

Matrix Factorization and Lifting

Palle E. T. Jorgensen

Department of Mathematics, The University of Iowa,
Iowa City, IA52242, USA
jorgen@math.uiowa.edu

Myung-Sin Song

Department of Mathematics and Statistics, Southern Illinois University Edwardsville
Edwardsville, IL62026, USA
msong@siue.edu

Abstract

As a result of recent interdisciplinary work in signal processing (audio, still-images, etc), a number of powerful matrix operations have led to advances both in engineering applications and in mathematics. Much of it is motivated by ideas from wavelet algorithms. The applications are convincing measured against other processing tools already available, for example better compression (details below). We develop a versatile theory of factorization for matrix functions. By a matrix valued function we mean a function of one or more complex variables taking values in the group GL_N of invertible $N \times N$ matrices. Starting with this generality, there is a variety of special cases, also of interest, for example, one variable, or restriction to the case $n = 2$; or consideration of subgroups of GL_N or SL_N , i.e., specializing to the case of determinant equal to one. We will prove a number of factorization theorems and sketch their applications to signal (image processing) in the framework of multiple frequency bands.

Key words and phrases : Signals, image processing, algorithms, lifting, matrix factorization, Hilbert space, numerical methods, Fourier analysis.

2000 AMS Mathematics Subject Classification — Primary 18A32, 42C40, 46M05, 47B10, 60H05, 62M15, 65T60.

1 Introduction

Starting with the early work on wavelets (the 1980's), there is now an important body of theory at the crossroads of a number of mathematical areas (harmonic analysis and function theory) on the one side and theoretical signal processing on the other. An especially convincing instance of this was the recent use of wavelet algorithms by the JPEG group. (The term “JPEG” is an acronym for the Joint Photographic Experts Group which created the standard.) The

achieved compression resulting from these techniques is used, for example in a variety of image file formats. JPEG/Exif is the most common of them used in digital cameras and other image devices. Moreover, these mathematical tools are now part of common formats for storing and transmitting photographic images on the web.

The marriage of the two subjects came from the early realization that filters generating the most successful wavelet bases could be obtained from an adaptation of more classical sub-band filtering operations used in signal-processing; with the notions of down-sampling and up-sampling being intrinsic to numerical wavelet algorithms used in for example compression of signals and more generally of images. For a lucid account, see e.g., [16].

A common feature for the more traditional processing tools is the division into sub-bands, but in modern applications such a sub-division is more subtle. Here we develop and refine a procedure which uses factorization of families of matrix-valued function. These operations are done on the frequency-side; but it is fairly easy, at the end, to convert back to the time-signal itself. Here we use the concept of “time-signal” widely allowing for systems of numbers indexed by pixels, such as grey-scale numbers for still-images; or, for color, more complicated configuration of pixel matrices.

To be successful, a signal processing algorithms must allow a practical procedure for breaking down an overall process into small processes. The role of factorization of matrix functions is precisely to accomplish this: In the case of two frequency bands, 2×2 matrix functions suffice and in this case, the corresponding factorization (see e.g., Sweldens et al, [25, 26] and [10]). In this case, the factorization goes by the name “lifting” and the product is a (perhaps) long string of upper and lower triangular matrices, alternation between upper and lower. But each of these basic factors will then just encode a function of the frequency variable, corresponding to a filtering step in the overall process.

Below we demonstrate how this is done in the case of a process involving multiple bands, as well as the features dictated by modern applications.

2 Matrix Valued Functions

Matrix valued functions of one or more complex variables, taking values in the group SL_2 , have a number of uses in both pure and applied mathematics. Here we will focus on a framework in the signal processing literature called “the lifting scheme,” or “lifting algorithms.” A main result there (suitable restrictions) is the assertion that, in the case of polynomial entries, these matrix functions factor into finite products of alternating upper and lower diagonal matrix functions.

Even though pioneering ideas are from engineering, we hope to show that they are of interest in pure mathematics as well, especially in operator theory.

The result is of special practical significance in building filters using of two

frequency bands with a recursive input-output model; using as input filtered signals from the low band and producing an additive perturbation to the high frequency channel. This is continued recursively, with reversal of the role of high and low in each step. For some of the literature, we refer to [22, 10, 12] and many papers in the engineering literature. Since early pioneering ideas by Wim Sweldens, e.g., [24], the subject has since branched off in a variety of directions both applied and pure; see [11] and the papers cited there.

One of our motivations here is the desire to extend and refine this method to the case of multiple bands. In the simplest case, by this we mean that signals are viewed as time functions (discrete time) and each time-function generates a frequency response function (generating function) of a complex variable. In applications it is possible to encode time-signals or their generating functions as vectors in a Hilbert space \mathcal{H} . And to do this in such a way that a finite selection of frequency bands will then correspond to a system of closed subspaces in \mathcal{H} . A direct generalization of the case $n = 2$ to $n > 2$ is not feasible. We note that the factorization conclusion for $n = 2$ into alternating products of upper and lower, does not carry over to $n > 2$; but, motivated by applications, we outline a version that does.

A new element in our approach is the introduction of certain operator theoretic methods into the study of sub-band filtering in [4]; see also [5, 15, 23].

While the notion of upper/lower factorization is both versatile and old, dating back to Gauss, it has a variety of modern incarnations, many of which are motivated by computation. On the pure side, we list the Iwasawa decomposition for semisimple Lie groups [13] and on the applied side, the matrix formulation of the algorithm of Gram and Schmidt for creating orthogonal vectors in complex Hilbert space [2].

In signal processing, the context is different: Here we deal with infinite-dimensional groups of matrix functions; functions taking values in one of the finite-dimensional Lie groups, different groups for different purposes.

Of the many presentations in the literature dealing with signal processing applications, the following papers are especially relevant to our present approach: [19, 18, 17, 16, 9, 7, 8]. Equally important are the papers [11, 26, 25]; as well as their presentation in the book [14].

3 Systems of Filters

In this section we sketch the mathematics of filtering of signals (speech or images). We also endeavor to offer a dictionary translating between engineering terminology on one side and mathematical operations on the other.

While the method we outline applies more generally, for clarity we select our initial figures to illustrate the idea only in the simplest case. We then proceed to refinements and applications in sections 5 through 7.

Mathematically a discrete time signal is a numerical sequence, say (b_n) for input and (c_n) for output. The index n will typically represent time. Now the corresponding frequency response functions will be represented by a Fourier series, $g(z)$ and $h(z)$ respectively. These are functions defined on the circle group \mathbb{T} .

Operations on input data are represented as black boxes. Three operations enter into signal processing algorithms: (i) filter in the form of a weighted average, (ii) down-sampling and (iii) up-sampling. While the observed data is in the form of numerical sequences, matrix operations are more practically done on the corresponding frequency response functions.

For example, a weighted average applied to a time-series turns into a Cauchy product acting on the frequency response functions. Figure 1 below illustrates the various operators combined into a sub-band filtering with input and output. Specifically, in Fig. 1, we have a fixed number N of assigned frequency sub-bands, so N bands in all. For each band we further have prescribed functions, so f_0, f_1, \dots, f_{N-1} , with the subscript 0 indicating the lowest band. In computations, it will be handy to index the bands and the filters, by elements in the cyclic group of order N , $\{0, 1, 2, \dots, N-1\}$, denoted \mathbb{Z}_N . (If $N = 2$, the bands are called low-pass and high-pass.)

In each band, as illustrated in Fig. 1, there is a sequence of steps of operations as follows: analysis of input, filter, down-sampling, up-sampling, dual filter and synthesis into output. In the lowest frequency band the filter f_0 is used and so on. In the most restricted form of signal processing, the frequency bands are arranged into orthogonal subspaces, but the orthogonality requirement will be relaxed in our present analysis. Rationale: In practice there are better properties available!

In this section we present the mathematics of some key concepts from signal processing. In their mathematical form, these ideas are timeless and pretty versatile; thus applying equally well to signals of a more basic nature, as well as to signal processing in wireless communication. With suitable adaptation, these in fact are tools for image processing as well.

