LLS97]. In this paper it is also explained how to construct such an invariant support set Ω. The restriction of $T_{h^{(\mathrm{au})}}$ to Ω is then represented by the matrix

$$T_{h^{(\mathrm{au})}} := \left(2h^{(\mathrm{au})}_{Dk-l}\right)_{k,l\in\Omega} , \tag{2.16}$$

where we use the same symbol.

But the convergence of the cascade algorithm is not sufficient for ϕ being stable, however the matrix $T_{h^{(\mathrm{au})}}$ also holds the key to check stability of ϕ.

Stability

Theorem 2.4 [LLS97] *Suppose $(h_k)_{k\in\mathbb{Z}^d}$ is a finitely supported sequence satisfying $\sum_k h_k = 1$, and $\phi \in L^2(\mathbb{R}^d)$ is given by (2.11). Then ϕ is stable if and only if*

(a) 1 is a simple eigenvalue of the matrix $T_{h^{(au)}}$ defined by (2.16)

(b) the Fourier transform of the eigenvector v corresponding to the eigenvalue 1 does not vanish, where the Fourier transform of v is defined by

$$\hat{v}_k := \sum_{l\in\Omega} v_l e^{-ikl} \quad \textit{for all} \quad k \in \mathbb{Z}^d .$$

Regularity

The *smoothness order* or so called *Sobolev regularity* of a function $f \in L^2(\mathbb{R}^d)$ is defined by

$$\nu(f) := \sup\{\nu : f \in W_2^\nu(\mathbb{R}^d)\} ,$$

where $W_2^\nu(\mathbb{R}^d)$ is the Sobolev space of all functions $f \in L^2(\mathbb{R}^d)$ that satisfy

$$\int |\hat{f}(x)|^2(1+|x|^2)^\nu \mathrm{d}x < \infty ,$$

with $\hat{f}$ being the Fourier transform of f. The smoothness order $\nu(f)$ of a function f states how often f can be weakly differentiated. We can determine the smoothness order $\nu(\phi)$ of a scaling function ϕ also by investigating its constructing sequence $(h_k)_{k\in\mathbb{Z}^d}$ and the eigenvalues of the corresponding matrix $T_{h^{(\mathrm{au})}}$ defined in equation (2.16).

Assume that the dilation matrix $D \in \mathbb{Z}^{d\times d}$ is similar to a diagonal matrix and possesses the eigenvalues $\eta_1, \ldots, \eta_d$ which are equal in modulus, i.e., $|\eta_1| = \cdots = |\eta_d|$. Moreover, let the scaling function ϕ be stable and able to reproduce polynomials up to degree $N-1$. Under these assumptions it was shown in [Jia99] and [JZ99] that

$$\nu(\phi) = -\frac{d \log_2 \rho_N}{2} \ ,$$

with

$$\rho_N := \max \big\{ |x| : x \in \sigma(T_{h^{(\mathrm{au})}}) \cap \{\eta^{-\alpha} : |\alpha| < 2N\} \big\},$$

where η denotes here the tuple $(\eta_1, \dots, \eta_d)$ and $\sigma(T_{h^{(\mathrm{au})}})$ the spectrum of the matrix $T_{h^{(\mathrm{au})}}$.

The dilation matrices D_1 and D_2 from equation (2.1) match the assumptions mentioned above with $\eta = (\sqrt{2}, -\sqrt{2})$ and $\eta = (1+i, 1-i)$, respectively. Since we only treat these cases in this thesis, these restrictive assumptions on the dilation matrix D are acceptable. Nevertheless in [CGV99] the smoothness order $\nu(\phi)$ is derived for arbitrary dilation matrices D. Moreover, an efficient algorithm computing $\nu(\phi)$ for sequences $(h_k)_{k \in \mathbb{Z}^d}$ that are symmetric, i.e., $h_k = h_{-k}$ for all $k \in \mathbb{Z}^d$, is discussed in [Han03].

3 Two-channel filter banks

Before we discuss the two-channel filter banks and link them to the fast wavelet transform, we present some preliminaries.

3.1 Preliminaries

We start with signals:

Signals

In this thesis we deal with discrete signals only, where a discrete signal $\mathbf{x}$ is just a real valued sequence

$$\mathbf{x} := (x_k \in \mathbb{R} : k \in K \subset \mathbb{Z}^d) = (x_k)_{k \in K} \ .$$

The set K can be finite. For instance an image obtained from a digital camera is a two-dimensional signal with finite K.

The z-transform of a signal $\mathbf{x} \in \mathbb{R}^K$ is defined by

$$\mathbf{x}(z) := \sum_{k \in K} x_k z^{-k} \ .$$

Below we are going to consider sequences which stem from multivariate polynomials $q \in \Pi_n^d$, where we use for any function $f : \mathbb{Z}^d \to \mathbb{R}^d$ the notation

$$q\big(f(\mathbb{Z}^d)\big) := \big(q(f(k))\big)_{k \in \mathbb{Z}^d} \ . \tag{3.1}$$

Up- and down-sampling

A basic operation in filter banks is up- and down-sampling of a signal, where up- and down-sampling is always connected to a dilation matrix D. Therefore the symbol for downsampling is chosen as $(\downarrow D)$ and similarly for upsampling as $(\uparrow D)$. Up- and down-sampling on a signal $\mathbf{x}$ is defined as

$$((\uparrow D)\mathbf{x})_k := \begin{cases} x_{D^{-1}k} & \text{if } D^{-1}k \in \mathbb{Z}^d \\ 0 & \text{else} \end{cases} \qquad \text{for} \quad k \in \mathbb{Z}^d$$

and

$$((\downarrow D)\mathbf{x})_k := x_{Dk} \quad \text{for} \quad k \in \mathbb{Z}^d \, .$$

So in the one-dimensional case downsampling with respect to $D = 2$ means nothing else than omitting every second element of the signal and upsampling just stretches the signal by pasting a zero between every element of the signal. In Figure 3.1 we demonstrate what downsampling looks like in the two-dimensional case when using the two dilation matrices D_1 and D_2 from equation (2.1). In Figure 3.1a the signal which is subject to downsampling is depicted. In Figure 3.1b the dilation matrix D_1 is used and 3.1a is downsampled twice, whereas in Figure 3.1c the dilation matrix D_2 is used and we sampled 3.1a down 8 times. Note that from Figure 3.1 it also can be seen that $D_1^2 = 2I$ and $D_2^8 = 16I$.

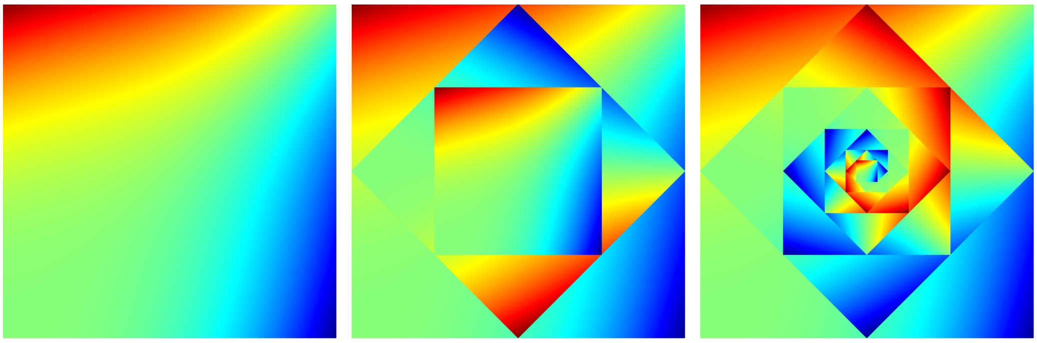

(a) Signal subject to downsampling (b) Sampled down by $D = D_1$ (c) Sampled down by $D = D_2$

Figure 3.1: Downsampling in the Quincunx-case using the dilation matrices D_1 and D_2

In the z-domain upsampling is defined by

$$(\uparrow D)\mathbf{x}(z) := \mathbf{x}(z^D)$$

and downsampling by

$$(\downarrow D)\mathbf{x}(z) := \frac{1}{2}\left(\mathbf{x}\left(z^{D^{-1}}\right) + \mathbf{x}\left(-z^{D^{-1}}\right)\right) \, .$$

with $z^D := [z^{d_1}, \dots, z^{d_n}]^T$ where d_i denotes here the i-th column of D. For more details, see, e.g., [VA91] or [Vai93].

Shifting

Below we also need to shift a signal by $t \in \mathbb{Z}^d$. The action of such a shift on a signal $\mathbf{x}$ is defined as

$$\left((\overrightarrow{t})\mathbf{x}\right)_k := x_{k-t} \quad \text{and} \quad \left((\overleftarrow{t})\mathbf{x}\right)_k := x_{k+t} \quad \text{for} \quad k \in \mathbb{Z}^d \, .$$

In the z-domain the shift is realized by just multiplying z^{-t} or z^t to the z-transform of the signal, i.e.,

$$(\overrightarrow{t})\mathbf{x}(z) := \mathbf{x}(z)z^{-t} \quad \text{and} \quad (\overleftarrow{t})\mathbf{x}(z) := \mathbf{x}(z)z^{t} \, .$$

Filters

A filter H is an operator which maps signals to signals and is defined by a real valued sequence $(h_k \in \mathbb{R} : k \in \mathbb{Z}^d)$. This sequence is also referred to as *impulse response sequence*. In this thesis we only consider *finite impulse response filters*, called FIR-filters. This means that only finitely many coefficients h_k are non-zero. So whenever we speak about filters we mean FIR-filters. The coefficients h_k are also referred to as *filter coefficients* or *filter taps*. The adjoint H^* of a filter H is given by $(h_{-k} \in \mathbb{R} : k \in \mathbb{Z}^d)$.

The action of a filter H on a signal $\mathbf{x} \in \mathbb{R}^K$ is defined by the convolution of the impulse response sequence of the filter and the signal itself

$$(H\mathbf{x})_k := \sum_{l \in K} h_{k-l} x_l \ .$$

If K is finite one has to extend the signal $\mathbf{x}$ outside K. One choice is to continue the signal with 0. This is referred to as zero-padding.

The filter H in the z-domain is defined by the z-transform of its impulse response sequence $H(z) = \sum_i h_i z^{-i}$. The adjoint then equals $H^*(z) = H(z^{-1})$

A filter H is called *interpolating* if its impulse response sequence satisfies $h_{Dk} = \delta_{0,k}$. In the one-dimensional case, this means that the impulse response sequence of the filter is 0 in all even locations except for the origin. Therefore such filters are also called half band filters. Applying a half band filter on a signal $\mathbf{x}$ that was upsampled results in a signal that stays unchanged at the positions Dk while at the positions $Dk + t_1$ it is a linear combination of the values at Dk. In the z-domain we can express a half band filter H by

$$H(z) = 1 + z^{t_1} H_o(z^D) \ , \tag{3.2}$$

for some filter H_o.

An important class of filters in the area of image processing are the *linear phase* filters with real frequency response because they produce less visual artifacts than non-linear phase filters, we refer for more details to [Lim90, page 196]. A linear phase filter H is called a *zero phase* filter if and only if

$$h_k = h_{-k} \quad \text{for all} \quad k \in \mathbb{Z}^d \ ,$$

see, e.g., [Vai93, page 553]. Thus, the impulse response sequence of such a zero phase filter is symmetric with respect to the origin. Therefore such filters are also called *symmetric*. Another good property of symmetric filters is that in numerical implementation the number of computational operations can be halved, see, e.g., [VK95, page 361].

3.2 Standard two-channel filter bank

So with the above presented operations it can easily be seen that the equations (2.8) and (2.9) used for decomposing the signal $(c_{j,k} : k \in \mathbb{Z}^d)$ mean nothing else than applying the signal to the adjoint of the filters $\tilde{H}$ and $\tilde{G}$, respectively and downsample the result afterwards with $(\downarrow D)$, yielding the two signals

$$\mathbf{c}_{j+1} := (c_{j+1,k} : k \in \mathbb{Z}^d) \quad \text{and} \quad \mathbf{d}_{j+1} := (d_{j+1,k} : k \in \mathbb{Z}^d) \ . \tag{3.3}$$

The reconstruction equation (2.10) states that the signals (3.3) are first upsampled by $(\uparrow D)$ and then are applied to the filters H and G, respectively. Afterwards, the results are added, yielding again $\mathbf{c}_j$. All this is depicted in Figure 3.2, where the standard two-channel filter bank is presented. The left side is called analysis part and the right side synthesis part. Since the output on the right side equals the input on the left this is also referred to as a *perfect reconstruction* filter bank. Similar to Section 2.1 this filter bank is said to have N primal and $\tilde{N}$ dual vanishing moments if

$$\begin{aligned} (\downarrow D) G q(\mathbb{Z}^d) &= 0 \quad \text{for all} \quad q \in \Pi_{N-1}^d \quad \text{and} \\ (\downarrow D) \tilde{G} q(\mathbb{Z}^d) &= 0 \quad \text{for all} \quad q \in \Pi_{\tilde{N}-1}^d \ . \end{aligned}$$

The structure depicted in Figure 3.2 was already introduced in the 1980s, cf. [Min85] and [SB86]. In these papers the filters where chosen such that $\tilde{H} = H$ and $\tilde{G} = G$. Choosing the same filters in the synthesis part as in the analysis part yields orthogonal wavelets and scaling functions. But this choice has some drawbacks, the most severe is that except of the trivial choice of the Haar-wavelet there exist no orthogonal two-channel filter bank that has linear phase FIR filters with real coefficients, cf. [VK95, Proposition 3.12]. Biorthogonal filter banks with linear phase filters were then investigated in [VLG89] and [NV89]. For more details on filter banks and a more signal theoretic perspective to them we refer to [Vai93], [VK95] and [SN96].

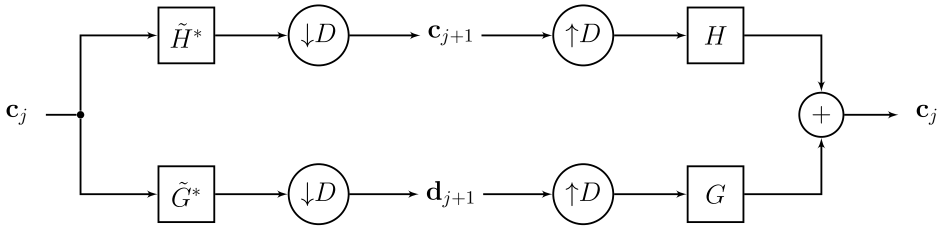


Figure 3.2: Analysis and synthesis part of the standard two-channel filter bank

Downsampling a signal $\mathbf{x}$ with a dilation matrix D that has determinant 2, means that half of the signal that is subject to downsampling is discarded afterwards and only the components x_{Dk} are kept. Following the one-dimensional case with $D = 2$ we call these components *even* and the components x_{Dk+t_1} that are discarded *odd*. Hence, the standard two-channel filter bank is not very efficient, because the whole signal is applied to the filter and then half of the result is thrown away. A more efficient way is to make use of the *polyphase representation*, which we are going to present in the following.

Polyphase representation

We start by explaining the word *polyphase*, where we follow the descriptive explanation of [SN96, page 114]. As stated above downsampling splits the signal in an even and an odd phase, where only the even phase is kept. So it is natural to follow the even phase $(x_{Dk})_{k\in\mathbb{Z}^d}$ and the odd phase $(x_{Dk+t_1})_{k\in\mathbb{Z}^d}$ of the signal $\mathbf{x}$ as they go through the filter bank. As it turns out below they are acted on by the two phases H_e and H_o of the filter H. The word *phase* is applied because the filter coefficients h_{Dk} of the even filter H_e are phase shifted to the filter coefficients h_{Dk+t_1} from the odd filter H_o. What follows in this paragraph is partially borrowed from [Sto09].

For the polyphase representation we need to present the so-called *noble identities*, which allow interchanging the action of sampling and filtering. Let H be a filter. Then the first noble identity is given by

$$H(z)(\downarrow D) = (\downarrow D) H(z^D)$$

and the second by

$$(\uparrow D) H(z) = H(z^D)(\uparrow D) \ ,$$

where $H(z)$ and $H(z^D)$ are considered to be operators here. For more details on the noble identities see [Vai93, page 604].

Consider a filter H. Then we can write its z-transform as

$$\begin{aligned} H(z) &= \sum_k h_k z^{-k} && (3.4)\\ &= \sum_k h_{Dk} z^{-Dk} + \sum_k h_{Dk+t_1} z^{-(Dk+t_1)} && (3.5)\\ &= \underbrace{\sum_k h_{Dk} z^{-Dk}}_{=:H_e(z^D)} + z^{-t_1} \underbrace{\sum_k h_{Dk+t_1} z^{-Dk}}_{=:H_o(z^D)} \; . && (3.6) \end{aligned}$$

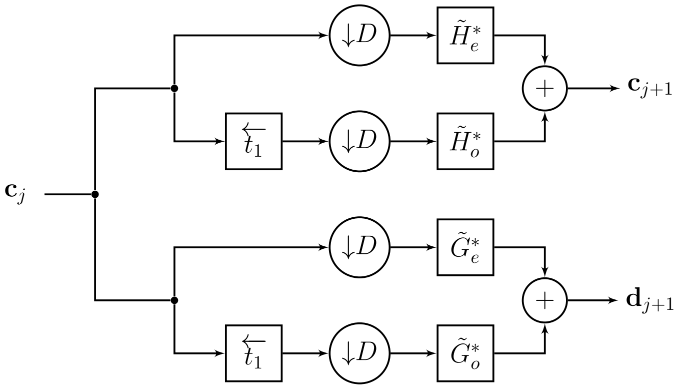
Hence by (3.6) and the noble identities, the following applies

$$\begin{aligned} (\downarrow D)H(z) =& (\downarrow D)H_e(z^D) + (\downarrow D)H_o(z^D)(\overrightarrow{t_1})\\ =& H_e(z)(\downarrow D) + H_o(z)(\downarrow D)(\overrightarrow{t_1}) \end{aligned} \qquad (3.7)$$

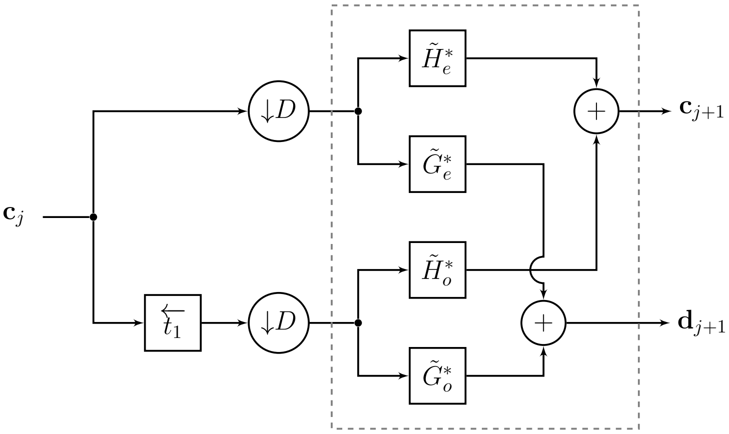
and

$$\begin{aligned} H(z)(\uparrow D) =& H_e(z^D)(\uparrow D) + z^{-t_1} H_o(z^D)(\uparrow D)\\ =& (\uparrow D)H_e(z) + z^{-t_1}(\uparrow D)H_o(z) \; . \end{aligned} \qquad (3.8)$$

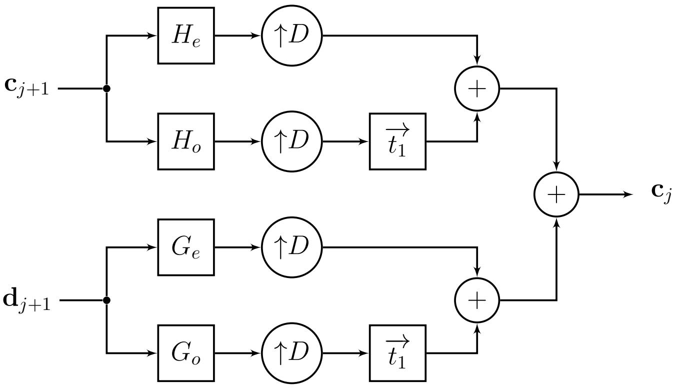
So by equation (3.7) we can transform the analysis part of the standard two-channel filter bank which is depicted in Figure 3.2 to:



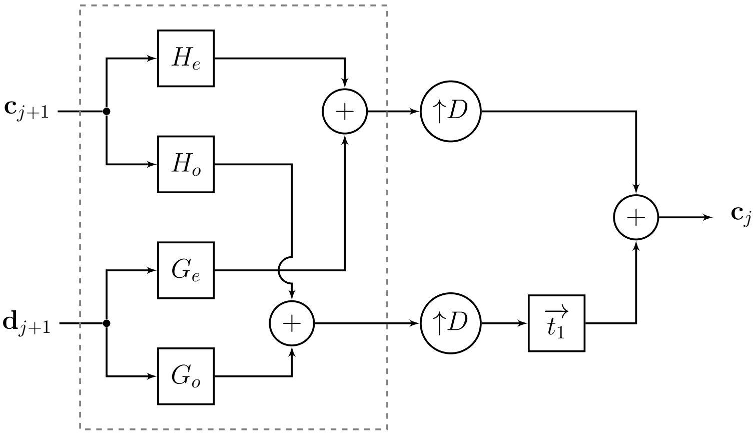
This can evidently be simplified to:



Similarly, by equation (3.8) the synthesis part of the filter bank from Figure 3.2 can be transformed to



and also be simplified to:



If one considers the gray dashed boxes, from the figures above, as one object with two inputs and two outputs, as it is depicted below in Figure 3.3 and furthermore considers the z-transforms of the inputs to be written in a vector, e.g., $[\mathbf{a}(z), \mathbf{b}(z)]^T$, the action of the gray dashed boxes can be expressed by the so called *polyphase matrices*

$$\tilde{\boldsymbol{\mathcal{P}}}^*(z) := \begin{bmatrix} \tilde{H}_e^*(z) & \tilde{H}_o^*(z) \\ \tilde{G}_e^*(z) & \tilde{G}_o^*(z) \end{bmatrix} \quad \text{and} \quad \boldsymbol{\mathcal{P}}(z) := \begin{bmatrix} H_e(z) & G_e(z) \\ H_o(z) & G_o(z) \end{bmatrix} .$$

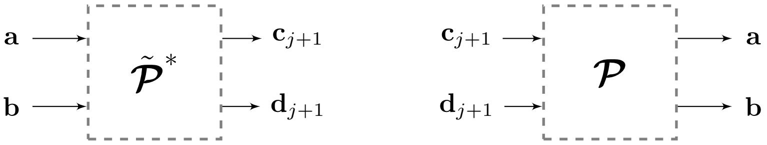


Figure 3.3: Polyphase blocks

In the next section we present the lifting scheme. There we show – by using the polyphase matrices $\tilde{\boldsymbol{\mathcal{P}}}^*(z)$ and $\boldsymbol{\mathcal{P}}(z)$ – how the filters of the lifting scheme have to be chosen such that the lifting scheme acts as the standard two-channel filter bank.

3.3 Lifting scheme

In Figure 3.4 the lifting scheme is depicted. As already mentioned in the introduction of this chapter the lifting scheme goes back to [Swe96]. In [KS00] a generalization of the lifting scheme to arbitrary dimensions and grids is introduced and an idea is presented on how to design new filters for the lifting scheme. We explain this in detail below.

We start by revealing that the lifting scheme can be transformed into the standard two-channel filter bank.