To introduce a key idea used below, consider the simplest case of a time-series, here corresponding to discrete time; in other words, signals represented by sequences of complex numbers. Hence time is represented by an arithmetic progression, here the integers \mathbb{Z} . As a result the dual frequency variable will be 2π -periodic. In the language of group duality, the circle group $\mathbb{T} := \text{one-torus}$, with Haar measure, is the dual of the additive group \mathbb{Z} . The introduction of group duality is helpful in several ways: it offers a compact representation of the particular Fourier analysis we need and more importantly it offers an economical framework for analysis of frequency bands. In our analysis below this entails representations of classical matrix groups G .

Let N be a fixed positive integer, 2 or larger. Then a sub-band system

corresponding to N frequency bands is a N -fold partition of the period interval, or equivalently of the group T and a system of N complex valued functions. The classical matrix groups G referred to above will now be groups of $N \times N$ complex matrices. Hence the groups G will act on N -vectors \mathbb{C}^N by matrix multiplication. We will present filter functions as functions defined on \mathbb{T} and taking values in \mathbb{C}^N . Further we will be using infinite-dimensional groups of functions U defined on \mathbb{T} and mapping into G , referred to as matrix functions. The action of a matrix function U on a vector function F will be pointwise multiplication, i.e.,

$$(UF)(z) := U(z)F(z), \quad z \in \mathbb{T}.$$

We will further need to introduce Hilbert spaces \mathbb{H} in such a manner that the groups acquire agreeable representations as they act on \mathbb{H} .

The purpose of our presentation here is to set up the problems for the framework of matrix analysis. By this we mean the study of functions in one or several complex variables, but taking values in a particular Lie group of invertible matrices, for example the general linear group GL_N , the group \mathcal{U}_N of unitary $N \times N$ matrices, or one of the symplectic groups, etc. The choice of group in our analysis depends on the problem at hand. While the Lie groups G in the above list are finite-dimensional, the moment we pass to the group of functions taking values in G , this will be an infinite-dimensional group.

Setting

- \mathbb{C} : the complex plane
- $D := \{z \in \mathbb{C}; |z| < 1\}$
- $\partial D := \mathbb{T} = \{z \in \mathbb{C}; |z| = 1\} = \{e^{i\theta}; \theta \in \mathbb{R}\}$, or $\mathbb{R}/2\pi\mathbb{Z}$
- Let $\Omega \subset \mathbb{C}$ be an open subset such that $\mathbb{T} \subset \Omega$. Algebras of functions and Fourier representation:

$$f(z) = \sum_k a_k z^k = \sum_k a_k e^{i2\pi k\theta}. \tag{1}$$

If $g(z) = \sum_k b_k z^k$, we shall impose conditions at $k \rightarrow \infty$ such that

$$f(z)g(z) = \sum_n c_n z^n, \quad \text{where}$$

$$c_n = \sum_k a_k b_{n-k} \tag{2}$$

can be justified.



3.1 Operations on Time-Signals

Filtering If $(b_m)_{m \in \mathbb{Z}}$ is a time-signal, we say that (2) is a filter acting on (b_m) .

Below we will be using the notations \uparrow and \downarrow to denote operators, i.e., transformations acting on spaces of signals. These two (upsampling and downsampling) are operations on sequences.

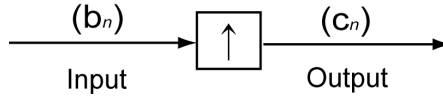
Upsampling \uparrow Fix $N \in \mathbb{Z}_+$, $N > 1$. Consider a time-signal (b_k) and a frequency response function $g(z) = \sum_k b_k z^k$. Here the function g is defined on \mathbb{T} and its coefficients are the sequences used in part 1 of the definition.

Action on the signal: $(b_k) \mapsto (c_n)$ where

$$c_n = \begin{cases} b_k & \text{if } N|n, \text{ i.e., } \exists k \text{ such that } n = N \cdot k, \\ 0 & \text{if } N \nmid n. \end{cases} \quad (3)$$

Action on the functions:

$$g(z) \mapsto h(z), \text{ where } h(z) = g(z^N). \quad (4)$$



Downsampling \downarrow Fix $N \in \mathbb{Z}_+$, $N > 1$. Consider a time-signal (b_k) and a frequency response function $g(z) = \sum_k b_k z^k$. Here the function g is defined on \mathbb{T} and its coefficients are the sequences used in part 1 of the definition.

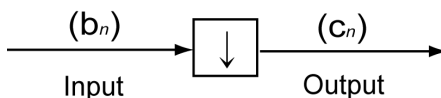
Action on Signal: $(b_k) \mapsto (c_n)$ where

$$c_n = b_{nN} \quad \text{for all } n \in \mathbb{Z} \quad (5)$$

(i.e., discard input b_k when k is not divisible by N .)

Action on functions: $g(z) \mapsto h(z)$, where g is defined on \mathbb{T} and taking values in \mathbb{C} or in \mathbb{C}^N and

$$h(z) = \frac{1}{N} \sum_{w \in \mathbb{T}, w^N = z} g(w), \quad \text{average over } N^{\text{th}} \text{ roots.} \quad (6)$$



Frequency bands: We say that a partition of $-\pi \leq \theta < \pi$ into N sub-intervals.

$$\frac{2\pi k}{N} - \frac{\pi}{N} \leq \theta < \frac{2\pi k}{N} + \frac{\pi}{N}$$

is a sub-band partition with $k = 0$ corresponding to the lowest band and $k = \lceil \frac{N}{2} \rceil$ the highest band.

Definition 3.1. Let $N \in \mathbb{Z}_+$ be given and set $\zeta_N := e^{i\frac{2\pi}{N}}$ = the principal N^{th} root of 1. Set

$$(A_N g)(z) := \frac{1}{N} \sum_{k=0}^{N-1} g(\zeta_N^k z). \tag{7}$$

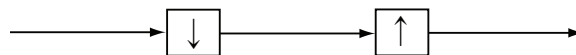
Here the two versions of the operator A_N represent transformations in sequence spaces. But by Fourier-duality, this turns into associated actions on spaces of functions, so functions defined on \mathbb{T} .

Note the summation in (7) is over the cyclic group $\mathbb{Z}_N = \mathbb{Z}/N\mathbb{Z} (= \{0, 1, \dots, N-1\})$.

Lemma 3.2.

$$A_N = \uparrow \downarrow \quad (\text{downsampling followed by upsampling}), \tag{8}$$

i.e., composition of operators.



Proof.

$$\begin{aligned} (A_N g)(z) &= (\downarrow g)(z^N) \\ &= \frac{1}{N} \sum_{\substack{w \in \mathbb{T}, \\ w^N = z^N}} g(w) \\ &= \frac{1}{N} \sum_{k \in \mathbb{Z}_N} g(\zeta_N^k z), \quad \text{summation over the cyclic group of order } N, \end{aligned}$$

which is the formula in (7). □

Corollary 3.3. *The action of A_N on the time-signal (b_k) is as follows*

$$(A_N b)_k = \begin{cases} b_k & \text{if } N|k, \\ 0 & \text{if } N \nmid k. \end{cases}$$

Definition 3.4. Let $N \in \mathbb{Z}_+$ be given. If F is a vector valued function defined on \mathbb{T} , by (f_k) we mean the corresponding coordinate functions. The same for G and (g_k) . If f is a scalar valued function, we denote the corresponding multiplication operator by M_f . Two systems of functions

$$F = (f_k)_{k \in \mathbb{Z}_N} \quad \text{and} \quad G = (g_k)_{k \in \mathbb{Z}_N}$$

are said to be a perfect reconstruction filter iff

$$\sum_{k \in \mathbb{Z}_N} M_{g_k} A_N M_{f_k} = I \quad (\text{see Fig. 1}) \quad (9)$$

where the operator I on the RHS in (9) is the identity operator.