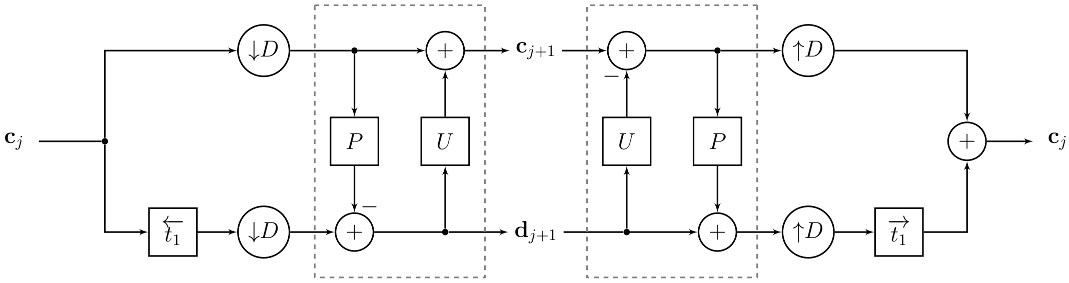


Figure 3.4: The lifting scheme

From Figure 3.4, which depicts the lifting scheme, it can be seen that the polyphase matrix representing the polyphase block of the analysis part equals

$$\tilde{\boldsymbol{\mathcal{P}}}^*(z) = \begin{bmatrix} 1 & U(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -P(z) & 1 \end{bmatrix} = \begin{bmatrix} 1 - U(z)P(z) & U(z) \\ -P(z) & 1 \end{bmatrix} . \tag{3.9}$$

Similarly, the polyphase matrix for the synthesis part of the lifting scheme reads

$$\boldsymbol{\mathcal{P}}(z) = \begin{bmatrix} 1 & 0 \\ P(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -U(z) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -U(z) \\ P(z) & -P(z)U(z) + 1 \end{bmatrix} . \tag{3.10}$$

Hence, if we choose P and U such that

$$\begin{bmatrix} \tilde{H}_e^*(z) & \tilde{H}_o^*(z) \\ \tilde{G}_e^*(z) & \tilde{G}_o^*(z) \end{bmatrix} = \begin{bmatrix} 1 - U(z)P(z) & U(z) \\ -P(z) & 1 \end{bmatrix} \tag{3.11}$$

and

$$\begin{bmatrix} H_e(z) & G_e(z) \\ H_o(z) & G_o(z) \end{bmatrix} = \begin{bmatrix} 1 & -U(z) \\ P(z) & -P(z)U(z) + 1 \end{bmatrix} \tag{3.12}$$

hold, the action of the analysis and synthesis part of the lifting scheme is equivalent to the corresponding parts of the two-channel filter bank discussed in Section 3.2.

Moreover, for any given filter pair P and U we can define new filters $\tilde{H}, \tilde{G}, H$ and G for the standard two channel filter bank by using the latter equations (3.11), (3.12) and equation (3.6), via

$$\tilde{H}(z) = 1 - U(z^{-D})P(z^{-D}) + z^{-t_1}U(z^{-D}) \; , \tag{3.13}$$

$$\tilde{G}(z) = -P(z^{-D}) + z^{-t_1} \; , \tag{3.14}$$

$$H(z) = 1 + z^{-t_1}P(z^D) \; , \tag{3.15}$$

$$G(z) = -U(z^D) + z^{-t_1}(1 - P(z^D)U(z^D)) \; . \tag{3.16}$$

Thus, by the latter equations (3.13)-(3.16) the lifting scheme can be transformed into the standard two-channel filter bank. Note that the filter H^* so constructed is directly a half band filter, cf. equation (3.2).

Now, we explain the basics of the d-dimensional lifting scheme, which was introduced in [KS00]. Like the two-channel filter bank from the previous section, the lifting scheme also consists of two parts, the analysis part on the left side and the synthesis part on the right side. As can be seen from Figure 3.4 the analysis part consists of three actions: *split*, *predict* and *update*. First, by downsampling and shifting, the signal is split into an even and an odd phase. Then the filter P tries to predict the odd phase from the even phase, where the predict filter P is chosen such that it yields exact prediction for polynomial sequences. This is explained in more detail below. Then the prediction is substracted from the odd phase, yielding the lower output. The lower output is then applied to the update filter U and then added to the even phase yielding the upper output. The update step is applied to achieve that the signal at the upper output has the same average as the signal at the input. How to exactly choose the filters P and U is explained below and in Section 4.

Since in the synthesis part of the lifting scheme all operations appear in reversed order to the analysis part, this filter bank is evidently a perfect reconstruction filter bank. This can also be seen by multiplying the two polyphase matrices from equation (3.9) and (3.10), so obtaining $\tilde{\boldsymbol{\mathcal{P}}}^*\boldsymbol{\mathcal{P}} = I$. This perfect reconstruction property directly implies biorthogonality, see [VK95, page 119] or [KS00].

Neville filters

In this section we explain how to choose the filters P and U such that the corresponding standard two channel filter bank has $\tilde{N}$ dual and N primal vanishing moments, respectively. The lifting scheme allows to construct these properties separately, where Neville filters play a central role as we learn below.

Definition 3.1 [KS00] *A filter P is called Neville filter of order N with shift $\tau \in \mathbb{R}^d$ if*

$$Pq(\mathbb{Z}^d) = q(\mathbb{Z}^d + \tau) \quad \textit{for all} \quad q \in \Pi^d_{N-1} \; . \tag{3.17}$$

Thus a polynomial sequence applied to a Neville filter of order N and shift τ just results in a sequence evaluated from the same polynomial but on the lattice shifted by τ, see again equation (3.1).

We start by determining filters that yield $\tilde{N}$ dual vanishing moments. Therefore we have to determine the filter P such that

$$(\downarrow D)\tilde{G}q(\mathbb{Z}^d) = 0 \quad \text{for all} \quad q \in \Pi^d_{\tilde{N}-1} \ ,$$

with $\tilde{G}$ defined as in equation (3.14). This can either be resolved using equation (3.7) together with equation (3.11) or by determining the filter P so that the lower output of the analysis part is zero for all polynomial sequences of order $< \tilde{N}$ that are applied to it. This is equivalent to finding a predict filter P that yields exact prediction for polynomial sequences $q(\mathbb{Z}^d)$ of order $< \tilde{N}$, i.e., for all $q \in \Pi^d_{\tilde{N}-1}$. Thus, applying an arbitrary polynomial sequence $q(\mathbb{Z}^d)$ of order $\tilde{N}$ to the analysis part of the lifting scheme results in

$$q(D\mathbb{Z}^d + t_1) - Pq(D\mathbb{Z}^d) \tag{3.18}$$

at the lower output. Let P be a Neville filter of order $\tilde{N}$ and shift τ, then by Definition 3.1

$$Pq(D\mathbb{Z}^d) = q(D\mathbb{Z}^d + D\tau) \ .$$

Hence equation (3.18) – and thus the lower output of the analysis part of the lifting scheme – is zero for all polynomial sequences of order $\tilde{N}$ if P is a Neville filter of order $\tilde{N}$ and shift $\tau = D^{-1}t_1$.

Now we explain how the filter U has to be chosen such that we get N primal vanishing moments with $N \leq \tilde{N}$. In order to do this we need the following proposition:

Proposition 3.2 [KS00] *Let P be a Neville filter of order N and shift τ. Then the adjoint filter P^* is a Neville filter of the same order N but shift $-\tau$.*

To obtain N primal vanishing moments it must hold for all $q \in \Pi^d_{N-1}$ that

$$(\downarrow D)Gq(\mathbb{Z}^d) = 0 \ , \tag{3.19}$$

with G defined as in equation (3.16). By equation (3.7) this is equivalent to

$$\left(G_e(\downarrow D) + G_o(\downarrow D)(\overrightarrow{t_1})\right) q(\mathbb{Z}^d) = 0 \ ,$$

which in turn by equation (3.12) is equal to

$$\left(-U^*(\downarrow D) + (-U^*P^* + 1)(\downarrow D)(\overrightarrow{t_1})\right) q(\mathbb{Z}^d) = 0 \ .$$

Hence we get

$$-U^*q(D\mathbb{Z}^d) + (-U^*P^* + 1)q(D\mathbb{Z}^d + t_1) = 0 \ .$$

Let P be a Neville filter of order $\tilde{N} \geq N$, then because of Proposition 3.2, equation (3.19) is equivalent to

$$2U^*q(D\mathbb{Z}^d) = q(D\mathbb{Z}^d + t_1) \ .$$

This yields the following theorem:

Theorem 3.3 [KS00] *Let $N \leq \tilde{N}$ and let P be a Neville filter of order $\tilde{N}$ and shift $\tau = D^{-1}t_1$. Furthermore, let V be a Neville filter of order N and shift τ and choose $U = \frac{1}{2}V^*$. Then with theses filters P and U the lifting scheme possesses N primal and $\tilde{N}$ dual vanishing moments.*

4 Construction and verification of new Neville filters

In this section we explicitly explain how to construct Neville filters of certain order N and shift τ for the lifting scheme. Furthermore, we provide new Neville filters for $d = 2$, which need considerably fewer filter coefficients than the filters derived in [KS00]. Moreover, some filters from [KS00] do not yield stable scaling functions, in contrast all our filters do. Furthermore, we present in Section 4.2 configurations of points that yield Neville filters with a minimal number of filter coefficients.

We start with the proposition from [KS00] which holds the key to construct appropriate Neville filters.

Proposition 4.1 [KS00] *A filter P is a Neville filter of order N and shift τ if and only if*

$$\sum_{k\in\mathbb{Z}^d} p_{-k}k^\alpha = \tau^\alpha \quad \textit{for} \quad |\alpha| < N\ . \tag{4.1}$$

Proof. [KS00] Let P be a Neville filter of order N and shift τ. Substituting monomials x^α with $|\alpha| < N$ in equation (3.17) yields

$$\sum_{k\in\mathbb{Z}^d} p_{-k}(l+k)^\alpha = (l+\tau)^\alpha \quad \text{for all} \quad |\alpha| < N\ .$$

Given that polynomial spaces are shift invariant it suffices to consider $l = 0$. Hence

$$\sum_{k\in\mathbb{Z}^d} p_{-k}k^\alpha = \tau^\alpha \quad \text{for all} \quad |\alpha| < N\ .$$

□

Hence, to construct a Neville filter of order N and shift τ one has to determine a set of points in $\mathbb{Z}^d$ that gives rise to a unique solution to the system of equations (4.1). Choosing $\dim \Pi^d_{N-1}$ distinct points results in a system of equations, where the matrix representing this system is just the matrix $M_{n,d}$ from equation (II.2.12). Hence, a set of $\dim \Pi^d_{N-1}$ distinct points has to be $(N-1,d)$-correct in order to obtain a unique solution to (4.1).

Thus, our approach to construct new Neville filters of order N and shift τ is to choose $\dim \Pi^d_{N-1}$ distinct points around the shift τ that form an $(N-1,d)$-correct set. In [KS00] a different ansatz is used. First a number of points around the shift τ is fixed and then it is checked in which polynomial space this set of points yields a unique solution to the system of equations (4.1) by using the Boor–Ron algorithm [BR90]. If the degree of the space is too low the number of points is increased until the desired degree is obtained. Therefore the approach in [KS00] usually results in more than $\dim \Pi^d_{N-1}$ filter coefficients, whereas our approach ends in at most $\dim \Pi^d_{N-1}$ filter coefficients. As we see below we even need less than $\dim \Pi^d_{N-1}$ filter coefficients since some filter coefficients turn out to be zero. For $d = 2$ we can even choose configurations of points that lead to a minimal number of non-zero filter coefficients, as we prove in Section 4.2.

Computation of the filter coefficients

Though the filter coefficients of a Neville filter can be obtained by just solving the system of equations (4.1), there is a more elegant way. Let K be an $(N-1,d)$-correct set, then for any $f : \mathbb{R}^d \to \mathbb{R}$ the unique polynomial $q \in \Pi^d_{N-1}$ that interpolates the points $(f(k) : k \in K)$ can be written as

$$q(x) = \sum_{k\in K} \ell_k(x) f(k)\ , \tag{4.2}$$

where the polynomials ℓ_k are called Lagrange fundamental polynomials and have the properties that $\ell_k(l) = \delta_{k,l}$ for $k, l \in K$ and that the ℓ_k only depend on the set K and not on f, see, e.g., [Coa66]. Now let $\boldsymbol{\delta}_\tau : f \mapsto f(\tau)$ for any $f : \mathbb{R}^d \to \mathbb{R}$ and consider the following evaluation rule

$$\boldsymbol{\delta}_\tau \approx \sum_{k \in K} \ell_k(\tau) \boldsymbol{\delta}_k \ ,$$

with ℓ_k from equation (4.2). Then this evaluation rule is evidently exact for all $f \in \Pi^d_{N-1}$ and thus also for all monomials x^α with $|\alpha| < N$, so yielding the equations (4.1). Hence the filter coefficients p_k of a Neville filter of order N and shift τ are equal to

$$p_{-k} = \ell_k(\tau) \quad \text{for} \quad k \in K \ .$$

So to obtain the filter taps p_{-k} we only have to compute the Lagrange fundamental polynomials ℓ_k which correspond to the correct set K and evaluate them at τ. The polynomials ℓ_k can efficiently be determined by [SX94, Algorithm 4.1].

Remark 4.2 *Let P be a Neville filter of order N and shift τ, then its filter coefficients sum up to one. This can be seen either by the system of equations (4.1) looking at the equation with $\alpha = 0$, or by choosing $f \in \Pi^d_0$ with $f \equiv 1$ then yielding*

$$f(\tau) = \sum_k \ell_k(\tau) f(k)$$

which equals

$$1 = \sum_k \ell_k(\tau) = \sum_k p_{-k} \ .$$

4.1 New family of Neville filters for the Quincunx case

We now introduce new Neville filters P of order $\tilde{N} \in \{2, 4, 6, 8\}$ with shift $\tau = D_1^{-1} t_1$ for $d = 2$. Recall from Section 2 that

$$D_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{and} \quad t_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \ .$$

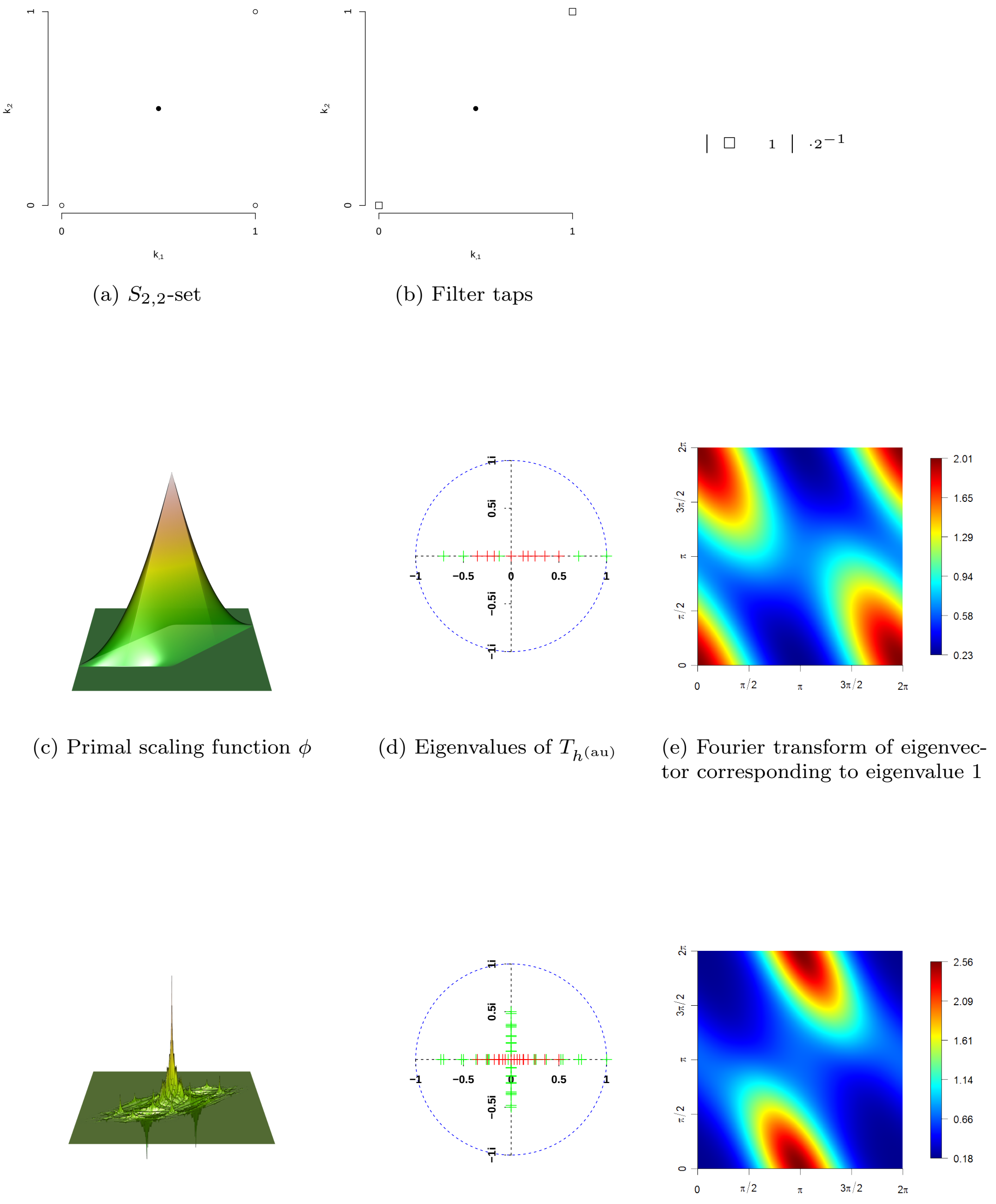
We only consider even orders $\tilde{N}$ here, since we are only interested in symmetric filters because they have zero phase, see again Section 3.1.

We use our recipe on $S_{n,d}$-sets, presented in Definition II.2.15, to choose $(\tilde{N} - 1, 2)$-correct sets, which in turn determine the filter coefficients for the Neville filters of order $\tilde{N}$ and shift $\tau = [0.5, 0.5]^T$. To be clear, it is not essential to use $S_{\tilde{N}-1,2}$-sets, we use them because the configuration of the corresponding points is quickly and easily changed.

The figures 4.1 to 4.4 on the next four pages are organized as follows: Every figure is devoted to one Neville filter of order $\tilde{N} \in \{2, 4, 6, 8\}$. On the top left of each figure the subfigure (a) displays the $S_{\tilde{N}-1,2}$-set which is used to determine the filter taps of the Neville filter P of order $\tilde{N}$. Subfigure (b) depicts the resulting non-zero filter taps, where filter taps with the same value get the same symbol. The value of each symbol is given on the right side of this plot. In both plots the shift τ is marked by a black solid dot $\bullet$.

In the second row (c)-(e) we present the primal scaling function ϕ induced by the filter P via the equations (3.15) and (2.2). We also check that this scaling function is stable by numerically verifying the assumptions of Theorem 2.4. Hence, we check if 1 is a simple eigenvalue of the matrix $T_{h^{(\mathrm{au})}}$ and that the Fourier transform of the corresponding eigenvector does not vanish. Therefore we plot the eigenvalues of $T_{h^{(\mathrm{au})}}$ where the green colored crosses depict eigenvalues with multiplicity 1 and the red ones those with multiplicity > 1. Next to this plot the mentioned Fourier transform of the eigenvector is plotted.

In the third row (f)-(h) we depict the dual scaling function which is induced by the filters P and $U = 0.5P^*$ and the equations (3.13) and (2.2). As for the primal scaling function we numerically verify the stability and plot the eigenvalues of $T_{\tilde{h}^{(\mathrm{au})}}$ and the Fourier transform of the eigenvector which corresponds to the single eigenvalue 1.



(a) $S_{2,2}$-set

(b) Filter taps

(c) Primal scaling function ϕ

(d) Eigenvalues of $T_{h^{(\mathrm{au})}}$

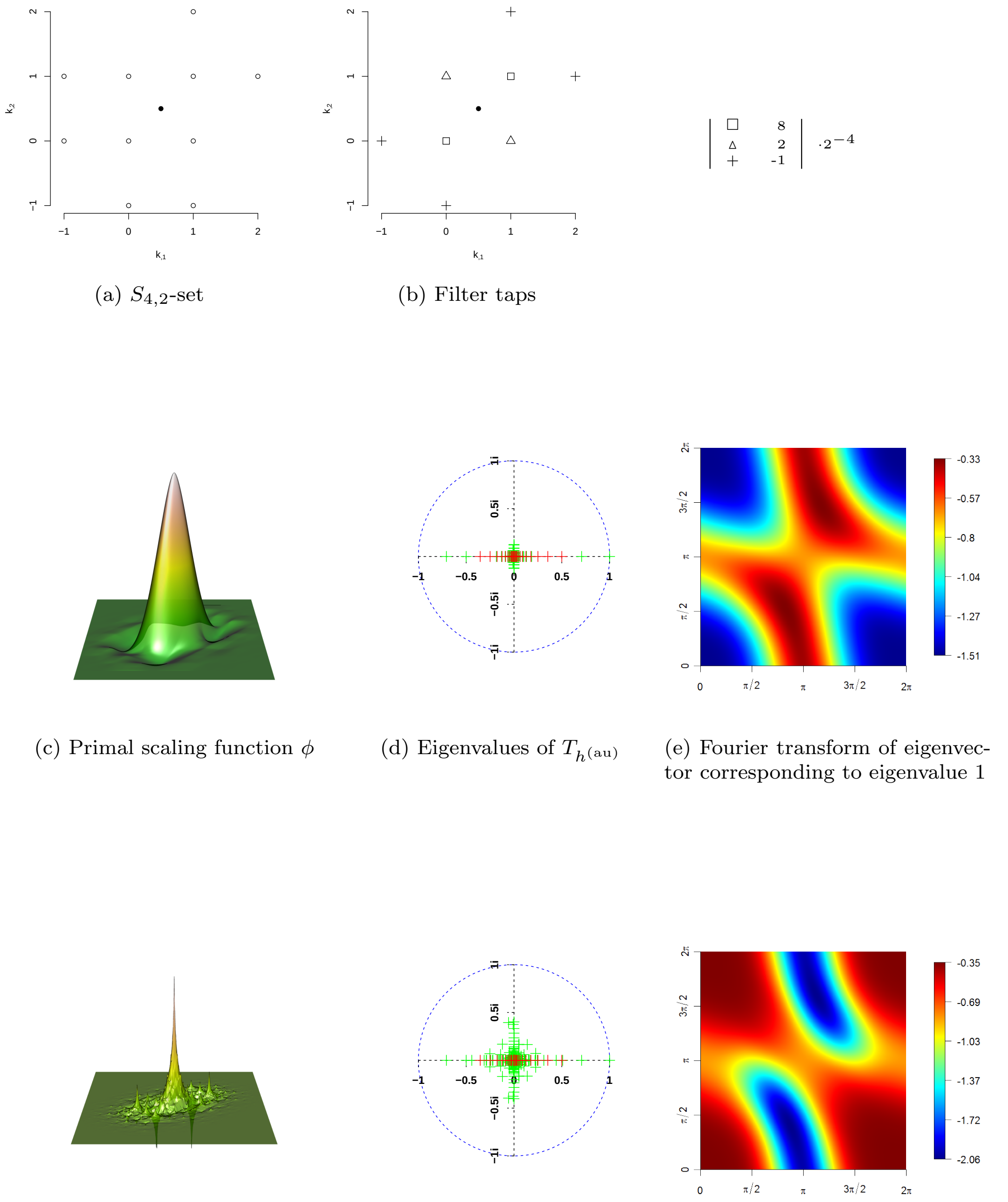
(e) Fourier transform of eigenvector corresponding to eigenvalue 1

(f) Dual scaling function $\tilde{\phi}$

(g) Eigenvalues of $T_{\tilde{h}^{(\mathrm{au})}}$

(h) Fourier transform of eigenvector corresponding to eigenvalue 1

Figure 4.1: New Neville filter of order 2



(a) $S_{4,2}$-set

(b) Filter taps

(c) Primal scaling function ϕ

(d) Eigenvalues of $T_{h^{(\mathrm{au})}}$

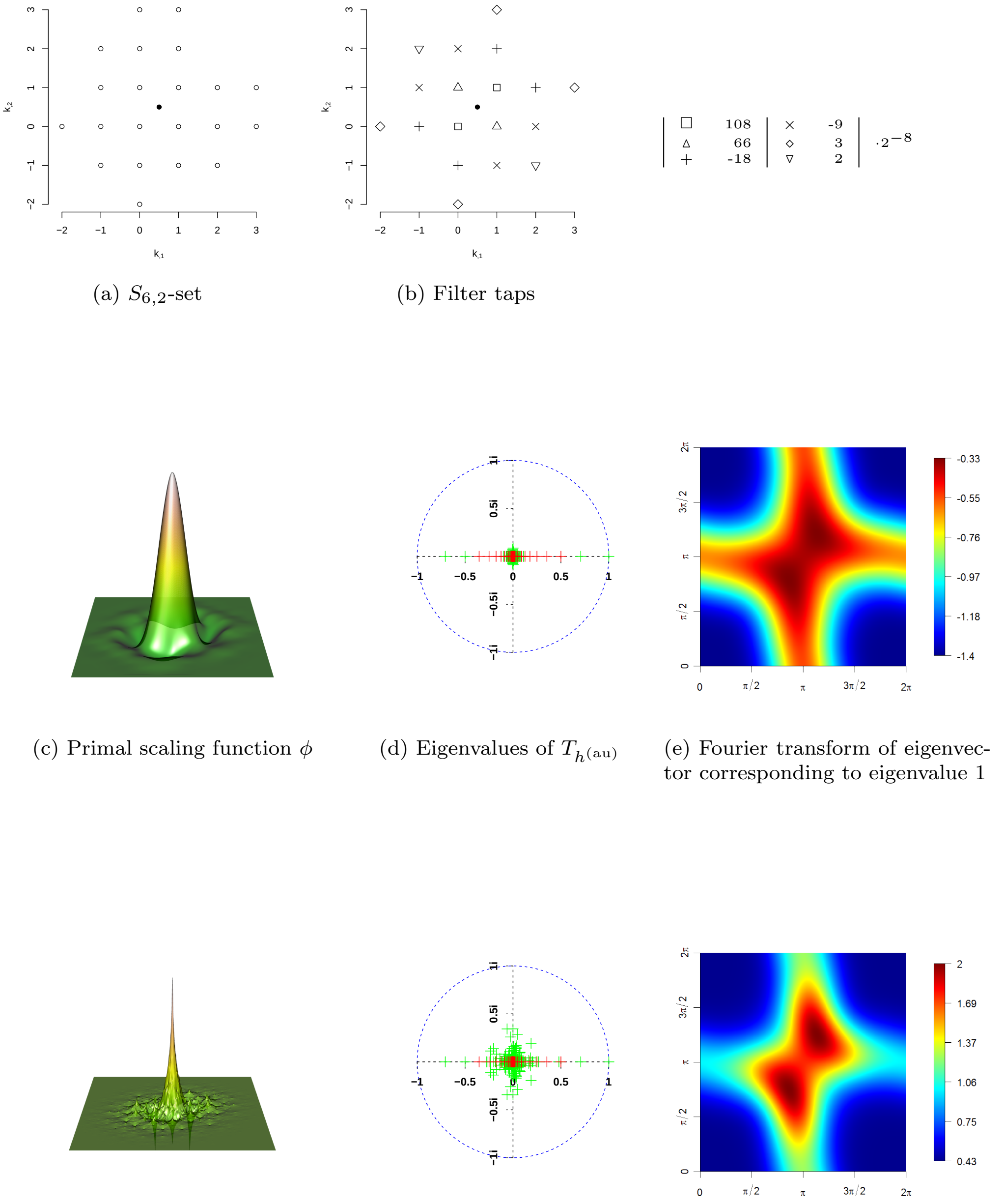
(e) Fourier transform of eigenvector corresponding to eigenvalue 1

(f) Dual scaling function $\tilde{\phi}$

(g) Eigenvalues of $T_{\tilde{h}^{(\mathrm{au})}}$

(h) Fourier transform of eigenvector corresponding to eigenvalue 1

Figure 4.2: New Neville filter of order 4



□	108	×	-9	$\cdot 2^{-8}$
△	66	◇	3	
+	-18	▽	2	

(a) $S_{6,2}$-set

(b) Filter taps

(c) Primal scaling function ϕ

(d) Eigenvalues of $T_{h^{(\mathrm{au})}}$

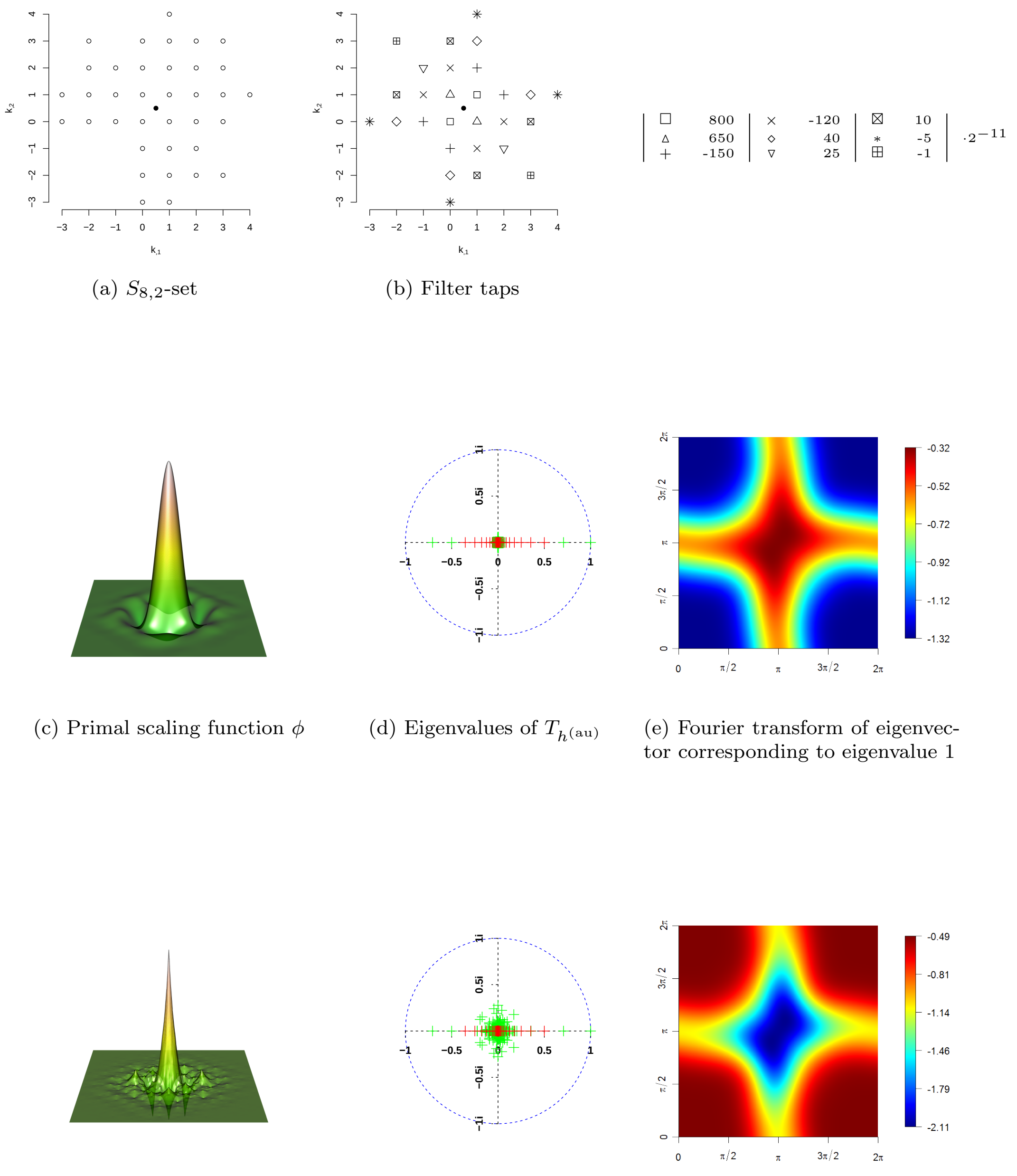
(e) Fourier transform of eigenvector corresponding to eigenvalue 1

(f) Dual scaling function $\tilde{\phi}$

(g) Eigenvalues of $T_{\tilde{h}^{(\mathrm{au})}}$

(h) Fourier transform of eigenvector corresponding to eigenvalue 1

Figure 4.3: New Neville filter of order 6



(a) $S_{8,2}$-set

(b) Filter taps

(c) Primal scaling function ϕ

(d) Eigenvalues of $T_{h^{(\mathrm{au})}}$

(e) Fourier transform of eigenvector corresponding to eigenvalue 1

(f) Dual scaling function $\tilde{\phi}$

(g) Eigenvalues of $T_{\tilde{h}^{(\mathrm{au})}}$

(h) Fourier transform of eigenvector corresponding to eigenvalue 1

Figure 4.4: New Neville filter of order 8

We summarize the new family of Neville filters of order $\tilde{N} \in \{2,4,6,8\}$ by displaying them together in one Figure 4.5. The values of the corresponding filter taps for the different Neville filters are given in the table on top of it.

$\tilde{N}$	# taps	□	△	+	×	◇	▽	⊠	∗	⊞	divided by
2	2	1									2
4	8	8	2	−1							2^4
6	18	108	66	−18	−9	3	2				2^8
8	28	800	650	−150	−120	40	25	10	−5	−1	2^{11}

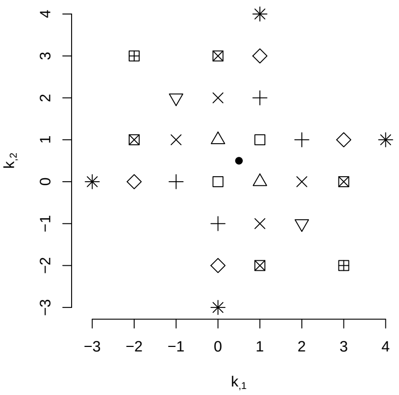


Figure 4.5: New Neville filters of order $\tilde{N}$

The Neville filters provided in [KS00] are depicted in Figure 4.6.

$\tilde{N}$	# of taps	□	△	+	×	◇	▽	⊠	divided by
2	4	1							2^2
4	12	10	−1						2^5
6	24	174	−27	2	3				2^9
8	44	23300	−4470	625	850	−75	9	−80	2^{16}

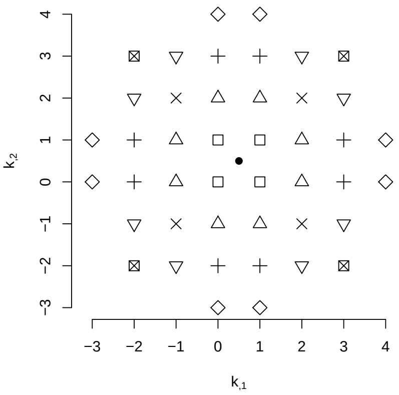


Figure 4.6: Neville filters of order $\tilde{N}$ from [KS00]

What one can conclude immediately from figures 4.5 and 4.6 is that our filters need much fewer filter taps than the filters from [KS00].

So far we just discussed filter pairs (P, U) for the lifting scheme that yield $N = \tilde{N}$ primal and dual vanishing moments. Evidently we can also provide filter pairs (P, U) that yield $\tilde{N}$ dual vanishing moments and $N \leq \tilde{N}$ primal vanishing moments by choosing P as Neville filter with order $\tilde{N}$ and $U = 0.5V^*$, where V is also one of the new constructed Neville filters with order $N \leq \tilde{N}$. The stability of all this filter pairs is numerically verified, but for a better overview we refrain from a detailed presentation as we did for the case $P = U$ on the previous pages. Instead, we present the smoothness order of the induced scaling functions of all our Neville filters in the following tables 4.1 and 4.2, where the positive smoothness orders also indicate stability. We also present the smoothness order of the induced scaling functions from [KS00] in the tables 4.3 and 4.4.

$\tilde{N}$	2	4	6	8
$-\log_2(\rho_{\tilde{N}})$	2	2.44	3.20	3.76

Table 4.1: Smoothness order of primal scaling functions corresponding to our new filters depicted in Figure 4.5

$N \setminus \tilde{N}$	2	4	6	8
2	0.44	0.59	0.39	0.32
4		0.97	0.88	0.88
6			1.06	1.17
8				1.49

Table 4.2: Smoothness order of dual scaling functions corresponding to our new filters depicted in Figure 4.5

$\tilde{N}$	2	4	6	8
$-\log_2(\rho_{\tilde{N}})$	1.58	2.45	3.15	3.78

Table 4.3: Smoothness order of primal scaling functions from [KS00], see Figure 4.6

$N \setminus \tilde{N}$	2	4	6	8
2	*n.s.*	*n.s.*	*n.s.*	*n.s.*
4		0.34	0.49	0.59
6			0.94	1.09
8				1.50

Table 4.4: Smoothness order of dual scaling functions from [KS00], see Figure 4.6

In Table 4.4 the term *n.s.* means that the dual scaling function is *not stable*. Here the major advantage of our filters compared to the filters from [KS00] can be seen, namely that all our filter pairs (P, U) with 2 primal vanishing moments yield stable dual scaling functions, which is not the case for the filters from [KS00]. Additionally the smoothness order of our scaling functions is either higher or approximately the same as the ones from [KS00]. In the next section we present filters of minimal length which are even more regular and also provide stable dual scaling functions in all cases.

4.2 Configuration of points yielding many zero filter coefficients

In Section 4.1 it could be seen that several filter taps of the Neville filters which we constructed vanished to zero. Therefore we looked for a result allowing to choose $(N-1,d)$-correct sets that yield Neville filters of order N and shift τ with many zero filter coefficients. We actually succeeded and came up with configurations that yield at most $\dim \Pi_{N-1}^{d-1}$ non-zero filter taps. For $d=2$ this means that we can construct Neville filters of order N with only N filter taps. Note that this is much less than the Neville filters presented above and in [KS00]. I first proved the result by exploiting the structure of the transpose of the multidimensional Vandermonde matrix resulting from the system of equations (4.1). But Carl de Boor gave me the hint for a much shorter proof, for which we need the following lemma.

Lemma 4.3 [Boo07, Fact 3(b)] *Let K be (n,d)-correct. Furthermore define for all $q \in \Pi^d$*

$$Z(q) := \{x \in \mathbb{R}^d : q(x) = 0\} \quad \text{and} \quad Z_K(q) := K \cap Z(q)\ .$$

Let $h \in \Pi_1^d$ with

$$\#(K \setminus Z(h)) = \dim \Pi_{n-1}^d\ ,$$

then h divides any $q \in \Pi_n^d$ for which

$$Z_K(q) \supset Z_K(h)\ .$$

Now we can prove the main result.

Theorem 4.4 *Let K be an $(N-1,d)$-correct set such that there exists a hyperplane H that contains exactly $\dim \Pi_{N-1}^{d-1}$ points from K. Moreover, let P be a Neville-filter of order N and shift τ with filter taps $p_{-k} = \ell_k(\tau)$ for all $k \in K$. Let $\tau \in H$, then*

$$p_{-k} = 0 \quad \text{for all} \quad k \in K \setminus H\ .$$

Proof. By assumption K is $(N-1,d)$-correct and there exists an $h \in \Pi_1^d$ such that the hyperplane

$$H = Z(h)$$

contains $\dim \Pi_{N-1}^{d-1}$ points from K. Hence $\#(K \setminus Z(h)) = \dim \Pi_{N-2}^d$. For $k \in K$ let ℓ_k be the corresponding Lagrange fundamental polynomials. Then by definition of the Lagrange fundamental polynomials it holds for all $k \in K \setminus Z_K(h)$ that

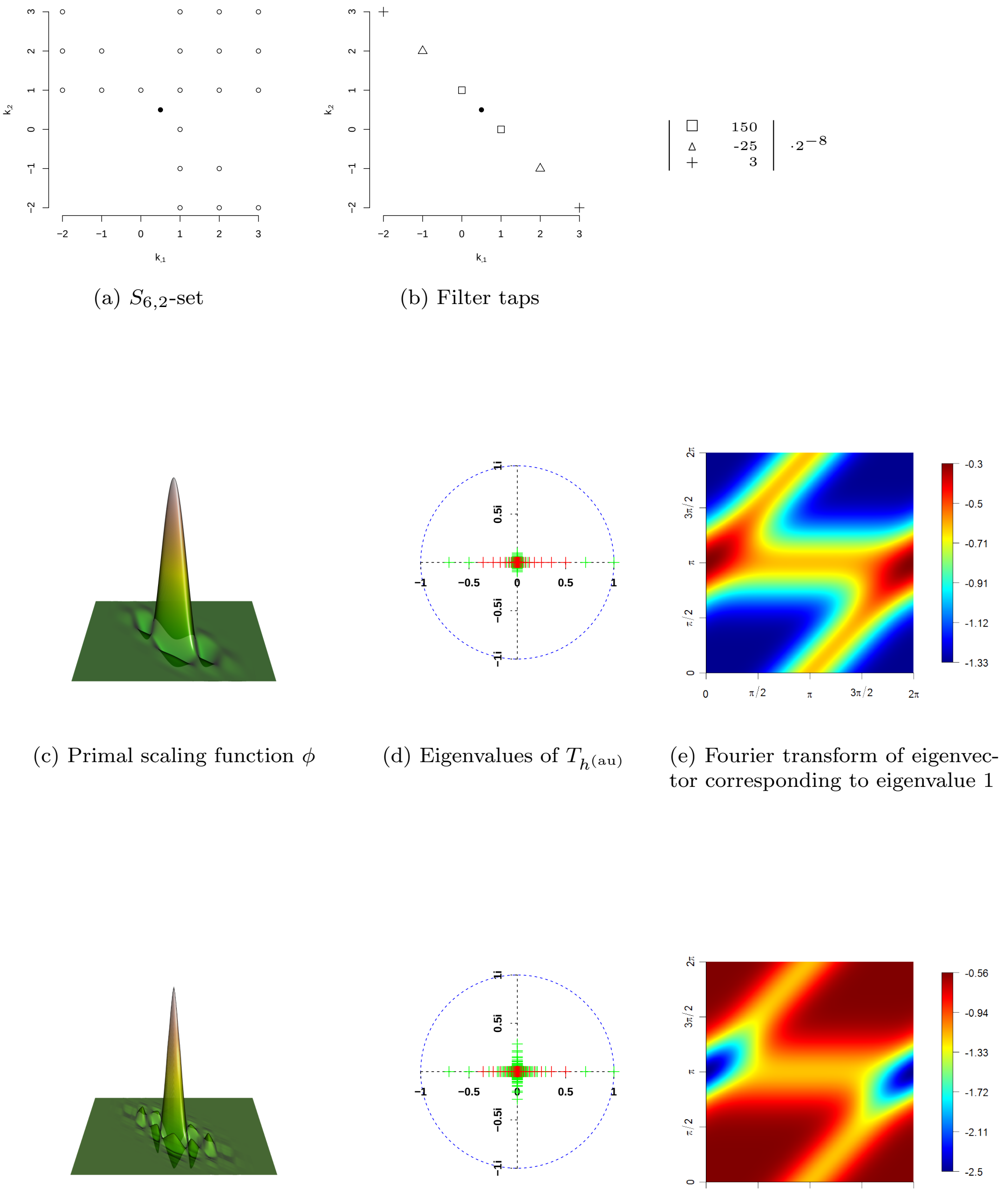
$$\ell_k(l) = 0 \quad \text{for all} \quad l \in Z_K(h)\ .$$

Hence $Z_K(\ell_k) \supset Z_K(h)$ for all $k \in K \setminus Z_K(h)$. Then we know by Lemma 4.3 that for any $k \in K \setminus Z_K(h)$ the polynomial h divides ℓ_k. Thus

$$\ell_k(\tau) = 0 \quad \text{for all} \quad k \in K \setminus Z_K(h)\ ,$$

because $\tau \in H$ implies $h(\tau) = 0$ and h is a factor of any ℓ_k with $k \in K \setminus H$. □

Hence for $d = 2$ we just need an $(N-1, 2)$-correct set K such that a line contains N points from K as well as τ. With this choice we obtain a Neville filter of order N and shift τ with at most N filter coefficients that are non-zero. In the following we present as above Neville filters of order $N \in \{2, 4, 6, 8\}$ and shift $\tau = [0.5, 0.5]^T$. In the next Figure 4.7 we present a $S_{6,2}$-set yielding only 6 non-zero filter taps and as above in Section 4.1 we numerically check the stability of the corresponding primal and dual scaling functions of order $\tilde{N} = N$. The Neville filters of order $N \in \{2, 4, 6, 8\}$ with a minimal number of filter taps are depicted in Figure 4.8. The corresponding scaling functions are also stable but are not presented in detail. Though the stability is also indicated by the smoothness order in the tables 4.5 and 4.6.



(a) $S_{6,2}$-set

(b) Filter taps

(c) Primal scaling function ϕ

(d) Eigenvalues of $T_{h^{(\mathrm{au})}}$

(e) Fourier transform of eigenvector corresponding to eigenvalue 1

(f) Dual scaling function $\tilde{\phi}$

(g) Eigenvalues of $T_{\tilde{h}^{(\mathrm{au})}}$

(h) Fourier transform of eigenvector corresponding to eigenvalue 1

Figure 4.7: Minimal Neville filter of order 6

N	# taps	□	△	+	×	divided by
2	2	1				2
4	4	9	−1			2^4
6	6	150	−25	3		2^8
8	8	1225	−245	49	−5	2^{11}

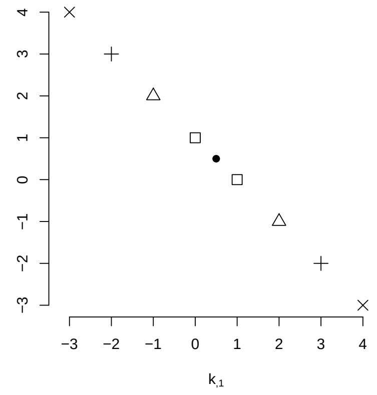


Figure 4.8: Minimal length Neville filters of order N

For the one-dimensional case it is already mentioned in [KS00] that the shortest Neville filters with shift $\tau = 0.5$ are the Deslauriers–Dubuc filters. They correspond to the Deslauriers–Dubuc subdivision, which can predict $q(\mathbb{Z}/2)$ from $q(\mathbb{Z})$ for polynomials $q \in \Pi^1_N$ of a certain degree N, see [DD87] and [DD89]. Not very surprisingly the filter taps of the Neville filters which we presented in Figure 4.8 have the same values as the Deslauriers–Dubuc filters in the one-dimensional case. So our filters can be seen as the two-dimensional extension or Quincunx extension of the one-dimensional Deslauriers–Dubuc filters.

In the two tables 4.5 and 4.6 we present the smoothness order of the scaling functions that correspond to the filters shown in Figure 4.8. From these tables it can be seen that the primal scaling functions have approximately the same smoothness order as those presented above. But the dual scaling functions are significantly smoother than the previous ones, compare Table 4.6 to tables 4.2 and 4.4.

$\tilde{N}$	2	4	6	8
$-\log_2(\rho_{\tilde{N}})$	2	2.44	3.18	3.79

Table 4.5: Smoothness order of corresponding primal scaling functions of order $\tilde{N}$

$N \setminus \tilde{N}$	2	4	6	8
2	0.44	0.59	0.65	0.68
4		1.18	1.32	1.41
6			1.77	1.91
8				2.31

Table 4.6: Smoothness order of corresponding dual scaling functions of order $N \leq \tilde{N}$

Chapter IV

Approximation of Scattered Data

This last chapter deals with approximation of scattered data, where we present a new idea using the lifting scheme. A major task in that method is to solve a least squares problem $\min_x \|Ax-b\|_2^2$, where we prove in Section 2.3 that the matrix A in our approach has a special structure, which is shown to be worth to be exploited in order to obtain better approximations. Therefore we take advantage of the results of Chapter I.

This chapter starts with a short introduction on scattered data approximation, which also contains the link to our method. Then in Section 2 our method is discussed in detail. In Section 2.4 we present some numerical experiments and compare our method to existing ones and in Section 2.5 we further discuss on the method, where we also present an idea to significantly reduce the computational effort.

1 Introduction

In many fields there is a need of reconstructing a surface out of a set of scattered data points. For instance in terrain modeling, where irregularly sampled measurements of a terrain have to be fitted by a surface to obtain a relief map. Another important field where scattered data reconstruction is needed is image processing, for example superresolution or inpainting, where we present an according example in Section 2.4.