In the engineering lingo, e.g. (9) is expressed as follows:

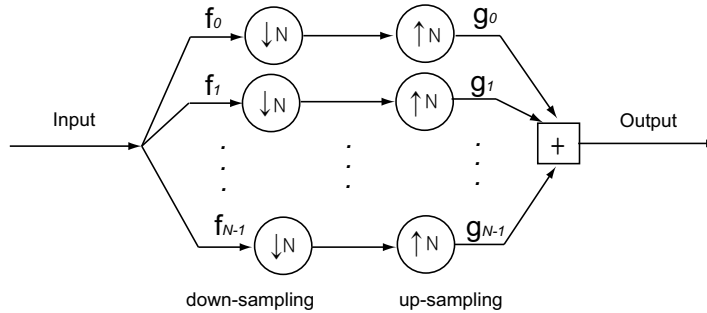


Figure 1: Perfect reconstruction in subband filtering as used in signal and image processing. Input is broken down into frequency bands, processes and then assembled (synthesis). Perfect reconstruction of output is desired.

Perfect reconstruction in subband filtering as used in signal and image processing. Input is broken down into frequency bands, processes, and then assembled (synthesis). Perfect reconstruction of output is desired.

Definition 3.5. For function $f(z) = \sum_{n \in \mathbb{Z}} a_n z^n$, set

$$f^*(z) = \sum_{n \in \mathbb{Z}} \overline{a_{-n}} z^n. \quad (10)$$

4 Groups of Matrix Functions

Groups of functions taking values in a particular Lie group G (see section 2 for details) act naturally on vector valued functions. This action is simply pointwise: If G is a group of $N \times N$ complex matrices, the action will then be on functions mapping into \mathbb{C}^N , i.e., complex N -space. This is important as the mathematics of filters in signal processing takes place on \mathbb{C}^N -valued functions. The way this is done is outlined below; keeping in mind our framework of factorization for a particular (infinite-dimensional) group of functions taking values in some Lie group G .

Definition 4.1. Let F be a \mathbb{C}^N -valued function defined on \mathbb{T} and we denote its coordinate functions (f_k) with the index k running over cyclic group \mathbb{Z}_N or order N . A system $F = (f_k)_{k \in \mathbb{Z}_N}$ is said to be an orthogonal filter (with N bands) iff (9) holds with $g_k = f_k^*$.

Proposition 4.2. A system $F = (f_k)_{k \in \mathbb{Z}_N}$ is an orthogonal filter with N bands iff the $N \times N$ matrix

$$U_F(z) := (f_j(\zeta_N^k z))_{(j,k) \in \mathbb{Z}_N \times \mathbb{Z}_N} \quad (11)$$

is unitary for all $z \in \mathbb{T} (= \partial D)$.

Proof. An application of the previous lemma. \square

We will consider functions on $\mathbb{T} (\subset \mathbb{C})$ taking values in (i) the scalars \mathbb{C} , (ii) in a fixed vector space, for example \mathbb{C}^N for some N ; or (iii) in some group of $N \times N$ complex matrices: In the first case we write

- (i) $\mathbb{T} \ni z \mapsto f(z) \in \mathbb{C}$,
- (ii) $\mathbb{T} \ni z \mapsto F(z) = (f_k(z))_{k=1}^N \in \mathbb{C}^N$,
- (iii) $\mathbb{T} \ni z \mapsto A(z) = (A_{j,k}(z))_{j,k=1}^N$

denotes a matrix function.

If A and B are matrix function with scalar functions as entries $(A_{j,k}(z))$ and $(B_{j,k}(z))$ respectively, set $C = AB$ where $C(z) = (C_{j,k}(z))$ is the usual pointwise matrix-product

$$C_{j,k}(z) = \sum_{l \in \mathbb{Z}_N} A_{j,l}(z) B_{l,k}(z).$$

Definition 4.3. An $N \times N$ matrix-valued function U is said to be unitary iff $U(z)$ is a unitary matrix for all $z \in \mathbb{T}$.

Let the set of all orthogonal N -filters be denoted \mathcal{OF}_N and the set, all unitary matrix functions by \mathcal{UM}_N .

Definition 4.4. Let U be an $N \times N$ matrix-function and let $F = (f_k)_{k \in \mathbb{Z}_N}$ be a function system. Set

$$G(z) := U(z^N)F(z), \quad (12)$$

or equivalently

$$g_k(z) = \sum_{j \in \mathbb{Z}_N} U_{k,j}(z^N) f_j(z). \quad (13)$$

Lemma 4.5. *With the action (12), the group \mathcal{UM}_F acts transitively on \mathcal{OF}_N .*

Proof. If $F \in \mathcal{OF}_N$ and $U \in \mathcal{UM}_F$, then the action (12) is easily seen to make $G(z) = U(z^N)F(z)$ an orthogonal filter.

Let F and G be in \mathcal{OF}_N and set

$$U_{j,k}(z) = \frac{1}{N} \sum_{\substack{w \in \mathbb{T}, \\ w^N = z}} g_k(w) \overline{f_j(w)}. \quad (14)$$

An inspection shows that $U = (U_{j,k})$ is in \mathcal{UM}_F and that (13) is satisfied. \square

Corollary 4.6. *Let $N \in \mathbb{Z}_+$ be given. Set*

$$b(z) = \begin{bmatrix} 1 \\ z \\ z^2 \\ \vdots \\ z^{N-1} \end{bmatrix}; \quad \text{then} \quad (15)$$

$$\begin{aligned} \mathcal{OF}_N &= \mathcal{UM}_F b \\ &=: \{U(z^N)b(z); U \in \mathcal{UM}_F\}. \end{aligned}$$

In our analysis below, we will be using more than one inner product. In a number of places a new inner product will be defined from a given one, entailing an average over the cyclic group of order N or a system of equi-partitioned points on the frequency circle. In these cases, the new inner product will be denoted $\ll \cdot, \cdot \gg_N$.

Definition 4.7. Let $N \in \mathbb{Z}_+$ be given and let $\langle \cdot, \cdot \rangle_N$ be the usual inner product in \mathbb{C}^N , i.e.,

$$\langle v, w \rangle_N := \sum_{k=1}^N \overline{v_k} w_k. \quad (16)$$

If F and G are \mathbb{C}^N -valued matrix, functions, set

$$\ll F, G \gg_N(z) = \frac{1}{N} \sum_{\substack{w \in \mathbb{T}, \\ w^N = z}} \langle F(w), G(w) \rangle_N = \downarrow \langle F, G \rangle_N(z). \quad (17)$$

Lemma 4.8. *Let $N \in \mathbb{Z}_+$ be fixed; and let A and B be matrix functions. Then*

$$\ll Ab, Bb \gg_N = \text{trace}(A^*(z)B(z)), \quad (18)$$

where b is given by (15).

Proof.

$$\begin{aligned} \ll Ab, Bb \gg_N &= \frac{1}{N} \sum_{w^N=z} \sum_j \sum_k \sum_l \overline{A_{j,k}(z)w^k} B_{j,l}(z)w^l \\ &= \sum_j \sum_k \sum_l \overline{A_{j,k}(z)} B_{j,l}(z) \frac{1}{N} \sum_{w^N=z} \overline{w^k} w^l \\ &= \sum_j \sum_k \sum_l \overline{A_{j,k}(z)} B_{j,l}(z) \delta_{k,l} \\ &= \sum_j \sum_k \overline{A_{j,k}(z)} B_{j,l}(z) \\ &= \text{trace}(A(z)^* B(z)), \end{aligned}$$

which is the desired conclusion. \square

Definition 4.9. Let $\mathcal{H} = L^2(\mathbb{T})$ be the Hilbert space of functions φ given by

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |\varphi(e^{i\theta})|^2 d\theta = \sum_{n \in \mathbb{Z}} |b_n|^2, \quad (19)$$

where $\varphi(e^{i\theta}) = \sum_{n \in \mathbb{Z}} b_n e^{in\theta}$ is the Fourier representation of φ .

Let $F = (f_j)_{j \in \mathbb{Z}_N}$ be a function system on set

$$(S_j \varphi)(z) = f_j(z) \varphi(z^N). \quad (20)$$

Lemma 4.10. *Let $N \in \mathbb{Z}_+$ be given and let $F = (f_j)_{j \in \mathbb{Z}_+}$ be a function system. Then $F \in \mathcal{OF}_N$ if and only if the operators S_j in (20) satisfy*

$$S_j^* S_k = \delta_{j,k} I$$

$$\sum_{j \in \mathbb{Z}_N} S_j S_j^* = I,$$

where I denotes the identity operator in $\mathcal{H} = L^2(\mathbb{T})$; compare with Fig. 1.