We denote the set of the scattered data sites by $\Xi := \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$. Furthermore let $s : \mathbb{R}^d \to \mathbb{R}$, then we define the values at the scattered data sites to be $s(x)$ for all $x \in \Xi$. The task of scattered data reconstruction is now to find a function $f : \mathbb{R}^d \to \mathbb{R}$ that fits the given data $s|_\Xi := \{s(x) : x \in \Xi\}$. Clearly, one choice is interpolation, i.e., determine an f such that $f(x) = s(x)$ for all $x \in \Xi$. One popular ansatz to interpolate scattered data is to use radial basis functions, where a function $\bar{\phi} : \mathbb{R}^d \to \mathbb{R}$ is called radial if there exists a function $\phi : [0, \infty) \to \mathbb{R}$ with the property $\bar{\phi}(x) = \phi(\|x\|_2)$ for all $x \in \mathbb{R}^d$, cf. [Wen05, Definition 6.15]. Thus, the values of $\bar{\phi}$ depend only on the Euclidean distance to its origin, which explains the term radial. Standard choices for a radial basis function are for instance the Gaussian $\phi(r) = e^{-cr^2}$ for some positive parameter c, the multiquadratic radial basis function $\phi(r) = \sqrt{r^2 + c^2}$ with c also being some positive parameter, or the so-called thin-plate spline radial basis function which is defined by $\bar{\phi}(x) = \|x\|_2^2 \log(\|x\|_2)$. The radial property with respect to the Euclidean norm is very useful in theoretical considerations, since in most cases it reduces the problem to a one-dimensional one. For more details on radial basis functions we refer to [Buh03]. Now, to every scattered data site $x \in \Xi$ one translate of the radial basis function $\bar{\phi}(\cdot - x)$ is assigned and the interpolant is determined by

$$f = \sum_{x \in \Xi} w_x \bar{\phi}(\cdot - x) \ , \tag{1.1}$$

and by the condition $f(x) = s(x)$ for all $x \in \Xi$. The uniqueness of the coefficients w_x is dependent on the radial basis function itself, where a radial basis function which ensures uniqueness is called *positive definite*. For instance the Gaussian radial basis function is positive definite. However, the multiquadratic and the thin-plate spline basis function are just conditionally positive definite, which means that an extra polynomial term has to be added to the right hand side of (1.1) to retain uniqueness, see, e.g., [Buh00], [Buh03] or [Wen05]. Moreover, in the 2-dimensional case the unique interpolant f, which is obtained by thin-plate spline radial basis functions additionally minimizes the so-called *bending energy*

$$E(f) := \iint \left(\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right) \mathrm{d}x \mathrm{d}y \ ,$$

see [Duc76] or [Duc77].

In case that the given data $s|_\Xi$ is for instance corrupted by noise or just a coarse approximation to the data is needed, approximation in contrast to interpolation of the data is more appropriate, i.e., determine a function f such that $f(x) \approx s(x)$ in some sense for all $x \in \Xi$. As in the interpolation case also approximation by radial basis functions is usually considered, where for $d = 2$ a popular choice is the so-called *thin-plate smoothing spline* (see, e.g., [Wah90, Section 2.4]) which is the solution to

$$\min_{f \in H^2} \sum_{x \in \Xi} (f(x) - s(x))^2 + \tau E(f) \ , \tag{1.2}$$

with H^2 being the Beppo–Levi space of functions whose partial derivatives of total order 2 are in $L^2(\mathbb{R}^2)$; for more details on H^2 see [Mei79]. The regularization parameter τ in (1.2) balances between the exact fitting and the roughness of the approximant, i.e., small τ leads to an approximant that fits the data $s|_\Xi$ accurately but is probably not very smooth and vice versa.

The major drawback of approximating or interpolating scattered data by radial basis functions is that the square matrix to be operated with is, due to the globally supported basis functions, fully occupied and additionally badly conditioned, cf. [JSX09] and the references therein. Additionally, the size of this square matrix is also directly connected to the number of scattered data sites $\#\Xi$. Thus, the approach gets computationally expensive for large sets of scattered data sites, especially for $d \geq 2$. Hence we also follow the argumentation in [JSX09], which states that it is worth to pursue alternative approaches. As the approach in [JSX09], which we are going to present below in Section 2.4, also our approach on scattered data approximation, which we introduce in Section 2.2, is based on shift invariant subspaces.

Approximation from shift invariant subspaces of $L^2(\mathbb{R}^d)$

Another ansatz to scattered data approximation is to consider shift invariant subspaces of $L^2(\mathbb{R}^d)$. These subspaces are determined by the closure of the integer translates of one continuous compactly supported function $\phi : \mathbb{R}^d \to \mathbb{R}$

$$S(\phi) := \overline{\{ \sum_{k \in \mathbb{Z}^d} c_k \phi(\cdot - k) : c_k \in \mathbb{R} \ \text{ with almost all } \ c_k = 0 \}} \ ,$$

see, e.g., [BDR94]. Approximation from shift invariant subspaces $S(\phi)$ of $L^2(\mathbb{R}^d)$ has two beneficial properties. Firstly all is set up by just one function ϕ and secondly if ϕ is capable to reproduce polynomials up to degree $N-1$ it provides approximation order N to sufficiently smooth functions $g \in L^2(\mathbb{R}^d)$, i.e., there exists a constant $C > 0$ such that

$$\inf_{f \in S^h(\phi)} \|f - g\|_{L^2(\mathbb{R}^d)} \leq Ch^N$$

for all $h > 0$ and $S^h(\phi) := \{f(\cdot/h) : f \in S(\phi)\}$, see [Jia98]. Special choices of h lead to spaces $S^h(\phi)$ which fit into the scheme of multiresolution analysis, see Definition III.2.2. Thus, a natural approach is to find an $f \in V_J$ that fits the scattered data in the least square sense, i.e.,

$$\min_{f \in V_J} \sum_{x \in \Xi} (f(x) - s(x))^2 = \min_{(c_{J,k})_{k \in \mathbb{Z}^d}} \sum_{x \in \Xi} \Big(\sum_{k \in \mathbb{Z}^d} c_{J,k}\phi_{J,k}(x) - s(x) \Big)^2 . \tag{1.3}$$

Usually the domain $\Omega \subset \mathbb{R}^d$ on which the scattered data is located and on which the approximation is determined is bounded. Therefore we denote by

$$\Omega_J := \{k \in \mathbb{Z}^d : \text{supp } \phi_{J,k} \cap \Omega \neq \emptyset\}$$

the set of indices k for which $\phi_{J,k}$ has influence on Ω. Furthermore, we denote the space which is spanned by these functions by

$$V_J(\Omega) := \{ \sum_{k \in \Omega_J} c_{J,k}\phi_{J,k}(\cdot - k) : c_{J,k} \in \mathbb{R}\} .$$

Hence, for bounded domains Ω equation (1.3) then becomes

$$\min_{f \in V_J(\Omega)} \sum_{x \in \Xi} (f(x) - s(x))^2 = \min_{(c_{J,k})_{k \in \Omega_J}} \sum_{x \in \Xi} \Big(\sum_{k \in \Omega_J} c_{J,k}\phi_{J,k}(x) - s(x) \Big)^2 , \tag{1.4}$$

which can be written in matrix vector form, if we define $\Omega_J =: \{k_1, \dots, k_{\#\Omega_J}\}$, as

$$\min_{\mathbf{c}_J} \|A_J \mathbf{c}_J - \mathbf{s}\|_2^2 :=$$

$$\min_{(c_{J,k})_{k \in \Omega_J}} \left\| \begin{bmatrix} \phi_{J,k_1}(x_1) & \cdots & \phi_{J,k_{\#\Omega_J}}(x_1) \\ \vdots & & \vdots \\ \phi_{J,k_1}(x_n) & \cdots & \phi_{J,k_{\#\Omega_J}}(x_n) \end{bmatrix} \begin{bmatrix} c_{J,k_1} \\ \vdots \\ c_{J,k_{\#\Omega_J}} \end{bmatrix} - \begin{bmatrix} s(x_1) \\ \vdots \\ s(x_n) \end{bmatrix} \right\|_2^2 . \tag{1.5}$$

Thus an approximant f to the scattered data $s|_\Xi$ on Ω can be obtained by

$$f(x) = \sum_{k \in \Omega_J} c^\star_{J,k}\phi_{J,k}(x) \quad \text{for all} \quad x \in \Omega , \tag{1.6}$$

with $\mathbf{c}^\star_J := \arg\min_{\mathbf{c}_J} \|A_J\mathbf{c}_J - \mathbf{s}\|_2$. In the following, bold symbols can denote sequences or vectors, the case gets clear from the context.

For the one-dimensional case this is basically done in [FE98], where J is chosen to be the smallest J for which the matrix A_J is overdetermined and has a good condition number. The approach from [FE98] is extended to the two-dimensional case in [NM99] by using tensor wavelets. However, the approach in [NM99] is limited to scattered data that is located on sublattices, i.e., the data cannot be scattered arbitrarily on $\Omega \subset \mathbb{Z}^2$. In [BLC04] the approach of [NM99] is explained in terms of the lifting scheme but suffers like the approach [NM99] from the fact that the data cannot be scattered arbitrarily.

Remark 1.1 *Note that all approaches which are based on generators ϕ that are implicitly defined by the refinement equation (III.2.2) are limited to scaled subsets of $\mathbb{Z}^d$, i.e., on $\Omega \subset \tau\mathbb{Z}^d$ for some $\tau > 0$. This is because the evaluation of ϕ is just possible at dyadic points. However fast evaluation is possible by applying the cascade algorithm.*

A method on scattered data approximation which is also based on shift invariant spaces can be found in [JSX09]. This approach also minimizes the regularized least squares problem (1.2), but with a shift invariant subspace $S^h(\phi)$ instead of the Beppo–Levi space H^2. We discuss the method from [JSX09] in more detail in Section 2.4, where we also sketch the ansatz from [CK05] which is also basically based on shift invariant spaces.

In the next section we introduce a method to approximate scattered data which is based on the lifting scheme and therefore also implicitly on shift invariant subspaces. Moreover, we reveal that we can exploit properties of the matrix A_J from equation (1.5) to obtain better approximants to the data $s|_\Xi$. For this we take advantage of the results from Chapter I.

2 Approximation of scattered data using the lifting scheme

In this section we introduce a method to approximate scattered data points by the lifting scheme. Using the lifting scheme makes it necessary to consider Ω as a bounded subset of $\mathbb{Z}^d$ (cf. Remark 1.1) in which case it holds that $\Omega_0 = \Omega$. Clearly in later applications Ω_0 can be treated as a scaled subset of $\mathbb{Z}^d$, i.e., $\Omega \subset \tau\mathbb{Z}^d$ for some $\tau > 0$, which implies $\Omega_0 = \tau^{-1}\Omega$.

So from now on let $\Omega_0 \subset \mathbb{Z}^d$ be bounded. Then equation (1.6) can be expressed in terms of the lifting scheme as it is depicted in Figure 2.1. Note that the filter U is not needed here, this also gets clear when considering again equation (III.3.15).

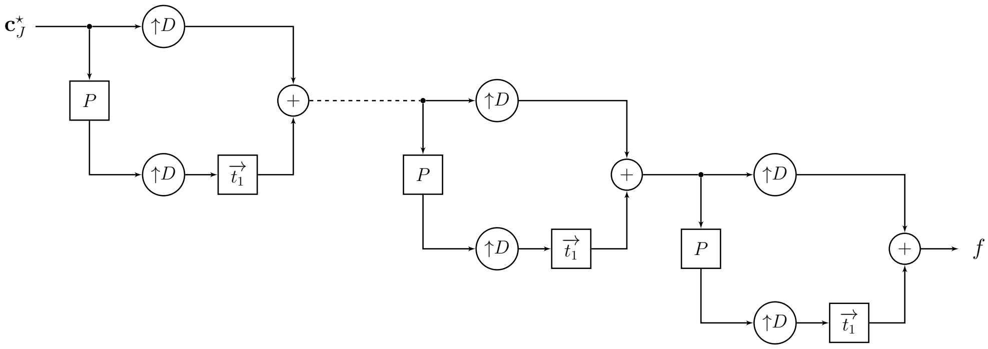


Figure 2.1: J connected synthesis parts of the lifting scheme without filter U

With Figure 2.1 in mind, it can quite intuitively be seen that approximating scattered data using the lifting scheme basically consists of four steps, namely:

1. Determine Ω_J from Ω_0 and the filter P.
2. Determine the relationship between $(c_{J,k})_{k\in\Omega_J}$ and the scattered data $s|_\Xi$, i.e., set up the matrix A_J.
3. Solve the least squares problem $\mathbf{c}_J^\star = \arg\min_{\mathbf{c}_J} \|A_J\mathbf{c}_J - \mathbf{s}\|_2$.
4. Apply $\mathbf{c}_J^\star$ to J connected synthesis parts of the lifting scheme and obtain the approximation f, see Figure 2.1.

In Section 2.1 we present a didactive and concrete example, explaining what the four steps look like in the 1-dimensional case. In Section 2.2 we explicitly explain the steps 1, 2 and 4 for arbitrary dimension d, whereas in Section 2.3 we discuss in detail step 3, i.e., how to solve the least squares problem $\min_{\mathbf{c}_J} \|A_J\mathbf{c}_J - \mathbf{s}\|_2^2$ properly, by exploiting properties of the matrix A_J.

2.1 Example of the approach for $d = 1$

This section is thought as a didactic, less mathematical, presentation of the approach to approximate scattered data by using the lifting scheme in the case $d = 1$. It is not necessarily required to read this section, but it might serve for a better understanding.

The filter P we use in this example is of the form

$$P(z) = p_{-1}z^1 + p_0z^0 + p_1z^{-1} \ ,$$

with $p_{-1}, p_0, p_1 \in \mathbb{R}$. Recall from Section III.3.1 that upsampling of a sequence $\mathbf{x} = (x_k)_{k\in\mathbb{Z}}$ in $1d$ is defined as

$$((\uparrow 2)\mathbf{x})_k = \begin{cases} x_{k/2} & \text{if } k/2 \in \mathbb{Z} \\ 0 & \text{else} \end{cases} .$$

In this example we assume $J = 1$ and $\Omega_0 := \{1, 2, 3, 4, 5, 6\}$.

Step 1 – Determine Ω_1 from Ω_0 and the filter P

We use Figure 2.2 to explain how to determine Ω_1 from Ω_0 and the filter P. We start at the right side ⓐ, with the grid $\Omega_0 = \{1, 2, 3, 4, 5, 6\}$ of an arbitrary signal $(x_k)_{k\in\Omega_0}$. Then, we investigate which elements on the left side ⓓ have an influence on elements located at Ω_0 ⓐ, i.e., on $(x_k)_{k\in\Omega_0}$. Therefore we first consider the upper path of the synthesis part, where we can say that due to upsampling only elements on the positions 1, 2 and 3 have an influence ⓑ. Secondly, on the lower path, due to shifting and downsampling, elements on 0, 1 and 2 ⓒ are influencing elements located at Ω_0. Since we know the structure of the filter P we see that the elements positioned at $-1, 0, 1, 2, 3$ have an influence on the elements located at $0, 1, 2$ ⓒ. Summing up, signals $(c_k)_{k\in\Omega_1}$ at ⓓ with $\Omega_1 := \{-1, 0, 1, 2, 3\}$ have an influence on $(x_k)_{k\in\Omega_0}$ ⓐ.

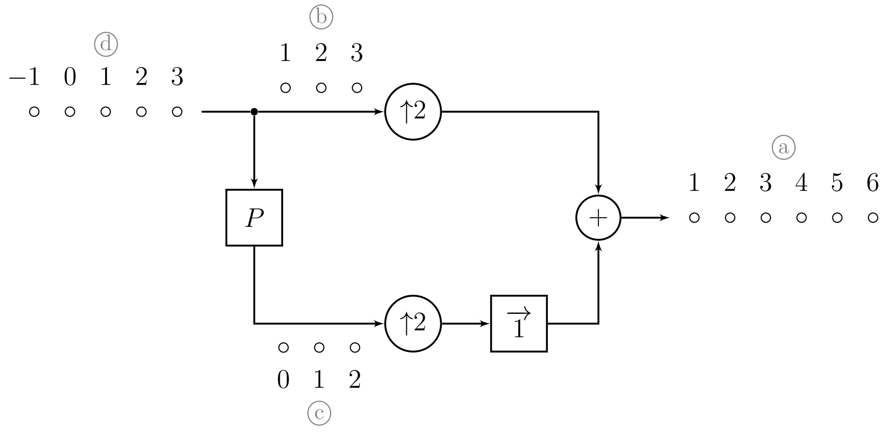
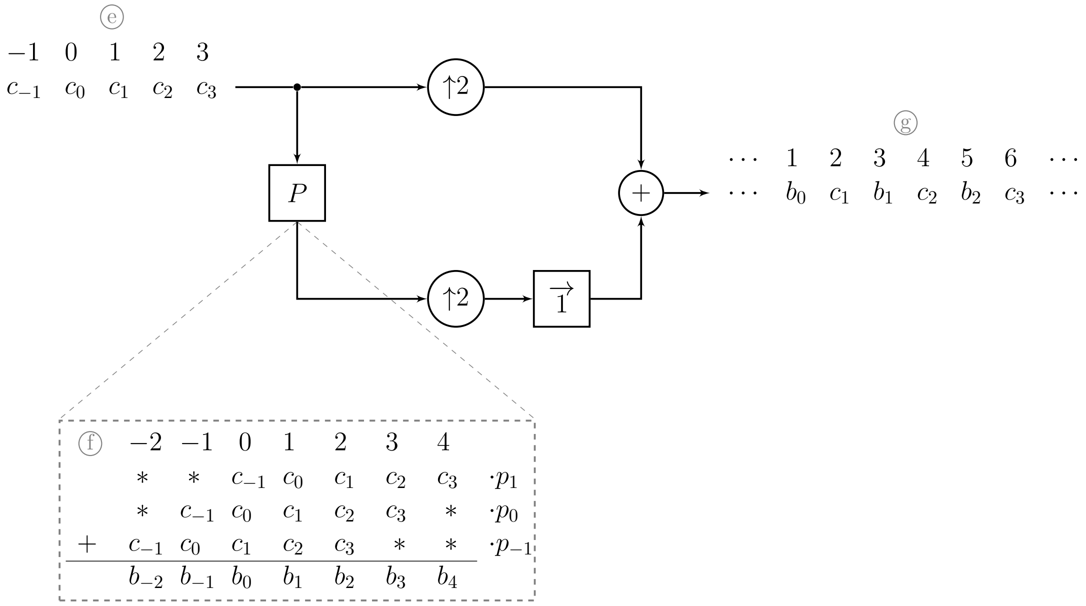


Figure 2.2: Sketch of how to obtain Ω_1 from Ω_0

Step 2 – Determine the matrix A_J

Let $\mathbf{c}_1 := (c_k)_{k\in\Omega_1}$ be an arbitrary signal. In this paragraph we determine the relationship between $\mathbf{c}_1$ and the result after applying it to the synthesis part of the lifting scheme. To explain this in an easy manner we use Figure 2.3.


ⓔ
-1 0 1 2 3
c_{-1} c_0 c_1 c_2 c_3
$\uparrow 2$
P
$\uparrow 2$
$\overrightarrow{1}$
$+$
ⓖ
$\cdots$ 1 2 3 4 5 6 $\cdots$
$\cdots$ b_0 c_1 b_1 c_2 b_2 c_3 $\cdots$

ⓕ	-2	-1	0	1	2	3	4	
	$*$	$*$	c_{-1}	c_0	c_1	c_2	c_3	$\cdot p_1$
	$*$	c_{-1}	c_0	c_1	c_2	c_3	$*$	$\cdot p_0$
$+$	c_{-1}	c_0	c_1	c_2	c_3	$*$	$*$	$\cdot p_{-1}$
	b_{-2}	b_{-1}	b_0	b_1	b_2	b_3	b_4	

Figure 2.3: Influence of $(c_k)_{k\in\Omega_1}$ on $(f_k)_{k\in\Omega_0}$

At ⓔ the signal $\mathbf{c}_1$ is depicted and at ⓕ the action of the filter P to $\mathbf{c}_1$ is presented. The stars $*$ there have to be defined by boundary conditions, for instance zero-padding. But we do not need them at all. This is because on ⓒ we saw that only elements positioned at $0, 1, 2$ are influencing the elements on Ω_0, and as one can see in ⓕ at $0, 1, 2$ no boundary conditions are needed. Hence, the result restricted to Ω_0 obtained by applying $(c_k)_{k\in\Omega_1}$ to the synthesis part of the lifting scheme is ⓖ

$$(f_k)_{k\in\Omega_0} := (b_0, c_1, b_1, c_2, b_2, c_3) \; .$$

So, we can completely describe $(f_k)_{k\in\Omega_0}$ by the filter coefficients (p_{-1}, p_0, p_1) and the signal $(c_k)_{k\in\Omega_1}$

$$\begin{bmatrix} p_1 & p_0 & p_{-1} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & p_1 & p_0 & p_{-1} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & p_1 & p_0 & p_{-1} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} b_0 \\ c_1 \\ b_1 \\ c_2 \\ b_2 \\ c_3 \end{bmatrix} \; . \tag{2.1}$$

Let $s(x)$ with $x \in \Xi := \{1, 4, 5\}$ be the scattered data located at Ω_0, then the idea is to choose $\mathbf{c}_1$ such that the scattered data is approximated in a least square sense by $(f_k)_{k\in\Xi}$, i.e., we need to solve the least squares problem

$$\mathbf{c}_1^\star := \arg\min_{\mathbf{c}_1} \left\| \begin{bmatrix} p_1 & p_0 & p_{-1} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & p_1 & p_0 & p_{-1} \end{bmatrix} \begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} - \begin{bmatrix} s(1) \\ s(4) \\ s(5) \end{bmatrix} \right\|_2 , \tag{2.2}$$

where the matrix just consists of the rows $1, 4, 5$ of the matrix from equation (2.1).

Step 3 – Solve the least squares problem (2.2), i.e., determine $\mathbf{c}_1^\star$

Step 4 – Compute the approximant to the scattered data

The approximation on whole Ω_0 is then obtained either by multiplying $\mathbf{c}_1^\star$ to the matrix from equation (2.1), or by just applying $\mathbf{c}_1^\star$ to the synthesis part of the lifting scheme and taking into account only the signal coefficients that are located at Ω_0. How to solve the least squares problem properly is discussed below in Section 2.3.

Remark 2.1 *In this particular example the signal at ⓓ is only* 1 *element shorter than the desired signal. This is due to the short length of* Ω_0*. If one takes a larger grid this changes and the signal at ⓓ is about half the size of* Ω_0*. Furthermore, we considered only one synthesis part, later we use* J *connected synthesis parts, as it is indicated in Figure 2.1.*

2.2 The approach for arbitrary d

As stated above, the approach of approximating scattered data consists of four steps. In this section we explicitly explain Steps 1, 2 and 4, whereas the discussion of Step 3 is postponed for a better clarity to Section 2.3.

Step 1 – Determine Ω_J from Ω_0 and filter P

The first step in our approach is to determine Ω_J from Ω_0 and the filter P. Therefore we first explain how to obtain Ω_j from Ω_{j-1} and P for an arbitrary $j \in \mathbb{Z}_+$. If this is clear one just iterates through $j \in 1{:}J$ and so obtains subsequently Ω_J starting from Ω_0. We use Figure 2.4 to sketch how Ω_j is obtained from Ω_{j-1} and P. Similarly to Section III.3.2 we denote elements using the upper path *even* and the ones using the lower path *odd*. Hence, the sets $\Omega_{j-1}^{\text{even}}$ and $\Omega_{j-1}^{\text{odd}}$ depicted in Figure 2.4 are equal to

$$\Omega_{j-1}^{\text{even}} = \{D^{-1}k : k \in \Omega_{j-1}, D^{-1}k \in \mathbb{Z}^d\}$$

and

$$\Omega_{j-1}^{\text{odd}} = \{D^{-1}(k - t_1) : k \in \Omega_{j-1}, D^{-1}k \notin \mathbb{Z}^d\} .$$

Let P be an arbitrary Neville filter of a certain order N and shift $D^{-1}t_1$:

$$P(z) = \sum_{i \in I} p_i z^{-i}$$

with $I \subset \mathbb{Z}^d$ finite. Then, we can describe the set Ω_j by

$$\Omega_j = \{k - i : k \in \Omega_{j-1}^{\text{odd}}, i \in I\} \cup \Omega_{j-1}^{\text{even}}$$

and thus starting from Ω_0 we can determine Ω_J iteratively.