Proof. This is a direct application of the two previous lemmas. \square

5 Group Actions

Let N be a positive integer, $N \geq 2$. A subband filter with N bands (as used in signal and image processing) is a system of functions $F := (f_0, f_1, \dots, f_{N-1})$ defined on a frequency band, say $-\pi \leq \theta < \pi$. We will take this in the form $e^{i\theta} \in \mathbb{T}$ and then view F as a function on \mathbb{T} taking values in \mathbb{C}^N . With Haar measure on \mathbb{T} we therefore consider the Hilbert space $L^2(\mathbb{T}, \mathbb{C}^N)$ below.

In this section we outline how the entire processing system in Fig. 1 may be encoded into a representation of a certain C^* -algebra, an algebra on N generators and two relations, called Cuntz-relations, or generalized Cuntz-relations. We state our first results regarding factorization in (infinite-dimensional) groups of functions taking values in some Lie group G ; matrix-functions for short.

We outline notational conventions and state the factorization problem in a simple case. Generalities will be added later. We begin with two key lemmas to be applied later.

Let $N \in \mathbb{Z}_+$ be given ($N > 1$) and consider $F = (f_j)_{j \in \mathbb{Z}_+}$ in $\mathcal{F}_2(N) := L^2(\mathbb{T}, \mathbb{C}^N) = \sum_0^{N-1} \oplus L^2(\mathbb{T})$ where the notation in the summation symbol means orthogonal direct sum with

$$\|F\|_2^2 = \sum_{j=0}^{N-1} \|f_j\|_{L^2(\mathbb{T})}^2 < \infty.$$

We will be making use of the special vector $b \in \mathcal{F}_2(N)$,

$$b(z) = \begin{bmatrix} 1 \\ z \\ z^2 \\ \vdots \\ z^{N-1} \end{bmatrix};$$

see Corollary 4.6.

Let

$$(S_j f)(z) = z^j f(z^N) \tag{21}$$

be the Cuntz-representation from Definition 4.9 and Lemma 4.10.

Lemma 5.1. *Let $N \in \mathbb{Z}_+$ be fixed, $N > 1$ and let $A = (A_{j,k})$ be an $N \times N$ matrix-function with $A_{j,k} \in L^2(\mathbb{T})$. Then the following two conditions are equivalent:*

(i) *For $F = (f_j) \in \mathcal{F}_2(N)$, we have $F(z) = A(z^N)b(z)$.*

(ii) *$A_{i,j} = S_j^* f_i$ where the operators S_i are from the Cuntz-relations (21).*

Proof. (i) \Rightarrow (ii). Writing out the matrix-operation in (i), we get

$$f_i(z) = \sum_j A_{i,j}(z^N)z^j = \sum_j (S_j A_{i,j})(z). \quad (22)$$

Using $S_j^* S_k = \delta_{j,k} I$, we get $A_{i,j} = S_j^* f_i$ which is (ii).

Conversely, assuming (ii) and using $\sum_j S_j S_j^* = I$, we get $\sum_j S_j A_{i,j} = f_i$ which is equivalent to (i) by the computation in (22) above. \square

Corollary 5.2. *Let $N \in \mathbb{Z}_+$ be fixed and let A and B be $N \times N$ matrix-functions with L^2 -entries. Then the following are equivalent:*

- (i) $A(z^N)b(z) = B(z^N)b(z)$ and
- (ii) $A \equiv B$.

5.1 Factorizations

We will now sketch the first step in the general conclusions about factorization.

In the arguments below, the size of the problem has two parts:

- (a) The matrix size, i.e., the size of N where we consider $N \times N$ matrices.
- (b) The number of factors in our factorizations.

To illustrate the idea, we begin with consideration of the case when $N = 2$ and the number of factors is also two.

Lemma 5.3. *Let*

$$A = \begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{pmatrix}$$

be a 2×2 matrix-function and let

$$\begin{cases} f_0(z) = \mathcal{A}(z^2) + z\mathcal{B}(z^2) \\ f_1(z) = \mathcal{C}(z^2) + z\mathcal{D}(z^2). \end{cases}$$

Let L and U be scalar functions. Then the following are equivalent:

(i)

$$\begin{pmatrix} 1 & 0 \\ L & 1 \end{pmatrix} \begin{pmatrix} 1 & U \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{pmatrix}.$$

(ii) $U = S_1^* f_0$ and $L = S_0^* f_1$.

Proof. This is a direct consequence of the lemmas in section 4. \square

5.2 Notational Conventions

- (i) Let $N \in \mathbb{Z}_+$ be fixed. We will denote $N \times N$ matrix function $A(z) = (A_{j,k}(z))_{j,k \in \mathbb{Z}_N}$ and N -column vector functions by

$$v(z) = \begin{bmatrix} v_0(z) \\ v_1(z) \\ \vdots \\ v_{N-1}(z) \end{bmatrix}.$$

We will consider A acting on the vector v as follows:

$$A_N[v](z) := A(z^N)v(z), \quad (23)$$

where the RHS in (23) is a $(N \times N)(N \times 1)$ matrix-product. Note the subscript N in the definition (23) above.

- (ii) If f and g are two scalar valued functions, we set

$$\langle f, g \rangle_N(z) = \frac{1}{N} \sum_{w \in \mathbb{T}} \sum_{w^N=z} \overline{f(w)}g(w), \quad (24)$$

i.e., this is an inner product taken values in spaces of functions.

- (iii) If f is given, we set

$$(S_f \varphi)(z) := f(z)\varphi(z^N) \quad (25)$$

and

$$(S_f^* \varphi)(z) := \frac{1}{N} \sum_{w \in \mathbb{T}} \sum_{w^N=z} \overline{f(w)}\varphi(w) = \langle f, \varphi \rangle_N(z) \quad (26)$$

- (iv) Note that S_f^* is the $L^2(\mathbb{T})$ -adjoint operator of S_f , i.e., if $\varphi, \psi \in L^2(\mathbb{T})$, then

$$\langle S_f \varphi, \psi \rangle_{L^2(\mathbb{T})} = \langle \varphi, S_f^* \psi \rangle_{L^2(\mathbb{T})} \quad (27)$$

where $\langle \cdot, \cdot \rangle_{L^2(\mathbb{T})}$ denotes the usual inner product in the Hilbert space $L^2(\mathbb{T})$.

Lemma 5.4. *Let f_0, f_1, \dots, f_{N-1} be a system of N complex functions. (For the present purpose, we only need to assume that each f_j is in $L^\infty(\mathbb{T})$.)*

Then the following three conditions are equivalent:

- (i) *The functions f_j satisfy*

$$\langle f_j, f_k \rangle_N(z) = \delta_{j,k} \mathbf{1}, \quad \forall z \in \mathbb{T}, \quad \text{module-orthogonality.} \quad (28)$$

(ii) The operator S_{f_j} satisfy the Cuntz-relations

$$\begin{cases} S_{f_j}^* S_{f_k} = \delta_{j,k} I_{L^2(\mathbb{T})}, & \text{and} \\ \sum_{j=0}^{N-1} S_{f_j} S_{f_j}^* = I_{L^2(\mathbb{T})}. \end{cases} \quad (29)$$

(iii) With $\zeta_N := e^{i\frac{2\pi}{N}}$, form the matrix function

$$M_N(z) = (f_j(\zeta_N^k z))_{j,k \in \mathbb{Z}_N}. \quad (30)$$

Then M_N is a unitary matrix-function.

Lemma 5.5. *The proof follows from a direct verification; see also the book [4], chapter 2.*

Definition 5.6. A system of functions $(f_j)_{j \in \mathbb{Z}_N}$ satisfying any one of the three conditions in Lemma 5.4 is called an orthogonal system of sub-band filters.

Remark 5.7. An advantage of the operator formalism in Lemma 5.4 (representations of Cuntz algebras) is that it enables the operators $P_j := S_j S_j^*$ to be a system of mutually orthogonal projections. i.e., projections onto the subspaces in $L^2(\mathbb{T}) \sim l^2(\mathbb{Z})$ corresponding to frequency bands with $P_0 =$ projection onto the subspace of the lowest band.

This representation simplifies in the case of just two bands: then the family

$$Q_i := S_1^i S_0 S_0^* S_1^{*i}, \quad i = 0, 1, 2, \dots$$

is infinite and mutually orthogonal. We get a well-defined infinite sum (of orthogonal projections):

$$\sum_{i=0}^{\infty} Q_i = I_{L^2(\mathbb{T})} \sim I_{l^2}. \quad (31)$$

To justify (31), we use that $\lim_{n \rightarrow \infty} S_1^n S_1^{*n} = 0$ holds in the strong operator topology. With this, we then get a useful version of the pyramid algorithm, and even an image-subdivision scheme; see Fig. 3 and Fig. 6.

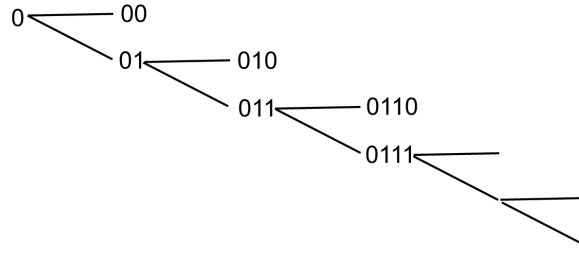


Figure 2: Pyramid Algorithm. Symbolic encoding of data with the use of finite words expressed in bits.

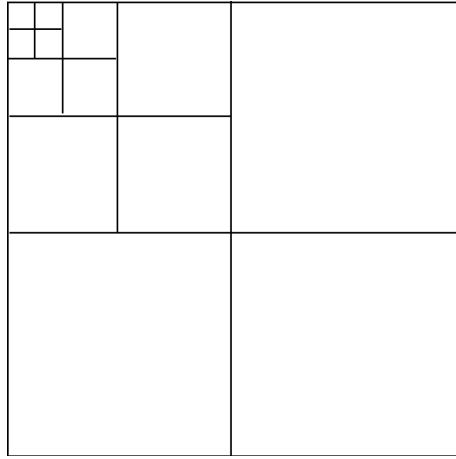


Figure 3: Wavelet decomposition on a image with $N = 4$.

Corollary 5.8. *Every orthogonal system of sub-band filters $F = [f_j]_{j \in \mathbb{Z}_N}$ has the form*

$$F = U_N[b], \quad (32)$$

where U is a unitary matrix-function, where

$$b = \begin{pmatrix} 1 \\ z \\ z^2 \\ \vdots \\ z^{N-1} \end{pmatrix}$$

and where $U_N[b](z) = U(z^N)b(z)$.

Definition 5.9. In the computations below we will outline a number of finite algorithms. They will involve operations on functions. The idea is to break down operations on large systems into a sequence of steps, each of the steps acting on a smaller subsystem. The subsystems will be represented by polynomial functions, so by finite Fourier expansions.

A matrix-function or a vector function is said to be of polynomial type, or a polynomial matrix-function, if its entries are polynomials: If $H \subset \mathbb{Z}$ is a finite subset of the integers and $a : H \mapsto \mathbb{C}$ is a function on H , by a polynomial we shall mean the expression

$$f_H(z) := \sum_{n \in H} a_n z^n \tag{33}$$

which is a finite Laurent expression.

The difference $D = \max H - \min H$ will be called the degree of f_H .

Let $N \in \mathbb{Z}_+$ be given and fixed. The following terminology will be used:

$GL_N(\text{pol})$: the $N \times N$ polynomial matrix function A such that A^{-1} is also polynomial.

$$SL_N(\text{pol}) := \{A \in GL_N(\text{pol}); \det A \equiv 1\}. \tag{34}$$

Our work on matrix functions gives the following:

Theorem 5.10. (Sweldens [24]) *Let $A \in SL_2(\text{pol})$, then there are $l, p \in \mathbb{Z}_+$, $K \in \mathbb{C} \setminus \{0\}$ and polynomial functions $U_1, \dots, U_p, L_1, \dots, L_p$ such that*

$$A(z) = z^l \begin{pmatrix} K & 0 \\ 0 & K^{-1} \end{pmatrix} \begin{pmatrix} 1 & U_1(z) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ L_1(z) & 1 \end{pmatrix} \cdots \begin{pmatrix} 1 & U_p(z) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ L_p(z) & 1 \end{pmatrix}. \tag{35}$$

Remark 5.11. Note that if

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \in SL_2(\text{pol}),$$

then one of the two functions $\alpha(z)$ or $\delta(z)$ must be a monomial.

6 Divisibility and Residues for Matrix-functions

The present section deals with some key steps in the proof of our two main theorems.

6.1 The 2×2 case

To highlight the general ideas, we begin with some details worked out in the 2×2 case; see equation (35).

First note that from the setting in Theorem 5.10, we may assume that matrix entries have the form $f_H(z)$ as in (33) but with $H \subset \{0, 1, 2, \dots\}$, i.e., $f_H(z) = a_0 + a_1z + \dots$. This facilitates our use of the Euclidean algorithm.

Specifically, if f and g are polynomials (i.e., $H \subset \{0, 1, 2, \dots\}$) and if $\deg(g) \leq \deg(f)$, the Euclidean algorithm yields

$$f(z) = g(z)q(z) + r(z) \quad (36)$$

with $\deg(r) < \deg(g)$. We shall write

$$q = \text{quot}(g, f), \quad \text{and} \quad r = \text{rem}(g, f). \quad (37)$$

Since

$$\begin{pmatrix} K & 0 \\ 0 & K^{-1} \end{pmatrix} \begin{pmatrix} 1 & U \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & K^2U \\ 0 & 1 \end{pmatrix} \begin{pmatrix} K & 0 \\ 0 & K^{-1} \end{pmatrix}, \quad (38)$$

we may assume that the factor

$$\begin{pmatrix} K & 0 \\ 0 & K^{-1} \end{pmatrix}$$

from the equation (35) factorization occurs on the rightmost place.

Let U represent scalar valued matrix entry in a matrix function. We now proceed to determine the polynomials $U_1(z), L_1(z), \dots$, etc. inductively starting with

$$A = \begin{pmatrix} 1 & U \\ 0 & 1 \end{pmatrix} B,$$

where U and B are to be determined. Introducing (32), this reads

$$A(z^2) \begin{pmatrix} 1 \\ z \end{pmatrix} = \begin{pmatrix} 1 & U(z^2) \\ 0 & 1 \end{pmatrix} B(z^2) \begin{pmatrix} 1 \\ z \end{pmatrix} = \begin{pmatrix} 1 & U(z^2) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} h(z) \\ k(z) \end{pmatrix}. \quad (39)$$

But the matrix function

$$A = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$

is given and fixed see Remark 5.11. Hence

$$\gamma(z^2) + \delta(z^2)z = k(z) \quad (40)$$

is also fixed. The two polynomials to be determined are u and h in (39). Carrying out the matrix product in (39) yields:

$$\alpha(z^2) + \beta(z^2)z = h(z) + u(z^2)k(z) = h_0(z) + h_1(z^2)z + u(z^2)\{\gamma(z^2) + \delta(z^2)z\}$$

where we used the orthogonal splitting

$$L^2(\mathbb{T}) = S_0 S_0^* L^2(\mathbb{T}) \oplus S_1 S_1^* L^2(\mathbb{T}) \quad (41)$$

from Lemma 4.10. Similarly, from (40), we get

$$\gamma(z^2) + \delta(z^2)z = k_0(z^2) + k_1(z^2)z;$$

and therefore $\gamma = k_0$ and $\delta = k_1$, by Lemma 5.1.