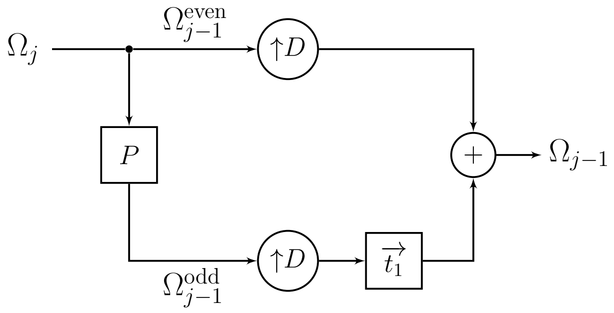


Figure 2.4: Sketch of how to obtain Ω_j from Ω_{j-1}

Step 2 – Setting up the least squares problem

Let $\mathbf{c}_J := (c_{J,k})_{k\in\Omega_J}$ be an arbitrary signal and let $(f_k)_{k\in\mathbb{Z}^d}$ denote the result of $\mathbf{c}_J$ applied to J connected synthesis parts of the lifting scheme, where we are just interested in $\mathbf{f} := (f_k)_{k\in\Omega_0}$ – the restriction of the result to Ω_0. There are different ways to determine the relationship between $\mathbf{c}_J$ and $\mathbf{f}$. One is to consider the signal $(\delta_k)_{k\in\mathbb{Z}^d}$, defined as

$$\delta_k := \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{else} \end{cases} ,$$

and to apply its translates $(\delta_{l-k})_{l\in\mathbb{Z}^d}$ for all $k \in \Omega_J$ to J connected synthesis parts of the lifting scheme. Let $(\Phi_{J,k}(l))_{l\in\mathbb{Z}^d}$ denote the result for each $k \in \Omega_J$. Then for each $l \in \Omega_0$

$$\Phi_{J,k}(l)$$

describes the relationship of $c_{J,k}$ to a single element f_l. Thus it holds that

$$f_l = \sum_{k\in\Omega_J} c_{J,k}\Phi_{J,k}(l) \quad \text{for all} \quad l \in \Omega_0 \ .$$

As above let $\Xi =: \{x_1, \ldots, x_n\}$ denote the location of the scattered data sites, with $\Xi \subset \Omega_0$. The values of the scattered data are given by $s|_\Xi$. To determine an approximant to the scattered data we solve a least squares problem, which, if we define $\Omega_J =: \{k_1, \ldots, k_{\#\Omega_J}\}$, can be written as

$$\min_{\mathbf{c}_J} \|A_J\mathbf{c}_J - \mathbf{s}\|_2^2 \ ,$$

with $A_J(l,m) := \Phi_{J,k_l}(x_m)$, $\mathbf{c}_J(l) := c_{J,k_l}$ and $\mathbf{s}(m) := s(x_m)$ for $l \in 1{:}\#\Omega_J$ and $m \in 1{:}n$.

Remark 2.2 *Note that it is sufficient to apply just the sequence $(\delta_k)_{k\in\mathbb{Z}^d}$ to J connected synthesis parts of the lifting scheme to determine the matrix A_J. This is because*

$$\Phi_{J,k}(l) = \Phi_{J,0}(l - D^J k) \quad \textit{for all} \quad k \in \Omega_J \ \textit{ and } \ l \in \Omega_0 \ ,$$

which holds because of the following lemma.

Lemma 2.3 *Let $\mathbf{x} = (x_k)_{k\in\mathbb{Z}^d}$ be an arbitrary sequence. Denote by $(y_k)_{k\in\mathbb{Z}^d}$ the result of applying $\mathbf{x}$ to the upper input of a synthesis part of the lifting scheme. Then it holds that applying the shifted sequence $(x_{k+t})_{k\in\mathbb{Z}^d}$ with $t \in \mathbb{Z}^d$ yields the same but shifted result $(y_{k+Dt})_{k\in\mathbb{Z}^d}$.*

Proof. Applying $\mathbf{x}$ to the upper input of the synthesis part of the lifting scheme reads

$$\begin{aligned}(y_k)_{k\in\mathbb{Z}^d} &= \Big((\uparrow D) + (\overrightarrow{t_1})(\uparrow D)P\Big)\,(x_k)_{k\in\mathbb{Z}^d}\\ &= (\uparrow D)(x_k)_{k\in\mathbb{Z}^d} + (\overrightarrow{t_1})(\uparrow D)P(x_k)_{k\in\mathbb{Z}^d}\\ &= (\uparrow D)(x_k)_{k\in\mathbb{Z}^d} + (\overrightarrow{t_1})(\uparrow D)\Big(\sum_{l\in\mathbb{Z}^d} p_{k-l}x_l\Big)_{k\in\mathbb{Z}^d}\ .\end{aligned}$$

Hence

$$y_k = \begin{cases} x_{D^{-1}k} & \text{if}\quad D^{-1}k\in\mathbb{Z}^d\\ \sum_{l\in\mathbb{Z}^d} p_{D^{-1}(k-t_1)-l}x_l & \text{else}\end{cases}\ . \tag{2.3}$$

Applying $(x_{k+t})_{k\in\mathbb{Z}^d}$ to the upper input of the synthesis part of the lifting scheme yields

$$(z_k)_{k\in\mathbb{Z}^d} = (\uparrow D)(x_{k+t})_{k\in\mathbb{Z}^d} + (\overrightarrow{t_1})(\uparrow D)P(x_{k+t})_{k\in\mathbb{Z}^d}\ ,$$

where $(P(x_{l+t})_{l\in\mathbb{Z}^d})_k$ is equal to

$$\sum_{l\in\mathbb{Z}^d} p_{k-l}x_{l+t} = \sum_{l\in\mathbb{Z}^d} p_{k+t-l}x_l\ ,$$

by index shift. Thus

$$z_k = \begin{cases} x_{D^{-1}k+t} & \text{if}\quad D^{-1}k\in\mathbb{Z}^d\\ \sum_{l\in\mathbb{Z}^d} p_{D^{-1}(k-t_1)+t-l}x_l & \text{else}\end{cases}\ .$$

Hence $z_k = y_{k+Dt}$. □

Remark 2.4 *Remember that applying $(\delta_k)_{k\in\mathbb{Z}^d}$ to the upper input of the synthesis part of the lifting scheme is the same as applying $(\delta_k)_{k\in\mathbb{Z}^d}$ to the upper input of the standard two-channel filter bank when we exploit equation (III.3.15). Hence our procedure of applying $(\delta_k)_{k\in\mathbb{Z}^d}$ to J connected synthesis parts of the lifting scheme is the same as applying the cascade algorithm (see again equation (III.2.12)) J times for some initial ϕ_0 satisfying*

$$\sum_{k\in\mathbb{Z}^d}\phi_0(x-k) = 1 \quad \textit{with} \quad x\in\mathbb{R}^d$$

and $\phi_0(k) = \delta_k$ for all $k\in\mathbb{Z}^d$. If the filter coefficients of the filter P induce over (III.3.15) and (III.2.11) a Riesz basis, then the cascade algorithm converges, see again Section III.2.2. Denote the limit by ϕ, then due to the interpolating property of the lifting scheme (2.3) $\Phi_{J,0}$ is the exact evaluation of $\phi(D^{-J}\cdot)$ on $\mathbb{Z}^d$, i.e.,

$$\Phi_{J,0}(k) = \phi(D^{-J}k) \quad \forall k\in\mathbb{Z}^d\ .$$

Step 3 – Solving the corresponding least squares problem

How to obtain an appropriate solution

$$\mathbf{c}_J^\star := \arg\min_{\mathbf{c}_J}\|A_J\mathbf{c}_J - \mathbf{s}\|_2$$

to the least squares problem is subject of Section 2.3 below.

Step 4 – Compute the approximant to the scattered data

With the solution $\mathbf{c}_J^\star$ to the least squares problem the approximant $\mathbf{f} = (f_k)_{k\in\Omega_0}$ to the scattered data can be obtained in two different ways. Either by applying $\mathbf{c}_J^\star$ to the J connected synthesis parts, where after each synthesis part $j = J-1, J-2, \dots, 0$ we restrict the result to the set Ω_j, or by

$$f_l = \sum_{k\in\Omega_J} c_{J,k}\Phi_{J,k}(l) \quad \text{for all} \quad l \in \Omega_0 \; .$$

2.3 Solving the corresponding least squares problem

In this section we discuss the solution of the least squares problem

$$\min_{\mathbf{c}_J} \|A_J\mathbf{c}_J - \mathbf{s}\|_2^2 \; . \tag{2.4}$$

Before we continue we want to highlight two important things. Firstly that the matrix A_J has a special property, namely

$$A_J E_n = E_m \quad \text{with} \quad E_n = [1, \cdots, 1]^T \in \mathbb{R}^n \tag{2.5}$$

and secondly that the result of a sequence $(x_k)_{k\in\mathbb{Z}^d}$, which is applied to the upper input of J connected synthesis parts of the lifting scheme, is constantly 1 if and only if the sequence $(x_k)_{k\in\mathbb{Z}^d}$ is constantly equal to 1 itself. We start by verifying the property (2.5) by proofing the following proposition. In the subsequent Proposition 2.8 we prove the second statement.

Proposition 2.5 *Consider the synthesis part of the lifting scheme with P being an arbitrary Neville filter of some order $N > 1$ and shift τ. Let $(\Phi_{0,k}(l))_{l\in\mathbb{Z}^d} := (\delta_{l-k})_{l\in\mathbb{Z}^d}$ and apply for all $k \in \mathbb{Z}^d$ these sequences to the upper input of J connected synthesis parts of the lifting scheme. Thus the result $(\Phi_{J,k}(l))_{l\in\mathbb{Z}^d}$ for all $k \in \mathbb{Z}^d$ is obtained by the recursion over $j \in 0{:}(J-1)$*

$$(\Phi_{j+1,k}(l))_{l\in\mathbb{Z}^d} = \Big((\uparrow D) + (\overrightarrow{t_1})(\uparrow D)P\Big)\,(\Phi_{j,k}(l))_{l\in\mathbb{Z}^d} \; , \tag{2.6}$$

cf. Figure 2.1. Then

$$\sum_{k\in\mathbb{Z}^d} \Phi_{J,k}(l) = 1 \quad \textit{for all} \quad l \in \mathbb{Z}^d \quad \textit{and all} \quad J \geq 0 \; .$$

Proof. The proof is by induction on J, where the case $J = 0$ directly holds because of the assumption $(\Phi_{0,k}(l))_{l\in\mathbb{Z}^d} = (\delta_{l-k})_{l\in\mathbb{Z}^d}$.

By equation (2.6) it holds that

$$\begin{aligned}(\Phi_{J+1,k}(l))_{l\in\mathbb{Z}^d} &= (\uparrow D)\,(\Phi_{J,k}(l))_{l\in\mathbb{Z}^d} + (\overrightarrow{t_1})(\uparrow D)P\,(\Phi_{J,k}(l))_{l\in\mathbb{Z}^d} \\ &= (\uparrow D)\,(\Phi_{J,k}(l))_{l\in\mathbb{Z}^d} + (\overrightarrow{t_1})(\uparrow D)\Big(\sum_{m\in\mathbb{Z}^d} p_{l-m}\Phi_{J,k}(m)\Big)_{l\in\mathbb{Z}^d} \; .\end{aligned}$$

Thus it holds that

$$\sum_{k\in\mathbb{Z}^d} \Phi_{J+1,k}(l) = \begin{cases} \sum_{k\in\mathbb{Z}^d} \Phi_{J,k}(D^{-1}l) & \text{if } D^{-1}l \in \mathbb{Z}^d \\ \sum_{k\in\mathbb{Z}^d}\sum_{m\in\mathbb{Z}^d} p_{D^{-1}(l-t_1)-m}\Phi_{J,k}(m) & \text{else} \end{cases} \; .$$

For the first case it holds that $\sum_{k\in\mathbb{Z}^d}\Phi_{J,k}(D^{-1}l)=1$, by induction hypothesis. For the second case the following applies

$$\sum_{k\in\mathbb{Z}^d}\sum_{m\in\mathbb{Z}^d} p_{D^{-1}(l-t_1)-m}\Phi_{J,k}(m)=\sum_{m\in\mathbb{Z}^d} p_{D^{-1}(l-t_1)-m}\sum_{k\in\mathbb{Z}^d}\Phi_{J,k}(m)\ ,$$

which by induction hypothesis is equal to

$$\sum_{m\in\mathbb{Z}^d} p_{D^{-1}(l-t_1)-m}\ ,$$

which in turn by Remark III.4.2 is equal to 1. □

Remark 2.6 *By Lemma 2.3 it holds that*

$$(\Phi_{J,k}(l))_{l\in\mathbb{Z}^d}=(\Phi_{J,0}(l-D^Jk))$$

and thus

$$\sum_{k\in\mathbb{Z}^d}\Phi_{J,0}(l-D^Jk)=1 \quad \textit{for all} \quad l\in\mathbb{Z}^d\ .$$

Remark 2.7 *Note that in Proposition 2.5 it is sufficient to consider P to be a FIR-Filter whose coefficients sum up to 1.*

Proposition 2.8 *Let $(x_k)_{k\in\mathbb{Z}^d}$ be a sequence and denote the result of this sequence applied to the upper input of J connected synthesis parts of the lifting scheme by $(y_k)_{k\in\mathbb{Z}^d}$. Denote a sequence which is constantly equal to 1 by $(e_k)_{k\in\mathbb{Z}^d}$. Then*

$$(x_k)_{k\in\mathbb{Z}^d}=(e_k)_{k\in\mathbb{Z}^d} \quad\Longleftrightarrow\quad (y_k)_{k\in\mathbb{Z}^d}=(e_k)_{k\in\mathbb{Z}^d}\ .$$

Proof. Let $(x_k)_{k\in\mathbb{Z}^d}=(e_k)_{k\in\mathbb{Z}^d}$, then for arbitrary $l\in\mathbb{Z}^d$

$$y_l=\sum_{k\in\mathbb{Z}^d}x_k\Phi_{J,k}(l)=\sum_{k\in\mathbb{Z}^d}\Phi_{J,k}(l)=1$$

by Proposition 2.5.

Now, let $y_k=1$ for all $k\in\mathbb{Z}^d$. Because of the interpolating property of the lifting scheme (cf. equation (2.3)) it holds that

$$y_{D^Jk}=x_k\ .$$

Hence $x_k=1$ for all $k\in\mathbb{Z}^d$. □

Thus by Proposition 2.5 it follows that $A_JE_n=E_m$. Now, assume that the scattered data has constant value, i.e., $s(x)=c$ for some $c\in\mathbb{R}$ and all $x\in\Xi$. Then the corresponding least squares problem is of the form

$$\min_{\mathbf{c}_J}\|A_J\mathbf{c}_J-cE_m\|_2^2\ . \tag{2.7}$$

Evidently, it would be natural that the approximation to this constant data is also constantly equal to c. By Proposition 2.5 and 2.8 this is the case if and only if the solution to the least squares problem (2.7) equals $\mathbf{c}_J=cE_n$. Now the results of Chapter I come into play, because the solution $\mathbf{c}_J=cE_n$ is obtainable either by all $\{1,3\}$-inverses $A^\natural$ that satisfy Problem I.3.1 with $E_n\in\mathcal{R}(Y)$, i.e., all $A^\natural$ with

$$A_J^\natural\mathbf{s}=\arg\min_{\mathbf{c}_J}\|A_J\mathbf{c}_J-\mathbf{s}\|_2 \quad\text{and}\quad A_J^\natural E_m=A_J^\natural(A_JE_n)=E_n\ ,$$

or by all Tikhonov regularizations

$$\min_{\mathbf{c}_J} \|A_J\mathbf{c}_J - \mathbf{s}\|_2^2 + \tau^2\|T\mathbf{c}_J\|_2^2 \, ,$$

with $\tau > 0$ and a regularization matrix T that satisfies $E_n \in \mathcal{N}(T)$ and $\mathcal{N}(A) \cap \mathcal{N}(T) = \{0\}$, cf. Problem I.4.1 and Proposition I.4.2.

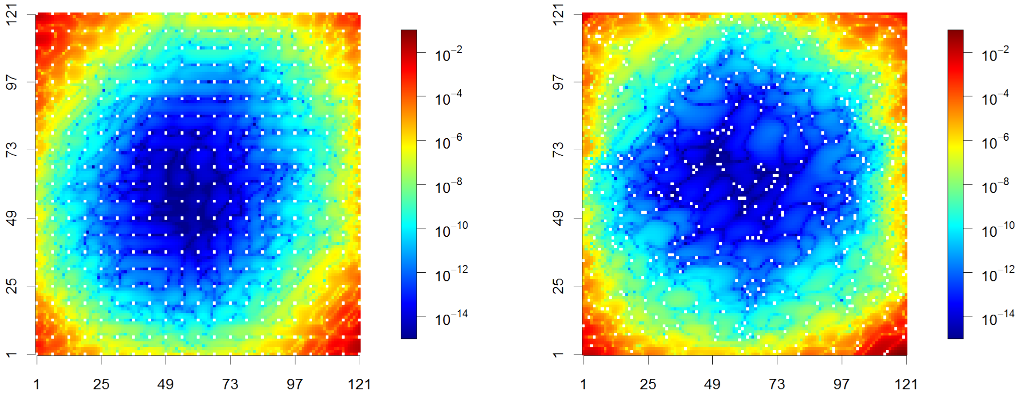
As we point out in the following two examples, the Moore–Penrose inverse does not necessarily satisfy Problem I.3.1. Additionally we learn that the use of the corresponding minimal norm solution can yield bad effects on the obtained approximation near the boundary of Ω_0.

Example 1: Approximation of constant data using the minimal norm solution

In the following example we choose $\Omega_0 = \{1{:}121 \times 1{:}121\}$ and $J = 6$. The filter P is chosen to be the Neville filter of order 4 derived in Chapter III, see Figure III.4.2. The purpose is to approximate scattered data that has constant value 1. We begin with two different distributions of points. In the first case we choose

$$\Xi_1 = \{x = (x_1, x_2) \in \Omega_0 : x_1 \equiv 1 \bmod 6 \quad \text{and} \quad x_2 \equiv 1 \bmod 6\} \ ,$$

i.e., the points $x \in \Omega_0$ that are equidistantly distributed with distance 6 in each direction. Hence $m = \#\Xi_1 = 441$. In the second case we sample randomly 441 points from a uniform distribution on Ω_0 and denote this set of points by Ξ_2. Let $s : \Omega_0 \to \mathbb{R}$ with $s \equiv 1$. To each set of the scattered data points we apply now the algorithm presented in Section 2.2, where we use the minimal norm solution to the corresponding least squares problem (2.7), with $c = 1$. In Figure 2.5 we present the result, where in Figure 2.5a the relative error between $s|_{\Omega_0}$ and the approximation $\mathbf{f}_1$ to the data $s|_{\Xi_1}$ is plotted. Similarly, in Figure 2.5b we depict the relative error between $s|_{\Omega_0}$ and the approximation $\mathbf{f}_2$ to the data $s|_{\Xi_2}$. In both plots the location of the scattered data sites, i.e., Ξ_1 and Ξ_2 respectively, is depicted by white dots. What can be seen in both cases is that in the middle of the domain Ω_0 the relative error is very small, but near the boundary it gets larger. The maximal relative error when approximating the equidistantly distributed data is around 2% and for the randomly distributed points around 10%. Since the matrix A_J differs for each different set of scattered data sites Ξ, we write in this paragraph $A_{J,\Xi}$ for a better distinction.

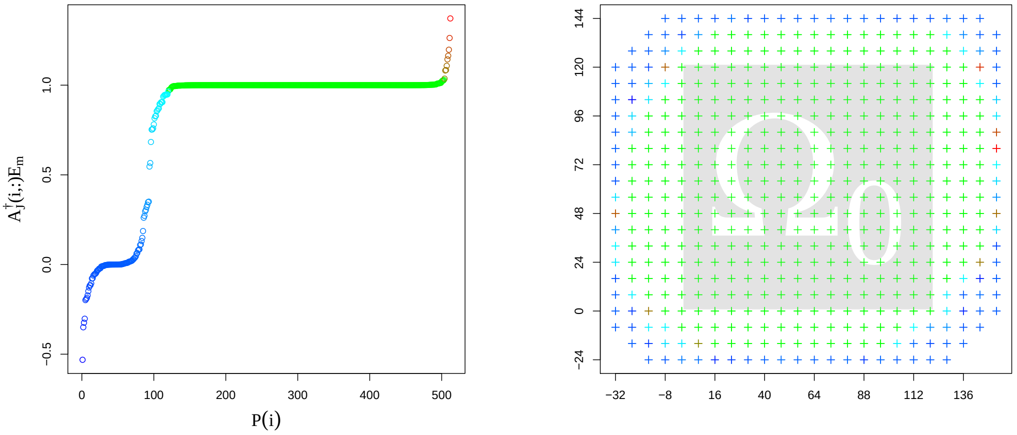


(a) Relative error between $s|_{\Omega_0}$ and $\mathbf{f}_1$

(b) Relative error between $s|_{\Omega_0}$ and $\mathbf{f}_2$

Figure 2.5: Approximation of scattered data $s|_{\Xi_1}$ and $s|_{\Xi_2}$ using the minimal norm solution

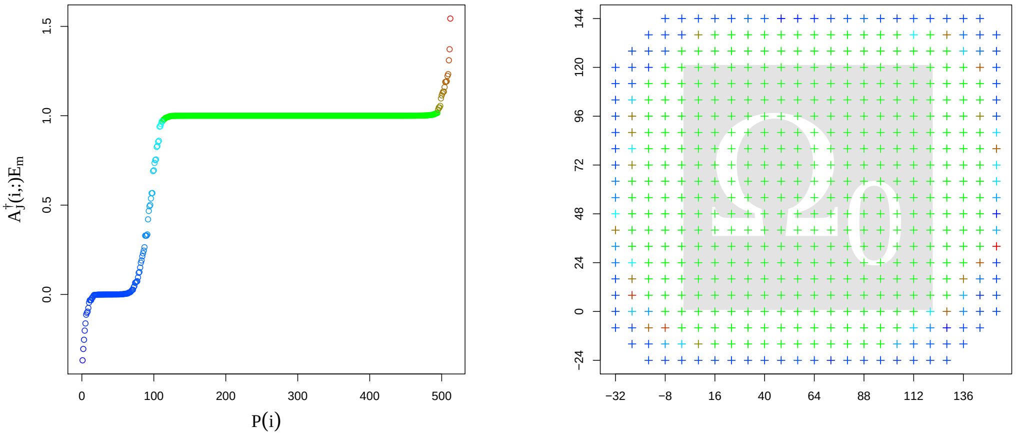
Naturally, the question arises why the minimal norm solution is not a good choice and why the effects are dominant at the boundary of the approximation. We answer the question by Figure 2.6. There in 2.6a the minimal norm solution $\mathbf{c}_J^\star = A_{J,\Xi_1}^\dagger s|_{\Xi_1}$ to the least squares problem $\min_{\mathbf{c}_J} \|A_{J,\Xi_1}\mathbf{c}_J - E_m\|_2^2$ is plotted, where the coefficients of $\mathbf{c}_J^\star$ have been rearranged in an ascending order. The first thing we notice is that $\mathbf{c}_J^\star$ is not constantly equal to 1. Thus by Proposition 2.8 the approximation cannot be constantly equal to $c = 1$. Recall that for each $k \in \Omega_J$ the coefficient $c_{J,k}^\star$ describes the influence of one single basis function $\Phi_{J,k}$ to the approximation, cf. Section 2.2 Step 2. In Figure 2.6b we mark the center of each basis function $\Phi_{J,k}$, i.e., the point $D_1^J k$, by a '+' sign where the color equals the color of its corresponding weight $c_{J,k}^\star$ from 2.6a. Thus in Figure 2.6b all basis functions that have an influence on values located at Ω_0 are represented. Moreover, in 2.6b the domain Ω_0 is indicated by a gray box. This makes clear which basis functions have its center inside or outside of Ω_0. The problem with the Moore–Penrose inverse within this approach is that it determines the solution to the least squares problem which has minimal norm. Because of the fast decay of the basis functions (see again Figure III.4.2) it is natural that the weights of the functions that have barely influence on values at Ω_0 get small values or are set to 0. This is compensated by weighting up other, but much fewer, functions. That circumstance can be well observed in Figure 2.6, where blue points outnumber red points by far. This explains why using the minimal norm solution results in the bad effects near the boundary of the approximations depicted in Figure 2.5. In Figure 2.7 we also provide the solution of the least squares problem $\min_{\mathbf{c}_J} \|A^{\dagger}_{J,\Xi_2}\mathbf{c}_J - E_m\|_2^2$ with respect to Ξ_2 and the presentation of the weights of the corresponding basis functions.