Collecting terms and using the orthogonal splitting (41) we arrive at the following system of polynomial equations:

$$\begin{cases} \alpha = h_0 + u\gamma \\ \beta = h_1 + u\delta; \end{cases} \quad (42)$$

or more precisely,

$$\begin{cases} \alpha(z) = h_0(z) + u(z)\gamma(z) \\ \beta(z) = h_1(z) + u(z)\delta(z). \end{cases}$$

It follows that the two functions u and h may be determined from the Euclidean algorithm. With (38), we get

$$\begin{cases} u = \text{quot}(\gamma, \alpha) \\ h_0 = \text{rem}(\gamma, \alpha) \\ h_1 = \text{rem}(\delta, \beta). \end{cases} \quad (43)$$

Remark 6.1. The relevance of the determinant condition we have from Theorem 5.10 is as follows:

$$\det A = \alpha\delta - \beta\gamma \equiv 1.$$

Substitution of (42) into this yields:

$$h_0\delta - h_1\gamma \equiv 1.$$

Solutions to (42) are possible because the two polynomials $\delta(z)$ and $\gamma(z)$ are mutually prime. The derived matrix

$$\begin{pmatrix} h_0 & h_1 \\ \gamma & \delta \end{pmatrix}$$

is obtained from A via a row-operation in the ring of polynomials.

For the inductive step, it is important to note:

$$\deg(h_0) < \deg(\gamma), \quad \text{and} \quad \deg(h_1) < \deg(\delta). \quad (44)$$

The next step, continuing from (39) is the determination of a matrix-function C and three polynomials p, q , and L such that

$$\begin{pmatrix} 1 & -U \\ 0 & 1 \end{pmatrix} A = \begin{pmatrix} 1 & 0 \\ L & 1 \end{pmatrix} C \quad (45)$$

and

$$\begin{pmatrix} 1 & -U(z^2) \\ 0 & 1 \end{pmatrix} A(z^2) \begin{pmatrix} 1 \\ z \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ L(z^2) & 1 \end{pmatrix} \begin{pmatrix} p(z) \\ q(z) \end{pmatrix}. \quad (46)$$

Here

$$\begin{pmatrix} p \\ q \end{pmatrix} = C(z^2) \begin{pmatrix} 1 \\ z \end{pmatrix}.$$

The reader will notice that in this step, everything is as before with the only difference that now

$$\begin{pmatrix} 1 & 0 \\ L & 1 \end{pmatrix}$$

is lower diagonal in contrast with

$$\begin{pmatrix} 1 & U \\ 0 & 1 \end{pmatrix}$$

in the previous step.

This time, the determination of the polynomial p in (46) is automatic. With

$$p(z) = p_0(z^2) + zp_1(z^2)$$

(see (41)) and we get the following system:

$$\begin{cases} p_0 = \alpha - u\gamma = h_0 \\ p_1 = \beta - u\delta = h_1; \quad \text{and} \end{cases}$$

$$\begin{cases} \gamma = L(\alpha - u\gamma) + q_0 = Lh_0 + q_0 \\ \delta = L(\beta - u\delta) + q_1 = Lh_1 + q_1 \end{cases}.$$

So the determination of $L(z)$ and $q(z) = q_0(z^2) + zq_1(z^2)$ may be done with Euclid:

$$\begin{cases} L = \text{quot}(\alpha - u\gamma, \gamma) = \text{quot}(h_0, \gamma) \\ q_0 = \text{rem}(\alpha - u\gamma, \gamma) = \text{rem}(h_0, \gamma) \\ q_1 = \text{rem}(\beta - u\delta, \delta) = \text{rem}(h_1, \delta). \end{cases} \quad (47)$$

Combining the two steps, the comparison of degrees is as follows:

$$\begin{cases} \deg(q_0) < \deg(h_0) < \deg(\gamma) \\ \deg(q_1) < \deg(h_1) < \deg(\delta) \end{cases}. \quad (48)$$

Two conclusions now follow:

- (i) the procedure may continue by recursion;
- (ii) the procedure must terminate.

Remark 6.2. In order to start the algorithm in (43) with direct reference to Euclid, we must have

$$\deg(\gamma) \leq \deg(\alpha) \tag{49}$$

where

$$A = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$

is the initial 2×2 matrix-function.

Now, suppose (49), i.e., that

$$\deg(\gamma) > \deg(\alpha).$$

Then determine a polynomial L such that

$$\deg(\gamma - L\alpha) \leq \deg(\alpha). \tag{50}$$

We may then start the procedure (43) on the matrix function

$$\begin{pmatrix} \alpha & \beta \\ \gamma - L\alpha & \delta \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -L & 1 \end{pmatrix} A.$$

If a polynomial U and a matrix function B is then found for

$$\begin{pmatrix} \alpha & \beta \\ \gamma - L\alpha & \delta \end{pmatrix}$$

then the factorization

$$A = \begin{pmatrix} 1 & 0 \\ L & 1 \end{pmatrix} \begin{pmatrix} 1 & U \\ 0 & 1 \end{pmatrix} B$$

holds; and the recursion will then work as outlined.

In the following, starting with a matrix-function A , we will always assume that the degrees of the polynomials $(A_{i,j})_{i,j \in \mathbb{Z}_N}$ have been adjusted this way, so the direct Euclidean algorithm can be applied.

6.2 The 3×3 case

The thrust of this section is the assertion that Theorem 5.10 holds with small modifications in the 3×3 case.

6.2.1 Comments:

In the definition of $A \in SL_3(\text{pol})$, it is understood that $A(z)$ has $\det A(z) \equiv 1$ and that the entries of the inverse matrix $A(z)^{-1}$ are again polynomials.

Note that if L, M, U and V are polynomials, then the four matrices

$$\begin{pmatrix} 1 & 0 & 0 \\ L & 1 & 0 \\ 0 & M & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ L & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & U & 0 \\ 0 & 1 & V \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 0 & U \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (51)$$

are in $SL_3(\text{pol})$ since

$$\begin{pmatrix} 1 & 0 & 0 \\ L & 1 & 0 \\ 0 & M & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -L & 1 & 0 \\ LM & -M & 1 \end{pmatrix} \quad \text{and} \quad (52)$$

$$\begin{pmatrix} 1 & U & 0 \\ 0 & 1 & V \\ 0 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & -U & UV \\ 0 & 1 & -V \\ 0 & 0 & 1 \end{pmatrix}. \quad (53)$$

Theorem 6.3. *Let $A \in SL_3(\text{pol})$; then the conclusion in Theorem 5.10 carries over with the modification that the alternating upper and lower triangular matrix-functions now have the form (51) or (52)-(53) where the functions L_j, M_j, U_j and $V_j, j = 1, 2, \dots$ are polynomials.*

6.3 The $N \times N$ case

Theorem 6.4. *Let $N \in \mathbb{Z}_+, N > 1$, be given and fixed. Let $A \in SL_N(\text{pol})$; then the conclusions in Theorem 5.10 carry over with the modification that the alternative factors in the product are upper and lower triangular matrix-functions in $SL_N(\text{pol})$. We may take the lower triangular matrix-factors $\mathcal{L} = (L_{i,j})_{i,j \in \mathbb{Z}_N}$ of the form*

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ L_p & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_{p+1} & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & . & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & . & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & . & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & L_{N-1} & 0 & 1 \end{pmatrix}$$

polynomial entries

$$\begin{cases} L_{i,i} \equiv 1, \\ L_{i,j}(z) = \delta_{i-j,p} L_i(z); \end{cases} \quad (54)$$

and the upper triangular factors of the form $\mathcal{U} = (U_{i,j})_{i,j \in \mathbb{Z}_N}$ with

$$\begin{cases} U_{i,i} \equiv 1, \\ L_{i,j}(z) = \delta_{i-j,p} U_i(z). \end{cases} \quad (55)$$

Proof. Notation. Let $U_1, \dots, U_N, L_1, \dots, L_N$ be polynomials and set

$$\mathcal{U}_N(U) = \begin{pmatrix} 1 & U_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & U_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & . & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & . & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & U_{N-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (56)$$

$$\mathcal{L}_N(L) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ L_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & . & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & . & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & . & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & L_{N-1} & 1 \end{pmatrix} \quad (57)$$

Note that both are in $SL_N(\text{pol})$; and we have

$$\mathcal{U}_N(U)^{-1} = \mathcal{U}_N(-U) \quad \text{and}$$

$$\mathcal{L}_N(L)^{-1} = \mathcal{L}_N(-L).$$

Step 1: Starting with $A = (A_{i,j}) \in SL_N(\text{pol})$. Then left-multiply with a suitably chosen $\mathcal{U}_N(-U)$ such that the degrees in the first column of $\mathcal{U}_N(-U)A$ decrease, i.e.,

$$\deg(A_{0,0}) \leq \deg(A_{1,0} - u_2 A_{1,0}) \leq \dots \leq \deg(A_{N-1,0}). \quad (58)$$

In the following, we shall use the same letter A for the modified matrix-function.

Step 2: Determine a system of polynomials L_1, \dots, L_{N-1} and a polynomial vector-function

$$\begin{bmatrix} f_0 \\ f_1 \\ \dots \\ f_{N-1} \end{bmatrix}$$

such that

$$A_N \begin{bmatrix} 1 \\ z \\ z^2 \\ \dots \\ z^{N-1} \end{bmatrix} = \mathcal{L}_N(L)_N \begin{bmatrix} f_0 \\ f_1 \\ \dots \\ f_{N-1} \end{bmatrix}, \quad (59)$$

or equivalently

$$\sum_{j=0}^{N-1} A_{i,j}(z^N)z^j = \begin{cases} f_0(z) & \text{if } i = 0 \\ L_i(z^N)f_{i-1}(z) + f_i(z) & \text{if } i > 0 \end{cases}.$$

Step 3: Apply the operators S_j and S_j^* from (21) to both sides in (59). First (59) takes the form:

$$\sum_{j=0}^{N-1} S_j A_{i,j} = \begin{cases} f_0 & \text{if } i = 0 \\ S_{f_{i-1}} L_i + f_i & \text{if } i > 0 \end{cases}.$$

For $i = 1$, we get

$$A_{1,j} = L_1 A_{0,j} + k_j \quad \text{where} \quad k_j = S_j^* f_1. \quad (60)$$

By (58) and the assumptions on the matrix-functions, we note that the system (60) may now be solved with the Euclidean algorithm:

$$\begin{cases} L_1 = \text{quot}(A_{0,j}, A_{1,j}) \\ k_j = \text{rem}(A_{0,j}, A_{1,j}) \end{cases} \quad (61)$$

with the same polynomial L_1 for $j = 0, 1, \dots, N-1$.

For the polynomial function f_1 we then have

$$f_1 = \sum_{j=0}^{N-1} S_j k_j; \quad (62)$$

i.e.

$$f_1(z) = k_0(z^N) + k_1(z^N)z + \dots + k_{N-1}(z^{N-1})z^{N-1}.$$

The process now continues recursively until all the functions $L_1, L_2, \dots, f_1, f_2, \dots$ have been determined.

Step 4: The formula (59) translates into a matrix-factorizations as follows: With L and F determined in (59), we get

$$A = \mathcal{L}_N(L)B \quad (63)$$

as a simple matrix-product taking $B = (B_{i,j})$ and

$$B_{i,j} = S_j^* f_i, \quad (64)$$

where we used Lemmas 4.10 and 5.1.

Step 5: The process now continues with the polynomial matrix-function from (63) and (64). We determine polynomials U_1, \dots, U_{N-1} and a third matrix function

$$C = (C(z)) = (C_{i,j}(z)) \quad \text{such that} \quad B = \mathcal{U}_N(U)C.$$

Step 6: As each step of the process we alternate L and U ; and at each step, the degrees of the matrix-functions is decreased. Hence the recursion must terminate as stated in Theorem 6.4. \square

7 Quantization

In addition to building algorithms for signal and image processing, there is the related problem of quantization. We define “quantization” broadly and indeed there is a variety of approaches.

Indeed the “signals” may have a subtle form; the time variable might correspond to numbers in a system of pixel grids. The tools we developed in the previous sections are sufficiently versatile. For clarity of discussion, it helps to separate quantization of the two sides, input and output; so for example, “time” as one and “magnitude” as the other. The idea is to select a finite set of possibilities on either side, be it points, e.g., by sampling; or one might make suitable selections of intervals on the two sides of the quantization problem.

In order to adapt to hardware and to reduce the number of computations, one selects a threshold. Specifically, when thresholding is applied to a set of numbers in an algorithm, the threshold function denoted Q below) sends quantities under a prescribed threshold (so insignificant relative to the selected threshold) to zero.

We will continue to use the method of thresholding as above: For signals, one makes a selection of a threshold and then programs the processing to implement that quantities under the threshold are discarded.

The same idea is used in image processing. In this case, the numbers in the process will instead be grayscale pixel values. Recall that digital images are represented by a matrix of grayscale pixels. In the case of a color image, it will instead be consisted of three such matrices, one for each of red, green and blue basic components.

In the thresholding process, applied to image processing, individual pixels are marked as object-pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as background-pixels otherwise; a convention known as threshold above. This contrasts threshold below, or threshold inside, where a pixel is labeled “object” if its value is between two thresholds; and threshold outside; the opposite of threshold inside.

Let $x \in \mathbb{R}$ be a pixel value. An example of thresholding called hard thresholding is defined as follows:

$$T(x) = \begin{cases} 0 & \text{if } |x| \leq \lambda \\ x & \text{if } |x| > \lambda, \end{cases} \quad (65)$$

where $\lambda \in \mathbb{R}_+$ is the thresholding value. [27]

Below we will outline briefly recursive quantization schemes. The purpose is to illustrate how the particular filters we developed in section 5, and choice of threshold function, have the effect of making the recursive quantization schemes run faster and be more effective. A popular method in recent papers (sigma delta quantization) is based on these ideas, plus the use of subtle difference/summation algorithms, see eg., [21, 20]

The literature on the subject is vast. A pioneering paper [3] opens up the door to the use of spectral analysis and stochastic processes, especially amenable to the present results. On the theoretical side, recent papers are relevant: [1, 6].

A key factor of the filtering algorithms from sections 3 and 6 is careful use of upsampling and downsampling. With a finite filter $(h_1 h_2, \dots)$, we get local input/output boxes (Fig. 4)

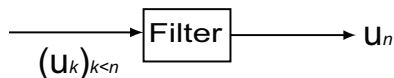


Figure 4: Standard filter. Input and output represented as time series; discrete time.

where

$$u_n = \sum_{j \geq 1} h_j u_{n-j} = h_1 u_{n-1} + h_2 u_{n-2} + \dots \quad (66)$$

or in matrix form

$$\begin{pmatrix} 0 & h_1 & h_2 & h_3 & \cdots \\ 0 & 0 & h_1 & h_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \quad (67)$$

For contrast, compare with the standard operator matrices from (26)

$$\begin{pmatrix} 0 & 0 & h_1 & h_2 & h_3 & h_4 & h_5 & \cdots \\ 0 & 0 & 0 & 0 & h_1 & h_2 & h_3 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & h_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \quad (68)$$

For emphasis, we give (68) in diagram form

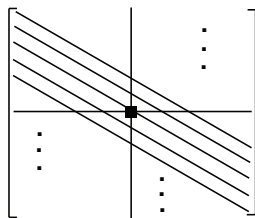


Figure 5: Filter operation with slanting. See Lemma 5.4. “Decimation” entails slanted matrix representations.

For the pyramid algorithm in Fig. 2, we use two versions of the slanted matrix in Fig. 5, high vs. low.

For the image processing (Fig. 3) we use four versions of the slanted matrices,

- (a) a matrix that takes the average in horizontal direction
- (b) a matrix that takes the average in vertical direction
- (c) a matrix that takes the difference in horizontal direction
- (d) a matrix that takes the difference in vertical direction.

which yield “average,” “horizontal,” “vertical” and “diagonal” details.

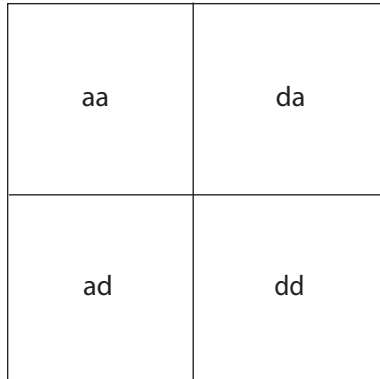


Figure 6: Level 1 decomposition on an image. Clockwise: Average, horizontal, diagonal and vertical details captured .