(a) $\mathbf{c}^{\star}_J = A^{\dagger}_{J,\Xi_1} E_m$ in ascending order

(b) Center of basis functions $\{D_1^J x : x \in \Omega_J\}$

Figure 2.6: Solution to the least squares problem $\min_{\mathbf{c}_J} \|A_{J,\Xi_1}\mathbf{c}_J - E_m\|_2^2$



(a) $\mathbf{c}^{\star}_J = A^{\dagger}_{J,\Xi_2} E_m$ in ascending order

(b) Center of basis functions $\{D_1^J x : x \in \Omega_J\}$

Figure 2.7: Solution to the least squares problem $\min_{\mathbf{c}_J} \|A_{J,\Xi_2}\mathbf{c}_J - E_m\|_2^2$

The effects at the boundary of the approximation become even worse when there is fewer data to approximate near the boundary as we depict in the following Figure 2.8 and 2.9, where Ξ_3 is a set of 300 random points sampled uniformly from Ω_0. Consider for instance the upper right corner of Figure 2.8. There no data to approximate are present and the relative error of the approximation is up to 150%. The reason for this can also be seen in Figure 2.9, where again the influence of each basis function is depicted.

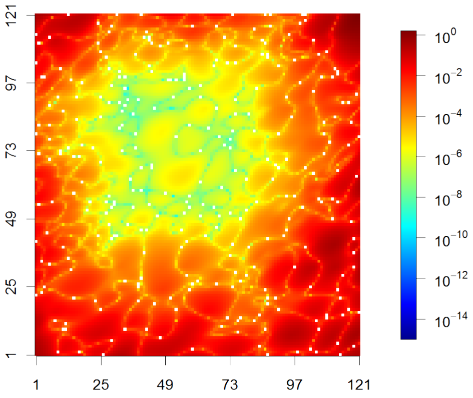
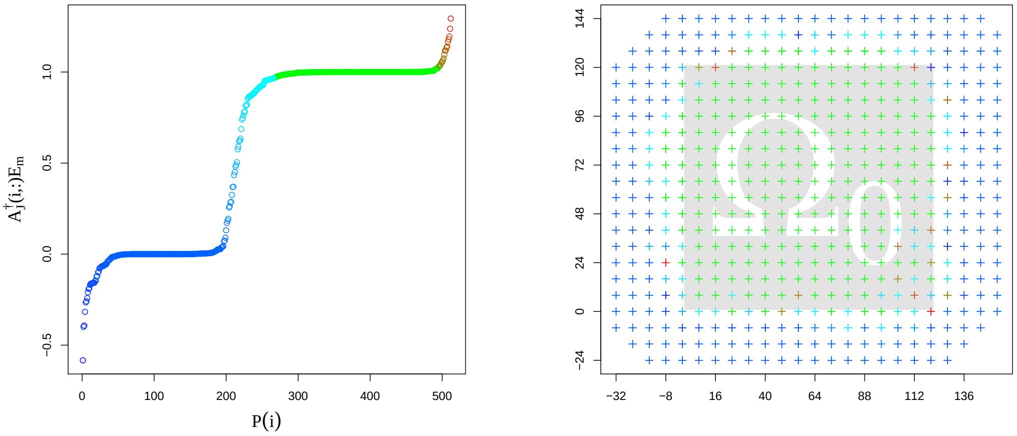


Figure 2.8: Relative error between $s|_{\Omega_0}$ and the approximation based on $s|_{\Xi_3}$



(a) $\mathbf{c}_J^\star = A_{J,\Xi_3}^\dagger E_m$ in ascending order

(b) Center of basis functions $\{D_1^J x : x \in \Omega_J\}$

Figure 2.9: Solution to the least squares problem $\min_{\mathbf{c}_J} \|A_{J,\Xi_3}\mathbf{c}_J - s|_{\Xi_3}\|_2^2$

Clearly, by construction, all solutions $A^\natural E_m$ with $A^\natural$ satisfying Problem I.3.1 with $Y = E_n$ yield an exact solution to all sets of constant valued scattered data sites. Of course also all Tikhonov regularizations with $E_n \in \mathcal{N}(T)$ and $\mathcal{N}(T) \cap \mathcal{N}(A) = \{0\}$ do, see Section I.4.

Example 2: Approximation of random samples from Franke's function using different solutions to the least squares problem

In the last preceding paragraph we saw that the use of the minimal norm solution within our approach does not guarantee exact approximation of constant valued scattered data. Moreover, we noticed that the effects or errors get dominant at the boundary of the approximant. In this paragraph we consider as scattered data random samples from a test function, which is choosen to be *Franke's function* [Fra79]. It is defined as a weighted sum of 4 exponentials

$$\begin{aligned}\mathrm{franke}(x,y) &= \frac{3}{4}e^{-((9x-2)^2+(9y-2)^2)/4}+\frac{3}{4}e^{-((9x+1)^2)/49-(9y+1)/10}\\ &\quad+\frac{1}{2}e^{-((9x-7)^2+(9y-3)^2)/4}-\frac{1}{5}e^{-(9x-4)^2-(9y-7)^2}\end{aligned}$$

and it is a standard choice for benchmarks on scattered data approximation. In this example the set Ξ of scattered data sites consists of 400 uniformly sampled random points from $\Omega_0 := \{1{:}201 \times 1{:}201\}$. The location of these sites is indicated within Figure 2.10 as white dots. In the subsequent paragraphs we compare approximations obtained from different solutions or regularizations to the least squares problem

$$\min_{\mathbf{c}_J} \|A_J\mathbf{c}_J - s|_\Xi\|_2^2\,, \tag{2.8}$$

where we consider:
a) the minimal norm solution, **b)** $(A_J)^{(1,3)}_{0,E_n}\mathbf{s}$, **c)** $(A_J)^{(1,3)}_{\mathcal{K},E_n}\mathbf{s}$, **d)+e)** two different regularizations using a discrete Laplacian as regularization matrix. We compute all solutions iteratively, where we use in all cases the GCV-method to compute the regularization parameter τ, see again Section I.2.2 and I.3.2. To better point out the differences between the different solutions we emphasize the effects near the boundary by adding a value of 30 to Franke's function, see Figure 2.10. The impact of the shift becomes clear in the example.

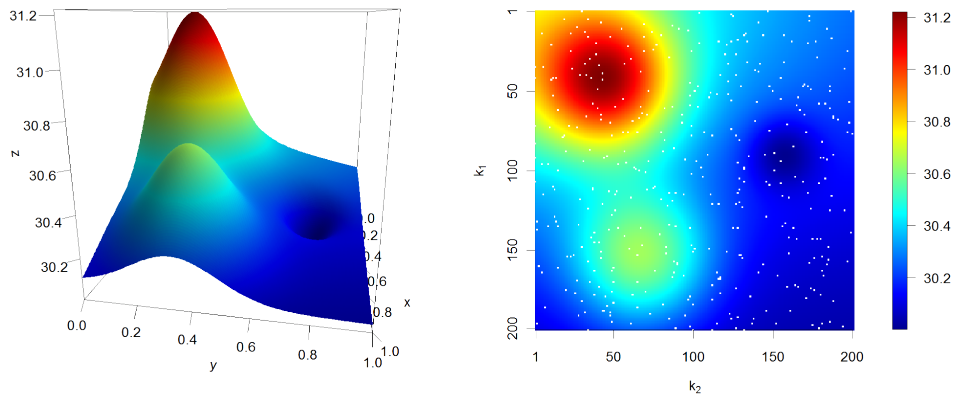


Figure 2.10: Franke's function on $[0,1]^2$ shifted by 30 in z-direction and the corresponding evaluation at Ω_0 with $x = (k_1-1)/200$ and $y = (k_2-1)/200$. White dots indicate Ξ.

Below we discuss the different solutions to the cases **a)-e)**. This is followed by Table 2.1, where the plots which correspond to the different results are depicted. For a better comparison of the different solutions we place all plots together in just one table, where each row of Table 2.1 is devoted to one case. Furthermore, the table consists of three columns, where in the first column the solutions or regularizations to the least squares problem (2.8) are presented. The white crosses × inside these figures mark the components of the solution whose corresponding basis functions have its center inside Ω_0.

In the second column of Table 2.1 the approximations which correspond to the solutions of the first column are depicted.

All figures in column one and two use the same coloring. Note that the values at the legend of the plots in the first column are linear between two labels but not overall.

The last column of the table contains the relative error plots, i.e., the relative error between Franke's function shifted by 30 and the approximation. Also, all error plots have the same scale, where for clarity we set all values that are smaller than 10^{-5} to 10^{-5}.

a) The minimal norm solution: As in the previous example, where we approximated constant valued scattered data, the minimal norm solution to (2.8) does not yield a good approximation. The argumentation for that reason stays also the same: The weights of the basis function whose center lie inside Ω_0 have acceptable values (marked by white ×). But the weights of the basis functions which have small influence to values located at Ω_0 can be found to be small or even 0, due to the fact that the 2-norm of the solution tends to be minimized. This in turn is compensated by a few components of the solution with higher values, which then results in the effects near the boundary. These effects can be well observed in the figures which are presented in the first row of Table 2.1.

b) Choosing $\left[(A_J)^{(1,3)}_{0,E_n}\mathbf{s}\right]$ as solution to (2.8)**:** In Section I.5 we stated that $(A_J)^{(1,3)}_{0,E_n}\mathbf{s}$ is the same as firstly computing the minimal norm solution to

$$\min_x \|A_J x - (\mathbf{s} - E_m^T\mathbf{s}/m)\|_2^2 \tag{2.9}$$

and adding $E_n E_m^T \mathbf{s}/m$ to the solution afterwards, where $m = \#\Xi$ and $n = \#\Omega_J$. In words, we substract the mean of $\mathbf{s}$ from $\mathbf{s}$, then compute the corresponding minimal norm solution and finally add the mean of $\mathbf{s}$ to the result again. This already explains why in this case the approximation has much less effects near the boundary: After subtracting the mean of $\mathbf{s}$ the minimal norm solution to (2.9) is indeed still zero at the boundary of Ω_J, but zero is now the mean of $\mathbf{s} - E_m^T\mathbf{s}/m$. Hence the minimal norm solution to (2.9) does not possess such huge compensations as the minimal norm solution in the standard case (2.8) does. This can also be well seen in the corresponding figures in Table 2.1. Note, because of adding again the mean of $\mathbf{s}$ to the minimal norm solution of (2.9), the overall solution $(A_J)^{(1,3)}_{0,E_n}\mathbf{s}$ near the boundary of Ω_J is equal to the mean of $\mathbf{s}$. This is because of the following:

Remark 2.9 *The lifting scheme is invariant under constant manipulation, i.e.,*

$$\sum_{k\in\Omega_J} c_{J,k}\Phi_{J,k}(l) = f_l$$

implies

$$\sum_{k\in\Omega_J} (c_{J,k} + \beta)\Phi_{J,k}(l) = f_l + \beta\ ,$$

for all $l \in \Omega_0$ and an arbitrary $\beta \in \mathbb{R}$. Evidently, this holds because of Proposition 2.5:

$$\sum_{k\in\Omega_J} (c_{J,k} + \beta)\Phi_{J,k}(l) = \sum_{k\in\Omega_J} c_{J,k}\Phi_{J,k}(l) + \beta \sum_{k\in\Omega_J} \Phi_{J,k}(l) = f_l + \beta\ .$$

c) Choosing $\left[(A_J)^{(1,3)}_{\mathcal{K},E_n}\mathbf{s}\right]$ **as solution to** (2.8)**:** In Section I.5 we already mentioned that $(A_J)^{(1,3)}_{\mathcal{K},E_n}\mathbf{s}$ can be seen as the limit $\tau \to 0$ of the following regularization

$$\min_{\mathbf{c}_J} \|A_J\mathbf{c}_J - \mathbf{s}\|_2^2 + \tau^2\|T\mathbf{c}_J\|_2^2 \quad \text{with} \quad T = I - \frac{E_nE_n^T}{n} \ , \tag{2.10}$$

see also Corollary I.3.17. Since $\frac{1}{n}\|T\mathbf{c}_J\|_2^2$ is equal to the variance of the components of $\mathbf{c}_J$, the regularization (2.10) balances the solution by keeping the variance of its components small. This can be seen in row **c)** of Table 2.1 where the solution to (2.10) has the same value at the boundary of Ω_J as the mean of the complete solution. This also explains why there are on the one hand almost no differences to the case **b)** above and on the other hand that there are few effects near the boundary of the approximation.

d) Regularization with T **as discrete Laplace operator and** $E_n \in \mathcal{N}(T)$ **:** In this case we use a Tikhonov regularization to the least squares problem (2.8), with the discrete Laplace operator as regularization matrix T. Choosing a Laplacian within a Tikhonov regularization is a common choice in two-dimensional smoothing and image restoration, see [Jen06, Chapter 5] and the references therein. Here, we choose the discrete Laplace operator with homogenous Neumann boundary conditions. Thus $T = (T_{k,l})_{k,l\in\Omega_J}$ with

$$T_{k,l} = \begin{cases} -1 & \text{if} \quad l \in \mathcal{U}(k) \\ \#\mathcal{U}(k) & \text{if} \quad l = k \\ 0 & \text{else} \end{cases} \qquad \text{where} \quad \mathcal{U}(k) := \{l \in \Omega_J : \|l-k\|_2 = 1\} \ . \tag{2.11}$$

Homogeneous Neumann boundary conditions mean that we have a free boundary, which can be observed very well at the corresponding figure in Table 2.1. A further advantage of the discrete Laplacian with homogenous boundary conditions is, besides its simplicity, the property $\mathcal{N}(T) = \mathcal{R}(E_n)$. On the one hand this property ensures that constant valued scattered data is approximated exactly, see Proposition 2.8 and I.4.2. On the other hand because of $\mathcal{N}(T) \cap \mathcal{N}(A) = \{0\}$ it ensures the uniqueness of the solution, see equation (2.5) and Section I.2.2.

e) Regularization with T **as discrete Laplace operator and** $E_n \notin \mathcal{N}(T)$ **:** This last case is to demonstrate that it is crucial to have $E_n \in \mathcal{N}(T)$. We choose $T = (T_{k,l})_{k,l\in\Omega_J}$ here slightly different than in **d)**, namely as discrete Laplace operator with homogenous Dirichlet boundary condition

$$T_{k,l} = \begin{cases} -1 & \text{if} \quad l \in \mathcal{U}(k) \\ 4 & \text{if} \quad l = k \\ 0 & \text{else} \end{cases} \qquad \text{with} \quad \mathcal{U}(k) := \{l \in \Omega_J : \|l-k\|_2 = 1\} \ . \tag{2.12}$$

Hence $E_n \notin \mathcal{N}(T) = \{0\}$ and the so regularized solution to (2.8) smoothly tends to 0 as it approaches the boundary of Ω_J, see the corresponding figure in Table 2.1. Forcing the solution to be 0 outside the boundary of Ω_J also explains the effects at the boundary of Ω_0 of the corresponding approximation.

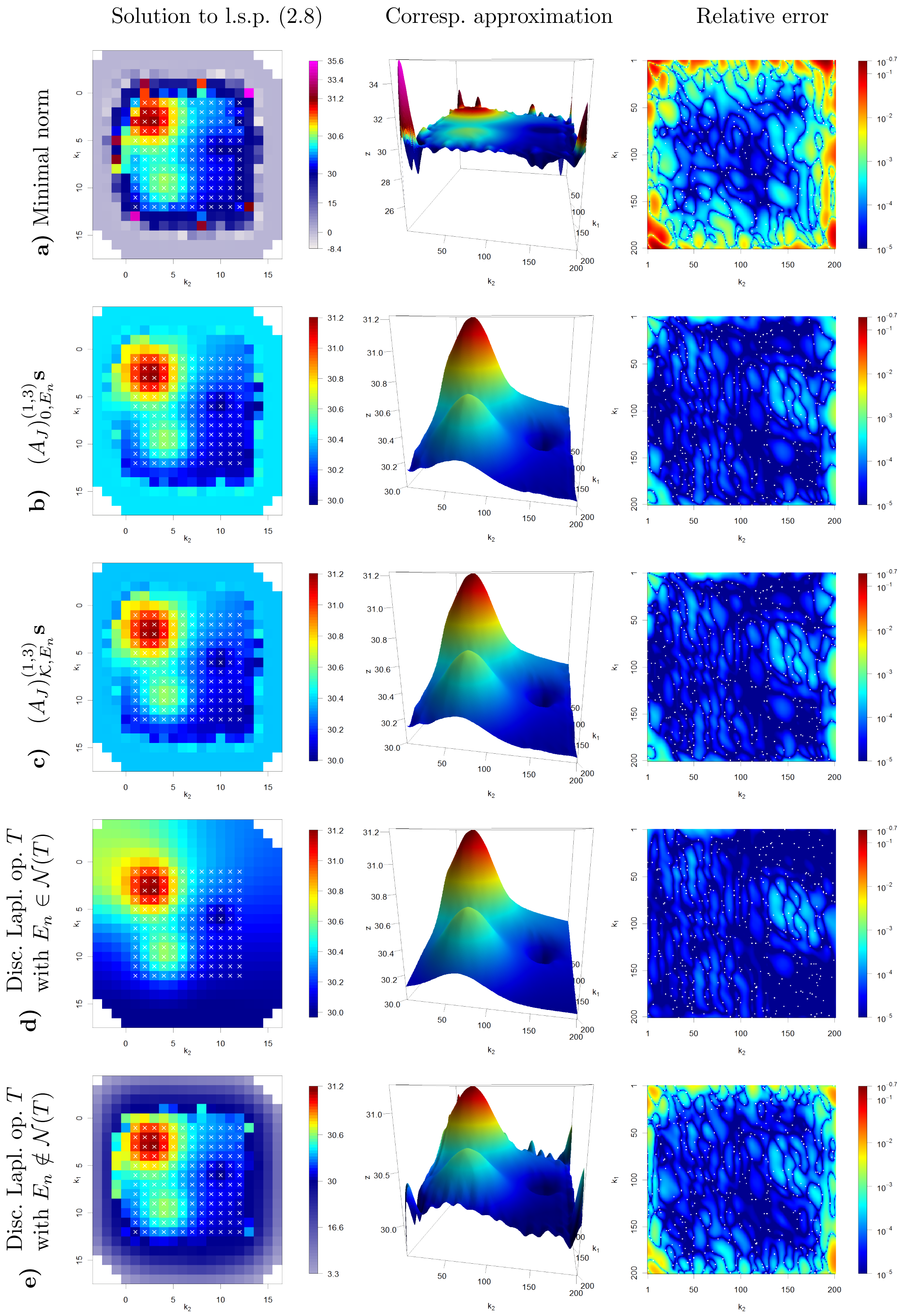


Table 2.1: Different solutions to the least squares problem and its corresponding approximations

Summarizing the different results from Table 2.1, we can easily state that in this particular example the use of the regularization with T chosen as discrete version of the Laplacian with free boundary delivered the best approximation to the scattered data among the different cases **a)-e)**. The difference between the two approximations which are obtained by using $A^{(1,3)}_{0,E_n}\mathbf{s}$ and $A^{(1,3)}_{\mathcal{K},E_n}$ is marginal. Moreover, in the interior of Ω_0 the relative error between Franke's function and these approximations is approximately the same as the relative error between Franke's function and the approximation obtained with the Laplacian **d)**. Only at the boundary there are some small effects, where the location of these effects corresponds to the places where few or none scattered data is available.

Below in Section 2.5 we put focus on the choice of J and the regularization to the least squares problem. Furthermore, we give a short note on the computational complexity, where we also introduce an approach to significantly reduce the numerical effort. Before that, we present the results of some more numerical experiments and compare our method to other, existing ones.

2.4 Numerical experiments

In this section we present some numerical experiments. We start by comparing our method to the approaches from [JSX09] and [CK05]. Then we present an inpainting example, where we restore an image from randomly sampled scattered data. In all experiments we use the regularization to the least squares problem with the discrete Laplace operator with homogenous Neumann boundary conditions as regularization matrix and the GCV-method to determine the regularization parameter τ.

Comparison of our method to [JSX09]

The approach to approximate scattered data in [JSX09] is, like our approach, also based on shift invariant spaces. In [JSX09] the method is designed for arbitrary dimension d and numerical experiments in the 1- and 2-dimensional case are presented on $\Omega = [0,1]$ or $\Omega = [0,1]^2$, respectively. In this paragraph we briefly explain the method from [JSX09] in the 2-dimensional case and compare benchmark results from [JSX09] to the results obtained by our method which we introduced in Section 2.2.

In [JSX09] the generator ϕ for the shift invariant space $S(\phi)$ is chosen as the tensor product of a cubic uniform B-spline, for more details on splines see, e.g., the classical book [Boo78]. Let

$$S^h(\phi,\Omega) := \{\sum_{k\in\mathbb{Z}^d} c_k\phi(\cdot/h-k) : c_k = 0 \;\text{ whenever }\; \operatorname{supp}\phi(\cdot/h-k)\cap\Omega = \emptyset\} \;,$$

which is the analog to the set $V_J(\Omega)$ defined in Section 1. Then, the scattered data $s|_\Xi$ is approximated by the solution of

$$\min_{f\in S^h(\phi,\Omega)} \sum_{x\in\Xi}(f(x)-s(x))^2 + \tau E(f) \;. \tag{2.13}$$

Thus, a regularized least squares problem is solved that also restricts the roughness of the solution. In [JSX09] the parameter τ is also determined by the GCV-method. The coefficients c_k of the approximant

$$f = \sum_k c_k\phi(\cdot/h-k)$$

are determined by solving a linear system of the form

$$(A^TA+\tau G)\mathbf{c} = A^T\mathbf{s} \;, \tag{2.14}$$

which stems from (2.13), using a conjugate gradient method. In [JSX09] it is suggested that for a faster convergence of the conjugate gradient method the solution should be determined in the wavelet domain. For this reason the method is called *WAVE*, for more details we refer to [JSX09]. In Section 2.5 we present a different approach to yield faster convergence of the conjugate gradient method.

In the 2-dimensional numerical experiments in [JSX09] three test functions

$$\begin{aligned} g_1(x,y) &= (-20.25(x-0.5)^2 + (y-0.5)^2)/3\ , \\ g_2(x,y) &= \frac{1.25+\cos(5.4y)}{6(1+(3x-1)^2)} \qquad \text{and} \\ g_3(x,y) &= \mathrm{franke}(x,y) \end{aligned}$$

are considered, where for each test function the following procedure is applied 50 times: First a set of scattered data sites Ξ is generated by uniformly sampling 400 random points from $\Omega = [0,1]^2$. Then the corresponding values of the scattered data sites are disturbed by adding Gaussian noise, i.e.,

$$s(x) = g_i(x) + \eta_x \ \text{ for } \ x \in \Xi \ \text{ and } \ \eta_x \sim N(0,\sigma_i^2) \ \text{ with } \ i \in \{1,2,3\}\ .$$

The data $(\Xi, s|_\Xi)$ is then subjected to approximation, where the quality of the approximation is measured by the *signal-to-noise ratio* (SNR), which is defined as

$$10\log_{10}\frac{\sum_{k\in\mathcal{X}} g_i(k)^2}{\sum_{k\in\mathcal{X}}(f(k)-g_i(k))^2}\ ,$$

where "higher" means "better". In [JSX09] $\mathcal{X}$ is chosen as $\{\frac{1}{200}(0{:}200) \times \frac{1}{200}(0{:}200)\}$ and the standard deviation σ_i is chosen such that the SNR of the noisy samples is about 20, which implies that $\sigma_1 = 0.01$, $\sigma_2 = 0.015$ and $\sigma_3 = 0.05$.

In [JSX09] the mean and the standard deviation of the SNR of the 50 different approximations obtained by the method WAVE is presented for each test function. The results are compared to the SNR obtained approximating the same data by thin-plate smoothing splines (TPSS, cf. Section 1). These results are presented in Table 2.2. In the numerical experiments in [JSX09] the scale parameter h is chosen such that the dimension of the linear system (2.14) is equal to 361 and thus close to the dimension of the system resulting from TPSS which is 400.

To make things also comparable to our method, we choose the scale parameter J in our method as $J = 8$ and $J = 9$, which results in $\#\Omega_8 = 420$ and $\#\Omega_9 = 280$ as system dimensions. Furthermore, we also choose a basis function of order 4 (see again Figure III.4.2), and $\Omega_0 = \mathcal{X}$. The results obtained by our method with these parameters can also be found in Table 2.2, where we abbreviate our method by *LIFT*.