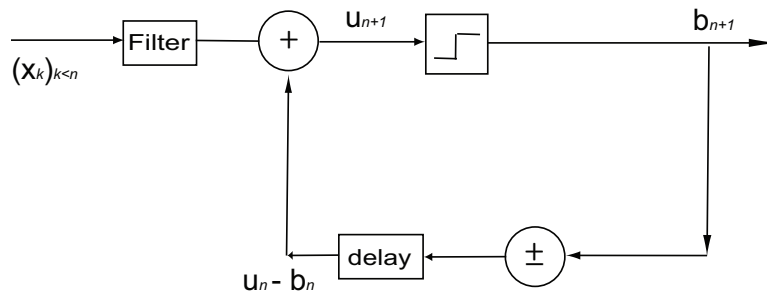


Figure 7: Quantization. The operations going into a typical quantization processing of a time series.

We now turn to the problem of quantizing signals. These signals could be the output in a processing (see sections 3 to 5). This step is often referred to as

a case of ‘Analogue to Digital’ (abbreviated A/D). Quantization in our present context entails a suitable symbolic encoding, turning data from a run of a signal process (involving sub-band filters as in section 4 above) into bits for subsequent computer-input. Here is how eq (69) illustrates this for the filters we constructed above. Quantization is essential in engineering applications; and the Q in (69) refers to a quantization operator.

$$\begin{cases} u_{n+1} = (Fu)_n + x_n - b_n \\ b_n = Q((Fu)_n + x_n) \end{cases} \quad (69)$$

Now take Figure 7 and equation (69) together. Fig 7 offers a sketch of a time series as it is processed in a simple quantization filter Q , see eq (69). The input is a signal x , represented as a time series and with a discrete time range k before n . So the input signal x is traced back from some fixed time n). The input is fed into one of the filters we designed in section 5. But now Fig 7 further illustrates the result of a loop for time n . The step from n to $n+1$ is spelled out in the first part in eq (69)) and the quantization operator Q is made precise in the second equation in (69). It is the loop in Fig 7 which involves thresholding and delay. Moving through the diagram in Fig 7, the next step is the process of adding the filtered signal to the output of a loop from time n . The resulting sum then contributes to time $n+1$. And the combined process is thus summarized in the discussion and in the visual in Fig. 7.

The filter F from the first eq in (69) and the first box in Fig 7, may be any one of those built in sections 4 through 6 above. So the particular filter F selected may itself be the result of a factorization algorithm as outlined above: It may be a time series, a wireless signal, or a system of pixel values; and in each case, it may involve any number of frequency bands.

The output from F (see Fig. 7) will pass through a thresholding filter Q , thus outputting b_{n+1} . In symbols, the next two steps are: ‘‘Take difference’’ and time-shift the result (‘‘delay’’), so from $n+1$ back to n . The first equation in (69) indicates how the process repeats itself, but with the output from the previous step, as input in the next.

ACKNOWLEDGEMENT

The authors thank Joo Hyun Song of the University of Iowa Electrical and Computer Engineering department for helpful discussions. The authors thank two anonymous referees for careful reading and for very helpful suggestions. The paper is better for it; both from the improvements in contents and in presentation.

References

- [1] Fatma Abdelkefi. Performance of sigma-delta quantizations in finite frames. *IEEE Trans. Inform. Theory*, 54(11):5087–5101, 2008.
- [2] N. I. Akhiezer. *The classical moment problem and some related questions in analysis*. Translated by N. Kemmer. Hafner Publishing Co., New York, 1965.
- [3] W. R. Bennett. Spectra of quantized signals. *Bell System Tech. J.*, 27:446–472, 1948.
- [4] Ola Bratteli and Palle Jorgensen. *Wavelets through a looking glass*. Applied and Numerical Harmonic Analysis. Birkhäuser Boston Inc., Boston, MA, 2002. The world of the spectrum.
- [5] Ola Bratteli and Palle E. T. Jorgensen. Wavelet filters and infinite-dimensional unitary groups. In *Wavelet analysis and applications (Guangzhou, 1999)*, volume 25 of *AMS/IP Stud. Adv. Math.*, pages 35–65. Amer. Math. Soc., Providence, RI, 2002.
- [6] John J. Benedetto, Onur Oktay, and Aram Tangboondouangjit. Complex sigma-delta quantization algorithms for finite frames. In *Radon transforms, geometry, and wavelets*, volume 464 of *Contemp. Math.*, pages 27–49. Amer. Math. Soc., Providence, RI, 2008.
- [7] Chris Brislawn and I. G. Rosen. Group lifting structures for multirate filter banks, i: Uniqueness of lifting factorizations.
- [8] Chris Brislawn and I. G. Rosen. Group lifting structures for multirate filter banks, ii: Uniqueness of lifting factorizations.
- [9] Chris Brislawn and I. G. Rosen. Wavelet based approximation in the optimal control of distributed parameter systems. *Numer. Funct. Anal. Optim.*, 12(1-2):33–77, 1991.
- [10] X. X. Chen and Y. Y. Chen. Self-lifting scheme: new approach for generating and factoring wavelet filter bank. *IET Signal Process.*, 2(4):405–414, 2008.
- [11] Ingrid Daubechies and Wim Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):247–269, 1998.
- [12] Yumin He, Xuefeng Chen, Jiawei Xiang, and Zhengjia He. Multiresolution analysis for finite element method using interpolating wavelet and lifting scheme. *Comm. Numer. Methods Engrg.*, 24(11):1045–1066, 2008.

- [13] Kenkichi Iwasawa. On some types of topological groups. *Ann. of Math. (2)*, 50:507–558, 1949.
- [14] A. Jensen and A. la Cour-Harbo. *Ripples in mathematics*. Springer-Verlag, Berlin, 2001. The discrete wavelet transform.
- [15] Palle E. T. Jorgensen and Myung-Sin Song. Analysis of fractals, image compression, entropy encoding, Karhunen-Loève transforms. *Acta Appl. Math.*, 108(3):489–508, 2009.
- [16] Wayne M. Lawton. Conjugate quadrature filters. In *Advances in wavelets (Hong Kong, 1997)*, pages 103–119. Springer, Singapore, 1999.
- [17] Wayne Lawton. Infinite convolution products and refinable distributions on Lie groups. *Trans. Amer. Math. Soc.*, 352(6):2913–2936, 2000.
- [18] Wayne Lawton. Global analysis of wavelet methods for Euler’s equation. *Mat. Model.*, 14(5):75–88, 2002. Second International Conference OFEA’2001 “Optimization of Finite Element Approximation, Splines and Wavelets” (Russian) (St. Petersburg, 2001).
- [19] Wayne M. Lawton. Hermite interpolation in loop groups and conjugate quadrature filter approximation. *Acta Appl. Math.*, 84(3):315–349, 2004.
- [20] B. W. K. Ling, C. Y. F. Ho, and J. D. Reiss. Control of sigma delta modulators via fuzzy impulsive approach. In *Control of chaos in nonlinear circuits and systems*, volume 64 of *World Sci. Ser. Nonlinear Sci. Ser. A Monogr. Treatises*, pages 245–270. World Sci. Publ., Hackensack, NJ, 2009.
- [21] M Lammers, A. M. Powell, and Özgür Yılmaz. Alternative dual frames for digital-to-analog conversion in sigma-delta quantization. *Adv. Comput. Math.*, 32(1):73–102, 2010.
- [22] Peng-Lang Shui, Zheng Bao, and Yuan Yan Tang. Three-band biorthogonal interpolating complex wavelets with stopband suppression via lifting scheme. *IEEE Trans. Signal Process.*, 51(5):1293–1305, 2003.
- [23] Myung-Sin Song. Wavelet image compression. In *Operator theory, operator algebras, and applications*, volume 414 of *Contemp. Math.*, pages 41–73. Amer. Math. Soc., Providence, RI, 2006.
- [24] Wim Sweldens and Dirk Roose. Shape from shading using parallel multigrid relaxation. In *Multigrid methods, III (Bonn, 1990)*, volume 98 of *Internat. Ser. Numer. Math.*, pages 353–364. Birkhäuser, Basel, 1991.
- [25] Wim Sweldens. The lifting scheme: a custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2):186–200, 1996.

- [26] Wim Sweldens. The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546 (electronic), 1998.
- [27] Walnut, D. F., *An Introduction to Wavelet Analysis*, (Birkhäuser, Boston, 2002).