Test function		LIFT ($J = 8$)		LIFT ($J = 9$)		WAVE		TPSS	
	σ	Mean	Std	Mean	Std	Mean	Std	Mean	Std
g_1	0.01	39.24	1.17	40.79	1.00	26.59	0.72	27.15	0.79
g_2	0.015	29.59	0.81	29.67	0.82	29.34	0.83	29.14	0.88
g_3	0.05	28.34	0.79	28.41	0.82	27.70	0.79	28.29	0.76

Table 2.2: Mean and std. of SNRs of our method (LIFT) and the one from [JSX09] (WAVE)

What we can conclude from Table 2.2 is that for g_2 and g_3 the results obtained by all three methods (LIFT, WAVE and TPSS) are similar. However, for g_1 our method yields significantly better results. The reason for this could be that the design of the filters from lifting scheme is based on polynomials and g_1 is itself a polynomial. Other numerical experiments (not presented here) with scattered data sampled from polynomials also yield comparable high SNR values. This observation is worth further research.

In [JSX09] it is not exactly stated how the boundary is treated, but in [JSX09] it is reported that WAVE behaves differently than TPSS near the boundary of Ω. In [JSX09] it is supposed that the reason therefore is that the smoothness penalty in TPSS is over $\mathbb{R}^2$ while WAVE restricts the smoothness penalty on Ω. It could be worth to consider in [JSX09] also the smoothness penalty on Ω_J as we have done it in our approach. This could also be content of further research as well as a thorough error analysis, which is also a main contribution in [JSX09].

Comparison of our method to [CK05]

In this paragraph we compare our method to the one from [CK05], which also is based on wavelets. We sketch the main idea of [CK05]: There one starts by approximating the scattered data by a function in $V_J = V_{J-1} \oplus W_{J-1}$, i.e., by

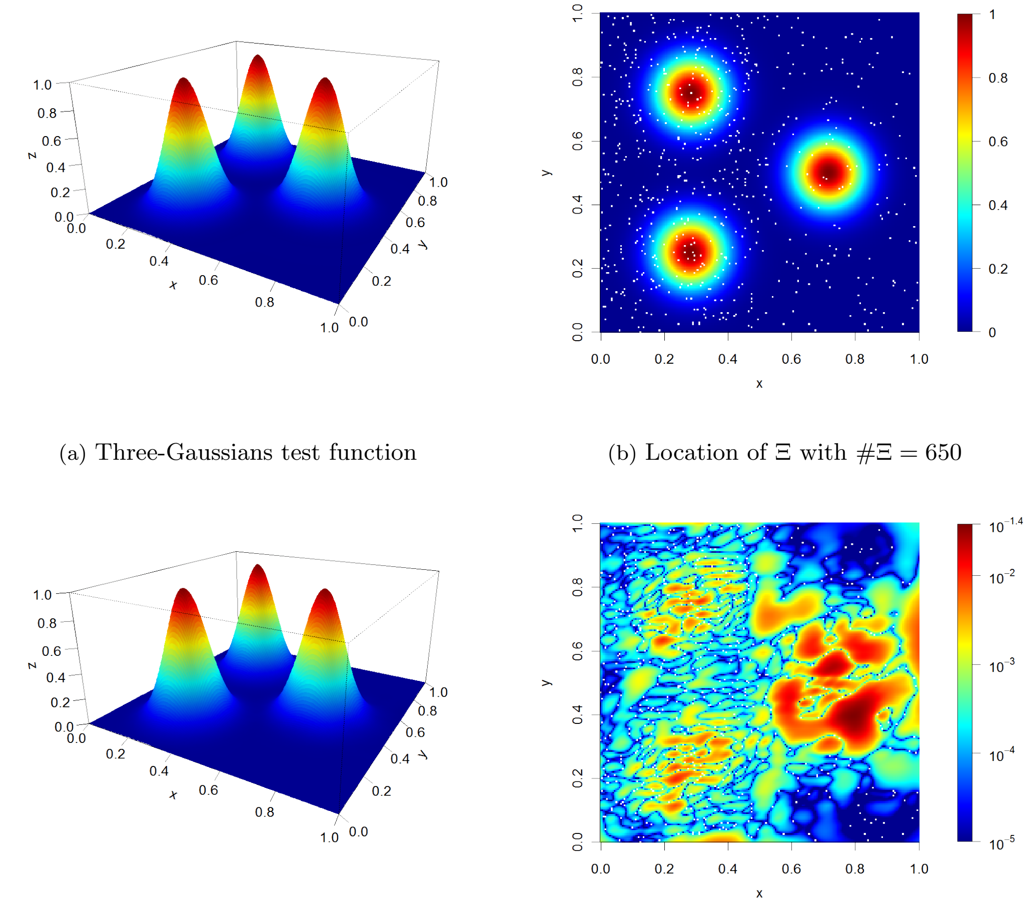
$$f = \sum_{k \in \mathbb{Z}^d} c_{J-1,k} \phi_{J-1,k} + \sum_{k \in \mathbb{Z}^d} d_{J-1,k} \psi_{J-1,k} \ . \tag{2.15}$$

Then also a regularized least squares fit to the scattered data is determined, where the corresponding system which is solved by a conjugate gradient method is of the form

$$(A^T A + \tau R)\mathbf{d} = A^T \mathbf{s} \ , \tag{2.16}$$

with $\mathbf{d}$ containing the coefficients $c_{J-1,k}$ and $d_{J-1,k}$ from (2.15). The matrix R in equation (2.16) is a differently weighted identity matrix. Thus E_n is not contained in the kernel of the space spanned by those rows of R which correspond to the scaling coefficients, which means that for $\tau > 0$ the approximation is not exact for constant data on the level J. Then the components of the solution $\mathbf{d}$ to (2.16) which correspond to the wavelet coefficients are tested. If they are above a certain threshold then in the neighborhood of these coefficients the resolution is increased, i.e., wavelets from lower levels are additionally considered, which then results in a wavelet tree structure. The main contribution in [CK05] is a multilevel version of the GCV-method to determine the regularization parameters τ efficiently for all steps inside the wavelet tree.

In the following experiment we choose a very similar setup as in [CK05, Figure 3], i.e., the test function from which the scattered data is generated is chosen to be three Gaussians (cf. Figure 2.11a where no explicit description of the test function is given). The scattered data sites are not distributed uniformly over Ω in this example. As in [CK05] we consider $\#\Xi = 650$ scattered data sites. We distribute 3/4 of it uniformly on the left half of Ω and 1/4 of it uniformly at the right half of Ω, see Figure 2.11b. We use the Neville filter of order 4 from Figure III.4.2 and $J = 6$.



(a) Three-Gaussians test function

(b) Location of Ξ with $\#\Xi = 650$

(c) Approximation obtained by our method

(d) Relative error between 2.11a and 2.11c

Figure 2.11: Approximation to non-uniformly distributed scattered data sites using our method

As we can conclude from Figure 2.11 our method is also capable to approximate scattered data that has a non-uniform distribution. The approximation 2.11c obtained by our method does not possess any visible effects, moreover it is hard to state a visual difference to the test function, whereas the approximations in [CK05] do all possess clearly visible effects, especially near the boundary (cf. [CK05, Figure 4, 8 and 9]). In [CK05] the quality of the approximation is measured by

$$Q(f) := \sqrt{\sum_{x\in\Xi} \left(f(x) - s(x)\right)^2}\,,$$

with f being the approximation. In [CK05] $Q(f)$ is around 0.7, whereas the result of our experiment from Figure 2.11 is $Q(f) = 0.011$. Even though the experiment is not exactly the same, the difference of a factor of more than 60 and the fact that in contrast to [CK05] there are no visible effects at the approximation 2.11c indicates the advantage of our method in this particular experiment.

Remark 2.10 *Note that the measure SNR used in [JSX09] is more appropriate than $Q(f)$ since the approximation*

$$f(x) = \begin{cases} s(x) & \textit{if } x \in \Xi \\ 0 & \textit{else} \end{cases}$$

yields $Q(f) = 0$ but is most likely not the result one is interested in.

Small example: Application of our method to an inpainting problem

Inpainting is a standard problem in image processing, where the purpose is to reconstruct missing or corrupted parts of an image, see, e.g., [BSCB00]. A popular class of methods to tackle inpainting problems is total variation inpainting – short TV-inpainting. These methods usually apply convex optimization or PDE-based diffusion algorithms, see [BHS09] for a recent approach.

In this last numerical experiment we apply our method (LIFT) to 8% uniformly sampled pixels (3200) from a 200 × 200 part of the Lena-image. The original image and the sampled pixels which are subjected to inpainting are shown in Figure 2.12a and 2.12b, respectively. The result of our approach to this data is presented in Figure 2.12c, where we used the filter from Figure III.4.3 and $J = 3$. We also applied the recent PDE-based approach [BHS09] which is available as MATLAB-code [Sch12]. The result is depicted in Figure 2.12d. For both results we also computed the *peak signal-to-noise ratio* (PSNR), a standard measure in image processing, which is defined for monochrome images as

$$10 \log_{10} \frac{255^2}{\sum\limits_{k \in \mathcal{X}} (\mathcal{I}(k) - f(k))^2} ,$$

with $\mathcal{I}$ being the original image with values in $[0, 255]$ on the grid $\mathcal{X} \subset \mathbb{Z}^2$ and with f being its reconstruction. The approach [BHS09]/[Sch12] and our approach yield very similar PSNR-values where the result obtained by [Sch12] has sharper contours while our approach is more detailed, for instance in the proximity of the eyes and nose. This brief example demonstrates that our method also delivers respectable results when applying it to an inpainting problem with scattered pixels.

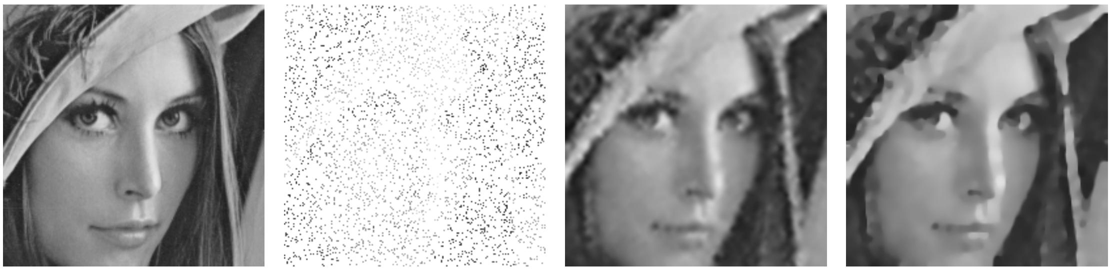

(a) Part of Lena-image (b) 8% sampled pixels (c) LIFT PSNR=25.01 (d) [Sch12] PSNR=25.08

Figure 2.12: Inpainting applying LIFT and [Sch12]

2.5 More on the method

In this section we discuss more on our method, which we introduced in Section 2.2 and also suggest at some points what can be further done in future work. Moreover, we present below an idea to reduce the numerical effort.

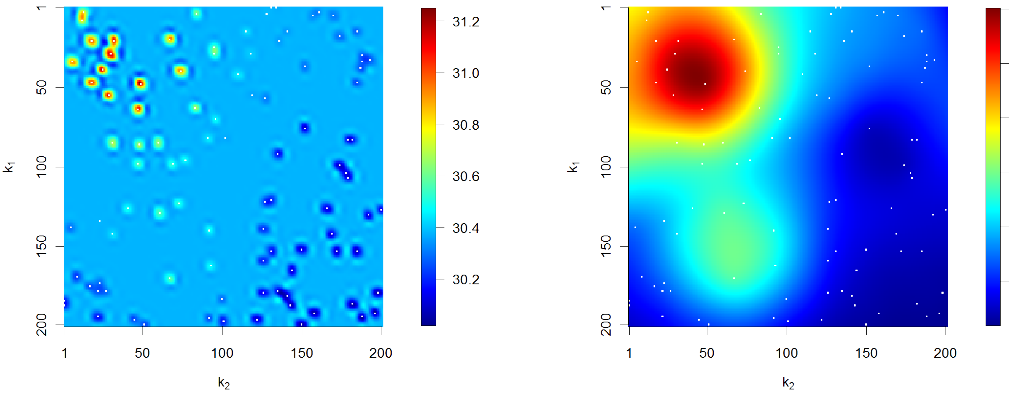
On the scaling parameter J and the solution/regularization to (2.4)

Evidently, the support of the basis functions $\phi_{J,k}$ is dependent on J: when J increases then also the support increases and vice versa. This property restricts the application of the solutions $A^{(1,3)}_{0,E_n}\mathbf{s}$ and $A^{(1,3)}_{\mathcal{K},E_n}\mathbf{s}$. Consider for instance the solution $A^{(1,3)}_{0,E_n}\mathbf{s}$, which is up to a shift a minimal norm solution as we have seen in Section I.5 and 2.3. Hence, if the support of the scaling functions is too small, the solution $A^{(1,3)}_{0,E_n}\mathbf{s}$ only weights those basis functions whose support intersects with an $x \in \Xi$ in order to keep the norm small. We demonstrate this effect in Figure 2.13a where we approximated 100 scattered data points sampled from Franke's function using $A^{(1,3)}_{0,E_n}\mathbf{s}$ as solution to the least squares problem. To obtain solutions $A^{(1,3)}_{0,E_n}\mathbf{s}$ and $A^{(1,3)}_{\mathcal{K},E_n}\mathbf{s}$ where all its components have an influence on the approximant, one has to choose J such that the support of each basis function has an intersection with an $x \in \Xi$. This implies that the solutions $A^{(1,3)}_{0,E_n}\mathbf{s}$ and $A^{(1,3)}_{\mathcal{K},E_n}\mathbf{s}$ should only be applied if the scattered data is scattered uniformly, or in case that one is just interested in a very coarse approximation. However also these solutions can yield respectable results, as we saw in Section 2.3 and as we demonstrate in Table 2.3 below. There we present the results of approximating the same data within the same setup as in the comparison to [JSX09] in Section 2.4 with $J = 10$.

Test function		$A^{(1,3)}_{0,E_n}\mathbf{s}$		$A^{(1,3)}_{\mathcal{K},E_n}\mathbf{s}$	
	σ	Mean	Std	Mean	Std
g_1	0.01	37.58	2.03	37.82	2.09
g_2	0.015	27.91	0.91	27.91	0.87
g_3	0.05	26.38	0.80	26.43	0.81

Table 2.3: Mean and std. of SNRs of our method (LIFT) using $A^{(1,3)}_{0,E_n}\mathbf{s}$ and $A^{(1,3)}_{\mathcal{K},E_n}\mathbf{s}$

Using a regularization with the discrete Laplace operator to the least squares problem restricts the roughness of the solution. Therefore it can also be applied in case that not every basis function intersects with an $x \in \Xi$. This is indicated in Figure 2.13b, where the same data as in Figure 2.13a is approximated, but with the discrete Laplace operator as regularization matrix. Moreover, using the discrete Laplacian as regularization matrix has the advantage that the solution is always unique, as we stated above in Section 2.3. This is not necessarily the case when one restricts the roughness of the approximant to Ω as it is done for instance in [JSX09] and [CK05]. There mild conditions on the distribution of the scattered data sites and the basis functions have to be posed to guarantee uniqueness of the solution, see [JSX09]. Applying the regularization directly to the coefficients $(c_{J,k})_{k\in\Omega_J}$ as it is done within our approach also restricts the roughness of the approximant because $c_{J,k}$ is the evaluation of the approximant at the positions $l = D^J k$ (if $l \in \Omega_0$), see Remark 2.4 and equation (2.3). Hence the discrete regularization works with a larger step-size than the step-size on Ω_0, but I conjecture that the smoothness of the coefficients $(c_{J,k})_{k\in\Omega_J}$ is due to the interpolating property of the lifting scheme directly connected to the smoothness of the corresponding approximant, this has to be investigated in future research.



(a) Using $(A_J)^{(1,3)}_{0,E_n}\mathbf{s}$

(b) Using Laplacian regularization

Figure 2.13: Two different approximations to 100 scattered data points sampled from Franke's function in case $J = 4$

What about the lower input?

In our method we only consider the upper input of the J connected synthesis parts of the lifting scheme, see Figure 2.1. Clearly, it is also possible to use some of the lower inputs. This would then correspond to additionally determine wavelet coefficients $d_{j,k}$ whose use can result in a more detailed approximation of the scattered data. Additional consideration of the wavelet coefficients is done in the above mentioned papers [FE98], [NM99] and [BLC04]. Since in all these papers the minimal norm solution to the least squares problem is used, the scale parameter J is restricted in these methods and has to be chosen such that the scaling functions $\phi_{J,k}$ have large enough support, see again the explanation from the last preceding paragraph. Hence the only way to get a more detailed approximation with these methods is to use also the lower input at step J. What actually is done in these methods, is that first an approximant f to the scattered data is determined by the coefficients $(c_{J,k})_{k\in\Omega_J}$. Then the difference between the approximant at the scattered data sites with the scattered data is taken, i.e., $e(x) := f(x) - s(x)$ for $x \in \Xi$ and then $(d_{J,k})_{k\in\tilde{\Omega}_J}$ is determined such that $\|B_J\mathbf{d}_J - e|_\Xi\|_2^2$ gets minimized, where B_J describes the influence of $\mathbf{d}_J$ to values located at the sites Ξ, and $\tilde{\Omega}_J$ is the set of indices k such that the support of the corresponding wavelet $\psi_{J,k}$ intersects with Ω_0.

In our case when using the regularization with T as the discrete Laplace operator (2.11) to the least squares problem we are not limited to some J. Hence, to obtain a more detailed approximant we can just reduce J and thus do not explicitly need to compute the wavelet coefficients. This is because of the decomposition $V_{J-1} = V_J \oplus W_J$, which means that $f \in V_{J-1}$ can be expressed as

$$f = \sum_{k\in\mathbb{Z}^d} c_{J-1}\phi_{J-1,k} = \sum_{k\in\mathbb{Z}^d} c_{J,k}\phi_{J,k} + \sum_{k\in\mathbb{Z}^d} d_{J,k}\psi_{J,k} \ ,$$

see again Section III.2.1 and the rest of Chapter III.

In the approach of [JSX09] the complete wavelet decomposition is exploited for a faster convergence of the conjugate gradient method. Below in this section we introduce a different ansatz to obtain a faster convergence of the conjugate gradient method. However, additional consideration of lower inputs within our approach could be investigated in the future.

How to handle scattered data and how to evaluate the approximant on $\mathrm{conv}(\Omega_0) \setminus \Omega_0$ for $d = 2$

Using our method to approximate scattered data requires $\Omega = \Omega_0$ to be a bounded subset of $\mathbb{Z}^2$, see again Remark 1.1. In most image processing applications this restriction is acceptable since the grid Ω of some digital image is evidently a bounded subset of $\mathbb{Z}^2$. However in some applications, like image superresolution, it might be useful to handle scattered data sites that are between two points and thus not directly located at Ω_0.

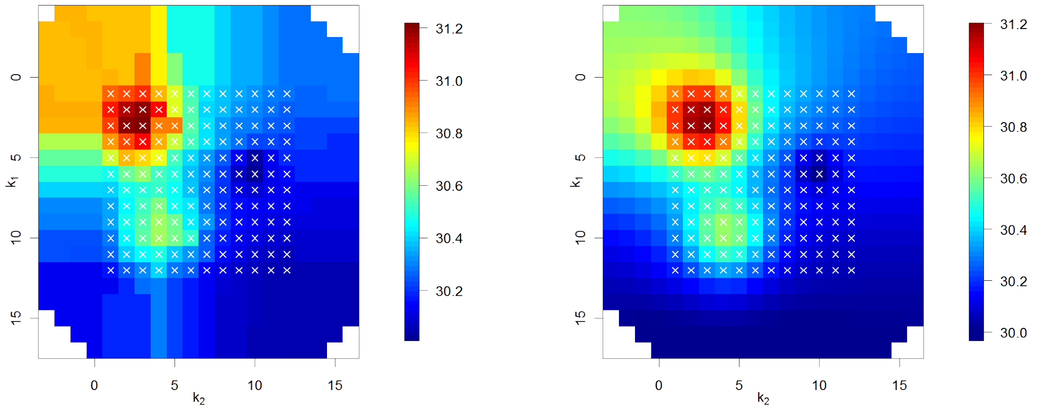
Assume that $\Omega_0 \subset \mathbb{Z}^2$ and the scale parameter J are fixed and let $x \in \Xi$ with $\Xi \subset \mathrm{conv}(\Omega_0)$. To set up the least squares problem we need evaluations of $\Phi_{J,k}$ at $x \in \Xi$ for all $k \in \Omega_J$. If $x \in \mathrm{conv}(\Omega_0) \setminus \Omega_0$ this is not possible straight away. But one can exploit Remark 2.4 and the fact that $D_1^2 = 2I$ (see again Figure III.3.1) stretches by a factor of 2 in each direction. Let $n \in \mathbb{Z}_+$ then evaluation at $l \in \mathrm{conv}(\Omega_0) \cap 2^{-n}\mathbb{Z}^2$ is possible via $\Phi_{J+2n,k}(2^n l)$.

On the other hand this idea is also applicable in the case that the coefficients $(c_{J,k})_{k \in \Omega_J}$ are already determined and one is interested to evaluate the corresponding approximant at some $x \in \mathrm{conv}(\Omega_0) \setminus \Omega_0$.

Short note on the numerical effort

So far little energy has been spent on investigating the numerical effort of our method. Surely, the main effort is to solve the regularized least squares problem from step 3 and the determination of the corresponding regularization parameter τ. The detailed consideration of the computational complexity is subject of future work. Nevertheless, we present some numbers to orientate on and a suggestion on how to reduce the numerical effort. Therefore we consider again the setup from Example 2 case d) in Section 2.3. We apply the routine *cgs* from MATLAB [MAT10] without preconditioner and relative residual tolerance 10^{-6} to the normal equation (I.2.14) which results from the regularized least squares problem. The cgs-routine takes 243 iterations to converge to the solution depicted in Figure 2.14b, where white $\times$ indicate again the components of the solution whose corresponding basis functions have its center inside Ω_0.

We introduce now an ansatz to significantly reduce the amount of iterations. We therefore exploit – once more – the interpolating property of the lifting scheme (2.3) and thereby provide a good initial guess to the iterative solver. The initial guess is built in the following way: For each $k \in \Omega_J$ the initial $c_{J,k}$ is set to the value $s(x)$, with $x \in \Xi$ being the scattered data site which is closest to the point $D^J k$ within the Euclidean distance. This can be cheaply accomplished by a nearest neighbor search. Applying this recipe results in the initial guess presented in Figure 2.14a. Evidently, this initial guess has already significant similarity to the solution displayed in Figure 2.14b and thus it is hardly surprising that with this initial guess the cgs-routine takes only 12 iterations to converge compared to the 243 iterations in the original setting without initial guess.



(a) Initial guess $\mathbf{c}_8$

(b) Solution $\mathbf{c}_8^\star$

Figure 2.14: Determination of an initial guess exploiting the interpolating property of the lifting scheme using nearest neighbor search – applied to the data from Example 2 d) in Section 2.3

In the following Table 2.4 we present results obtained on a 2GHz machine with 2GB of RAM applying the cgs-routine to the same data as in Example 2 case d) in Section 2.3 for $J \in \{4, 6, 8, 10\}$.

Scale parameter J	10	8	6	4
System dimension	208	424	1072	3232
Sparsity of sys. in %	47.8	71.2	90.0	98.3
Number of iterations				
without initial guess	246	243	222	252
with initial guess	14	12	12	21
Computing time in seconds				
without initial guess	0.32	0.48	3.98	30.45
with initial guess	0.04	0.04	0.26	2.73
for the initial guess	< 0.01	< 0.01	≈ 0.01	≈ 0.03

Table 2.4: Experiment on the numerical effort applying the cgs-routine with relative residual tolerance 10^{-6}

What we can conclude from Table 2.4 is firstly that the computation of the initial guess is very cheap, secondly the cgs-routine converges more than 10 times faster with this initial guess and thirdly the number of iterations remains pretty much constant with increasing system dimension.

Remark 2.11 *We chose here a standard MATLAB routine on the normal equations to give numbers one can compare with. The reason why we considered the cgs-routine was because it was the fastest within MATLAB's standard routines in our particular examples. In Section I.2.2 we mentioned the CGLS algorithm, which also seems preferable here, since one saves the* *computational effort of determining the normal equations. Finding the routine which fits best to our method can also be seen as future research.*

To determine the regularization parameter τ with the GCV-method one has to minimize a functional, whose evaluation costs 2 solutions to a Tikhonov regularization, see again Section I.2.2. Finding the minium of this functional can for instance be done by a Newton-type algorithm or the Nelder–Mead method [NM65], which is used in MATLAB's *fminsearch*-routine. In most numerical experiments that we performed a starting value of 10^{-3} was a good initial value from which MATLAB's *fminsearch*-routine needed between 1 and 5 iterations to converge within standard tolerances.

Summary and Future Research

In the following list the main results of this thesis are summarized:

- In Section I.3, more precisely in Theorem I.3.2, we characterize all matrices $A^\natural$ that satisfy Problem I.3.1, i.e., a special subset of $\{1,3\}$-inverses $A^\natural$ with partially prescribed image spaces, i.e., all $A^\natural$ that satisfy $A^\natural(AY) = Y$ in case that $A \in \mathbb{C}^{m\times n}$, $Y \in \mathbb{C}^{n\times \ell}$ and $\operatorname{rank} AY = \ell$.
- In Theorem I.3.5 we show that one can obtain particular generalized inverses $A^\natural$, satisfying Problem I.3.1, that coincide with the Moore–Penrose inverse on certain subspaces of $\mathbb{C}^m$. This result yields the two natural choices $A_{Y,0}^{(1,3)}$ and $A_{Y,\mathcal{K}}^{(1,3)}$, cf. (I.3.5) and (I.3.6).
- In Section I.3.2 we prove that the two choices $A_{Y,0}^{(1,3)}$ and $A_{Y,\mathcal{K}}^{(1,3)}$ are robust in the sense of cropping of singular values, cf. Proposition I.3.15. Moreover we reveal how the solutions $A_{Y,0}^{(1,3)}b$ and $A_{Y,\mathcal{K}}^{(1,3)}b$ to the least squares problem can be efficiently computed.
- In Section I.4 we reformulate Problem I.3.1 to the case when using Tikhonov regularizations, see Problem I.4.1. Moreover we prove that in the limit $\tau \to 0$ all solutions to a Tikhonov regularization can also be obtained by a $\{1,2,3\}$-inverse.
- In Section II.2.4 we introduce a new characterization of a class of correct sets, see Definition II.2.11 and Theorem II.2.12. Moreover, we prove that this class is more general than the class of fully generalized principal lattices, see Theorem II.2.13.
- In Section II.2.4 we also introduce a new and concrete recipe that yields correct sets, see Definition II.2.15, where we also present an alternative proof, see Theorem II.2.17.
- In Section III.4.1 we construct a new family of Neville Filters in the two-dimensional Quincunx case which have fewer filter coefficients than existing ones see Figure III.4.5. Moreover we numerically verify that all filters induce stable scaling functions.
- In Theorem III.4.4 we introduce a result on a geometrical configuration, which yields many zero filter coefficients. Exploiting this result yields a two-dimensional extension of the one-dimensional Deslauriers–Dubuc filters, see Figure III.4.8.
- In Section IV.2.2 we introduce a method to approximate scattered data, which is based on the lifting scheme. By Propositions IV.2.5 and IV.2.8 we prove that constant valued scattered data is approximated exactly if and only if one uses solutions or regularizations to the least squares problem which are based on $\{1,3\}$-inverses or Tikhonov regularizations satisfying Problem I.3.1 or Problem I.4.1 with $Y = E_n$, respectively.
- In Section IV.2.4 we compare our method to similar, existing, ones by performing several numerical experiments and show that our method delivers similar or even better results.

- In Section IV.2.5 we present an idea that is based on the interpolating property of the lifting scheme and a nearest neighbor search which significantly reduces the number of iterations of a conjugate gradient method applied to the normal equations of the least squares problem that has to be minimized in our approach.

We see the main potential for further research in our method to approximate scattered data, where problems to be considered in future work could be:

- A thorough error analysis, as it is done in [JSX09], for our method. Most likely the concepts used in [JSX09] can be applied one-to-one to our case since the theoretical background of both methods is the same.
- Investigating if the "smoothness" of the coefficients $(c_{J,k})_{k\in\Omega_J}$ is directly connected to the smoothness of the corresponding approximation $(\sum_{k\in\Omega_J} c_{J,k}\Phi_{J,k}(l))_{l\in\Omega_0}$, due to the interpolating property of the lifting scheme.
- Additional consideration of lower inputs could be investigated and whether their use results, as in the approach of [JSX09], in a faster convergence of the conjugate gradient method.
- A detailed study on the computational complexity.

Glossary of Notation

1:N – $\{1,\dots,N\}$, 23

$A^\dagger$ – Moore–Penrose inverse of A, 6
$A^\natural$ – generalized inverse of A, satisfying Problem I.3.1, 13
$A^{(i,j,\dots,k)}$ – an $\{i,j,\dots,k\}$-inverse of A, 6
$A\{i,j,\dots,k\}$ – set of all $\{i,j,\dots,k\}$-inverses of A, 6
$A^{(1,3)}_{Y,K}$, 13
A^T – transposed of A, 17
A^* – conjugate transposed of matrix A, 6
$A|_U$ – restriction of A to U, 7

$\mathbb{C}$ – field of complex numbers, 6
$\mathbb{C}^{m\times n}$ – space of $m \times n$ complex matrices, 6
$\mathbb{C}^m$ – $\mathbb{C}^{m\times 1}$, 7
conv$X$ – convex hull of X, 95

$\deg p$ – degree of a polynomial p, 28
$\delta_{k,l}$ – Kronecker delta, 42
δ_k – $\delta_{0,k}$, 76
$\det(A)$ – determinant of square matrix A, 40

$\mathbb{F}$ – either $\mathbb{R}$ or $\mathbb{C}$, 27
$\flat(X)$ – affine hull of X, 32

$\Gamma_{n,d}$ – set of multi-indices, 27
$\tilde{\Gamma}_{n,d}$ – set of homogenized multi-indices, 30

H^* – adjoint of a filter H, 48

$\mathcal{L}(U,V)$ – linear transformation from U to V, 7
$L^2(\mathbb{R}^d)$ – space of square integrable functions, 39

$l^2(\mathbb{Z}^d)$ – space of square summable sequences, 41

$N(0,\sigma^2)$ – normal distribution with standard deviation σ and mean 0, 89
$\mathcal{N}(A)$ – kernel of A, 7

P_U – orthogonal projector on U, 8
Π^d – polynomial ring in d variables, 27
Π^d_n – space of d-variate polynomials with $\deg p \le n$, 28

$q(f(\mathbb{Z}^d))$ – polynomial q evaluated at $f(k)\ \forall\ k \in \mathbb{Z}^d$, 46

$\mathbb{R}$ – field of real numbers, 27
$\mathbb{R}^{m\times n}$ – space of $m \times n$ real matrices, 12
$\mathbb{R}^m$ – $\mathbb{R}^{m\times 1}$, 25
$\mathcal{R}(A)$ – range of A, 7

supp$f$ – support of function f, 32
$\#X$ – cardinality of the set X, 28

$(\overrightarrow{t})$ – shift of a signal by t, 47
$\tau^\dagger$ – Moore–Penrose inverse of scalar τ, 6

$U^\perp$ – orthogonal complement of U, 7
$(\uparrow D)$, $(\downarrow D)$ – up- and down-sampling with dilation matrix D, 46

$x \perp y$ – x perpendicular to y, 7
$\langle x, y\rangle$ – inner product of x and y, 7

$\mathbb{Z}$ – set of integers, 30
$\mathbb{Z}_+$ – set of nonnegative integers, 27

Bibliography

[Apo11] A. Apozyan. *Multivariate polynomial interpolation. GC_n-sets and Gasca-Maeztu.* PhD thesis, Institute of the NAS RA, 2011.

[BDR94] C. de Boor, R.A. DeVore, and A. Ron. Approximation from shift-invariant subspaces of $L_2(\mathbb{R}^d)$. *Trans. Amer. Math. Soc.*, 341(2):787–806, 1994.

[BHS09] M. Burger, L. He, and C.B. Schönlieb. Cahn-Hilliard inpainting and a generalization for grayvalue images. *SIAM J. Imaging Sci.*, 2(4):1129–1167, 2009.

[BI02] A. Ben-Israel. The Moore of the Moore–Penrose inverse. *Electron. J. Linear Algebra*, 9:150–157, 2002.

[Bie03] O. Biermann. Über näherungsweise Cubaturen. *Monatshefte für Mathematik*, 14:211–225, 1903.

[BIG03] A. Ben-Israel and T.N.E. Greville. *Generalized Inverses: Theory and Applications.* CMS books in mathematics. Springer, 2nd edition, 2003.

[Bje58] A. Bjerhammar. A generalized matrix algebra. *Trans. Roy. Inst. Tech. Stockholm*, 52, 1958.

[Bjö96] A. Björck. *Numerical Methods for Least Squares Problems.* SIAM, 1996.

[BLC04] N.K. Bose, S. Lertrattanapanich, and M.B. Chappalli. Superresolution with second generation wavelets. *Sig. Proc.: Image Comm.*, 19(5):387–391, 2004.

[Boo78] C. de Boor. *A practical guide to splines.* Springer, 1978.

[Boo07] C. de Boor. Multivariate polynomial interpolation: conjectures concerning GC-sets. *Numer. Algorithms*, 45:113–125, 2007.

[Boo09] C. de Boor. Multivariate polynomial interpolation: Aitken-Neville sets and generalized principal lattices. *J. Approx. Theory*, 161(1):411–420, 2009.

[BR90] C. de Boor and A. Ron. On multivariate polynomial interpolation. *Constr. Approx.*, 6:287–302, 1990.

[BSCB00] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[Buh00] M.D. Buhmann. Radial basis functions. *Acta Numerica*, 9:1–38, 2000.

[Buh03] M.D. Buhmann. *Radial basis functions: theory and implementations.* Cambridge university press, 2003.

[Bus90] J. Busch. A note on Lagrange interpolation in $\mathbb{R}^2$. *Rev. Union Matem. Argent.*, 36:33–38, 1990.

[CDF92] A. Cohen, I. Daubechies, and J.C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Commun. Pur. Appl. Math.*, 45(5):485–560, 1992.

[CG10] J.M. Carnicer and M. Gasca. Multivariate polynomial interpolation: some new trends. *Monografias de la Real Academia de Ciencias de Zaragoza*, 32:197–208, 2010.

[CGS06] J.M. Carnicer, M. Gasca, and T. Sauer. Interpolation lattices in several variables. *Numer. Math.*, 102:559–581, 2006.

[CGS09] J. Carnicer, M. Gasca, and T. Sauer. Aitken–Neville sets, principal lattices and divided differences. *J. Approx. Theory*, 156(2):154–172, 2009.

[CGV99] A. Cohen, K. Gröchenig, and L.F. Villemoes. Regularity of multivariate refinable functions. *Constr. Approx.*, 15(2):241–255, 1999.

[CK05] D. Castaño and A. Kunoth. Multilevel regularization of wavelet based fitting of scattered data – some experiments. *Numer. Algorithms*, 39(1):81–96, 2005.

[Coa66] C. Coatmélec. Approximation et interpolation des fonctions différentiables de plusieurs variables. *Ann. Sci. École Norm. Sup.*, 83:271–341, 1966.

[CR72] P. Ciarlet and P. Raviart. General Lagrange and Hermite interpolation in $\mathbb{R}^n$ with applications to finite element methods. *Arch. Rational Mech. Anal.*, 46(3):177–199, 1972.

[Cul79] J. Cullum. The effective choice of the smoothing norm in regularization. *Math. Comput.*, 33:149–170, 1979.

[CY77] K.C. Chung and T.H. Yao. On lattices admitting unique Lagrange interpolations. *SIAM J. Numer. Anal.*, 14(4):735–743, 1977.

[Dau92] I. Daubechies. *Ten lectures on wavelets*, volume 61. SIAM, 1992.

[dBC57] C.G. den Broeder and A. Charnes. Contributions to the theory of generalized inverses for matrices. *Dept. of math., Purdue University, Lafayette*, 1957. Reprinted as ONR Res. Memo. No. 39, Northwestern University, Evanston, IL, 1962.

[DD87] G. Deslauriers and S. Dubuc. *Interpolation dyadique,* In: Fractals: Dimensions non entières et applications. Paris: Masson, 1987.

[DD89] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. *Constr. Approx.*, 5:49–68, 1989.

[DS12] T. Damm and D. Stahl. Linear least squares problems with additional constraints and an application to scattered data approximation. *Linear Algebra Appl.*, 2012. accepted, http://dx.doi.org/10.1016/j.laa.2012.08.015.

[Duc76] J. Duchon. Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *ESAIM, Math. Model. Numer. Anal.*, 10(R3):5–12, 1976.

[Duc77] J. Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*, volume 571, pages 85–100. Springer, 1977.

[FE98] C. Ford and D.M. Etter. Wavelet basis reconstruction of nonuniformly sampled data. *IEEE T. Circuits-II*, 45(8):1165–1168, 1998.

[Fra79] R. Franke. A critical comparison of some methods for interpolation of scattered data. *Naval Postgraduate School Techn. Rep.*, NPS-53-79-003, March 1979.

[GGM84] P. Goupillaud, A. Grossmann, and J. Morlet. Cycle-octave and related transforms in seismic signal analysis. *Geoexploration*, 23(1):85–102, 1984.

[GM82] M. Gasca and J.I. Maeztu. On Lagrange and Hermite interpolation in $\mathbb{R}^k$. *Numer. Math.*, 39:1–14, 1982.

[GM92] K. Gröchenig and W.R. Madych. Multiresolution analysis. Haar bases, and self-similar tilings of $\mathbb{R}^n$. *IEEE T. Inform. Theory*, 38(2):556–568, 1992.

[GR70] R. Guenter and E. Roetman. Some observations on interpolation in higher dimensions. *Math. Comp.*, 24(111):517–521, 1970.

[GS00a] M. Gasca and T. Sauer. On the history of multivariate polynomial interpolation. *J. Comput. Appl. Math.*, 122(1-2):23–35, 2000.

[GS00b] M. Gasca and T. Sauer. Polynomial interpolation in several variables. *Adv. Comput. Math.*, 4(12):377–410, 2000.

[GVL96] G.H. Golub and C.F. Van Loan. *Matrix computations.* Johns Hopkins University Press, 3rd edition, 1996.

[GVM97] G.H. Golub and U. Von Matt. Generalized cross-validation for large-scale problems. *J. Comput. Graph. Stat.*, pages 1–34, 1997.

[Han03] B. Han. Computing the smoothness exponent of a symmetric multivariate refinable function. *SIAM J. Matrix Anal. A.*, 24(3):693–714, 2003.

[Han10] P.C. Hansen. *Discrete Inverse Problems: Insight and Algorithms.* SIAM, 2010.

[Hut90] M.F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Commun. Stat. Simulat.*, 19(2):433–450, 1990.

[Jen06] T. Jensen. *Stabilization Algorithms for Large-Scale Problems.* PhD thesis, Technical University of Denmark, 2006.

[Jia98] R.Q. Jia. Approximation properties of multivariate wavelets. *Math. Comput.*, 67(222):647–666, 1998.

[Jia99] R.Q. Jia. Characterization of smoothness of multivariate refinable functions in Sobolev spaces. *T. Am. Math. Soc.*, 351:4089–4112, 1999.

[JS94] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analyses. *SIAM Rev.*, pages 377–412, 1994.

[JSX09] M.J. Johnson, Z. Shen, and Y. Xu. Scattered data reconstruction by regularization in B-spline and associated wavelet spaces. *J. Approx. Theory*, 159(2):197–223, 2009.

[JZ99] R.Q. Jia and S. Zhang. Spectral properties of the transition operator associated to a multivariate refinement equation. *Linear Algebra Appl.*, 292(1-3):155–178, 1999.

[KS00] J. Kovačević and W. Sweldens. Wavelet families of increasing order in arbitrary dimensions. *IEEE Trans. Image Process.*, 9(3):480–496, 2000.

[KV99] A. Karoui and R. Vaillancourt. Nonseparable biorthogonal wavelet bases of $L^2(\mathbb{R}^n)$. *Spline functions and the theory of wavelets*, 18:135–152, 1999.

[Lim90] J.S. Lim. *Two-dimensional signal and image processing.* Prentice-Hall, 1990.

[LLS97] W. Lawton, S.L. Lee, and Z. Shen. Stability and orthonormality of multivariate refinable functions. *SIAM J. Math. Anal.*, 28(4):999–1014, 1997.

[LLS98] W. Lawton, S.L. Lee, and Z. Shen. Convergence of multidimensional cascade algorithm. *Numer. Math.*, 78:427–438, 1998.

[LMR94] A.K. Louis, P. Maaß, and A. Rieder. *Wavelets: Theorie und Anwendungen.* Teubner, 1994.

[Mal89] S.G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE T. Pattern Anal.*, 11(7):674–693, 1989.

[MAT10] MATLAB. *Version 7.11.0.584 (R2010b) 32-bit.* The MathWorks Inc., 2010.

[MB35] E.H. Moore and R.W. Barnard. *General analysis.* The American philosophical society, 1935.

[Mei79] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *Z. Angew. Math. Phys.*, 30:292–304, 1979.

[Mey90] Y. Meyer. *Ondelettes et opérateurs. I - Ondelettes.* Hermann, Paris, 1990.

[Min85] F. Mintzer. Filters for distortion-free two-band multirate filter banks. *IEEE T. Acoust. Speech*, 33(3):626–630, 1985.

[Moo20] E.H. Moore. On the reciprocal of the general algebraic matrix. *Bull. Amer. Math. Soc.*, 26:394–395, 1920.

[NM65] J.A. Nelder and R. Mead. A simplex method for function minimization. *Comput. J.*, 7(4):308–313, 1965.

[NM99] N. Nguyen and P. Milanfar. A wavelet-based interpolation-restoration method for superresolution. *Circuits Systems Signal Process.*, 19(4):321–338, 1999.

[NV89] T.Q. Nguyen and P.P. Vaidyanathan. Two-channel perfect-reconstruction FIR QMF structures which yield linear-phase analysis and synthesis filters. *IEEE T. Acoust. Speech*, 37(5):676–690, 1989.

[Pen55] R. Penrose. A generalized inverse for matrices. *Proc. Camb. Phil. Soc.*, 51:406–413, 1955.

[Pen56] R. Penrose. On best approximate solutions of linear matrix equations. *Proc. Camb. Phil. Soc.*, 52:17–19, 1956.

[Rad48] J. Radon. Zur mechanischen Kubatur. *Monatshefte der Math. Physik*, 52(4):286–300, 1948.

[Rad56] R. Rado. Note on generalized inverses of matrices. *Proc. Camb. Phil. Soc.*, 52:600–601, 1956.

[Sau06] T. Sauer. *Polynomial Interpolation in Several Variables: Lattices, Differences, and Ideals*, volume 12 of *Studies in Computational Mathematics*. Elsevier, 2006.

[SB86] M. Smith and T. Barnwell. Exact reconstruction techniques for tree-structured subband coders. *IEEE T. Acoust. Speech*, 34(3):434–441, 1986.

[SB11] D. Stahl and C. de Boor. On Radon's recipe for choosing correct sites for multivariate polynomial interpolation. *J. Approx. Theory*, 163(12):1854–1858, 2011.

[Sch12] C.B. Schönlieb. Higher-order total variation inpainting. MATLAB Central File Exchange. Retrieved Aug 07, 2012.
http://www.mathworks.com/matlabcentral/fileexchange/34356.

[SN96] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.

[Sto09] A. Stoffel. Wavelets und Filterbänke, 2009.
http://alex.nt.fh-koeln.de/wavemat/wavelet.pdf.

[Swe96] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. A.*, 3(2):186–200, 1996.

[SX94] T. Sauer and Y. Xu. On multivariate Lagrange interpolation. *Math. Comp.*, 64:1147–1170, 1994.

[VA91] E. Viscito and J.P. Allebach. The analysis and design of multidimensional FIR perfect reconstruction filter banks for arbitrary sampling lattices. *IEEE T. Circuits Syst.*, 38(1):29–41, 1991.

[Vai93] P.P. Vaidyanathan. *Multirate systems and filter banks*. Prentice-Hall, 1993.

[VBU05] D. Van De Ville, T. Blu, and M. Unser. On the multidimensional extension of the quincunx subsampling matrix. *IEEE Signal Proc. Let.*, 12(2):112–115, 2005.

[VK95] M. Vetterli and J. Kovačević. *Wavelets and subband coding*. Prentice-Hall, 1995.

[VLG89] M. Vetterli and D. Le Gall. Perfect reconstruction FIR filter banks: Some properties and factorizations. *IEEE T. Acoust. Speech*, 37(7):1057–1071, 1989.

[Wah90] G. Wahba. *Spline models for observational data*. SIAM, 1990.

[Wen05] H. Wendland. *Scattered data approximation*. Cambridge monographs on applied and computational mathematics. Cambridge University Press, 2005.

Scientific Career

09/2012 – 02/2013 Visiting professor at FH Kaiserslautern

10/2009 – 04/2013 PhD studies in mathematics at TU Kaiserslautern

08/2007 – 02/2008 Semester abroad at TU Eindhoven

10/2004 – 06/2009 Studies in mathematics with specialization *Systems and Control* and minor subject electrical engineering at TU Kaiserslautern (Title of Diplom-thesis: *Particle filter - Model predictive control*)

06/2003 Abitur at Fachgymnasium Technik; BBS II, Braunschweig

Publications

2012 with Tobias Damm – Linear least squares problems with additional constraints, *Linear Algebra and its Applications*, accepted, http://dx.doi.org/10.1016/j.laa.2012.08.015, 11 pages

2012 with Tobias Damm – Approximation of scattered data using the lifting scheme, *PAMM* 12 (1) (2012) 739–740

2011 with Carl de Boor – On Radon's recipe for choosing correct sites for multivariate polynomial interpolation, *Journal of Approximation Theory* 163 (12) (2011) 1854–1858

2011 with Jan Hauth – Particle filter-model predictive control, *Systems & Control Letters* 60 (8) (2011) 632–643

Talks

2012 Parameterization of all $(1, 2, 3)$-generalized inverses with an application to scattered data approximation, *83rd Annual Scientific Conference of the International Association of Applied Mathematics and Mechanics*, Darmstadt

2011 Superresolution using the lifting scheme and an adapted pseudoinverse, *17th Conference of the International Linear Algebra Society*, Braunschweig

Wissenschaftlicher Werdegang

09/2012 – 02/2013 Vertretungsprofessur an der FH Kaiserslautern

10/2009 – 04/2013 Promotion an der TU Kaiserslautern Fachbereich Mathematik

08/2007 – 02/2008 Auslandsaufenthalt an der TU Eindhoven

10/2004 – 06/2009 Studium der Mathematik mit Vertiefungsrichtung *System- und Kontrolltheorie* und Nebenfach Elektrotechnik an der TU Kaiserslautern (Titel der Diplomarbeit: *Particle filter - Model predictive control*)

06/2003 Abitur am Fachgymnasium Technik; BBS II, Braunschweig

Veröffentlichungen

2012 mit Tobias Damm – Linear least squares problems with additional constraints, *Linear Algebra and its Applications*, akzeptiert, http://dx.doi.org/10.1016/j.laa.2012.08.015, 11 Seiten

2012 mit Tobias Damm – Approximation of scattered data using the lifting scheme, *PAMM* 12 (1) (2012) 739–740

2011 mit Carl de Boor – On Radon's recipe for choosing correct sites for multivariate polynomial interpolation, *Journal of Approximation Theory* 163 (12) (2011) 1854–1858

2011 mit Jan Hauth – Particle filter-model predictive control, *Systems & Control Letters* 60 (8) (2011) 632–643

Vorträge

2012 Parameterization of all $(1,2,3)$-generalized inverses with an application to scattered data approximation, *83rd Annual Scientific Conference of the International Association of Applied Mathematics and Mechanics*, Darmstadt

2011 Superresolution using the lifting scheme and an adapted pseudoinverse, *17th Conference of the International Linear Algebra Society*, Braunschweig