UAB
Universitat Autònoma de Barcelona

Computational Approach to some Problems in Discrete Dynamical Systems

Salvador Borrós Culle

Universitat Autònoma de Barcelona
Departament de Matemàtiques

PhD Thesis

Computational Approach to some Problems in Discrete Dynamical Systems

Candidate:
Salvador Borrós Cullell

Advisor:
Lluís Alsedà i Soler

Thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in Mathematics

Declaration of Authorship

I, Salvador Borrós Cullell, declare that this thesis titled, “On some problems of Discrete Dynamical Systems” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Salvador Borrós Cullell
Date: October 30th 2023.

A la Maria i al Salvador petit en camí. Us estimo molt.

Acknowledgements

First of all, I would like to thank my advisor Prof. Lluís Alsedà for the wise guidance during the very long process the writing of this thesis has taken, always finding some time for me despite his very tight schedule.

També voldria agair als meus pares tot el suport que m'han donat durant els alts i baixos que acompanyen tota tesi doctoral. Un altre cop, han estat una font de serenor i consol en els moments més difícils.

Finalment, però no menys important, agrair a la Maria que hagi volgut seguir essent companya de vida fins i tot després del període final de redacció de la tesi. Sense ella tot hagués estat un ordre de magnitut més difícil.

Contents

Presentation

This thesis is divided into two parts. The first one is devoted to an algorithm to compute the rotation interval of circle maps. The other one deals with semi-analytic methods to compute the regularity of *Strange Nonchaotic Attractors (SNAs)*. The first part is completely contained in Chapter 1. On the other hand the second part is much longer, spanning Chapters 2, 3, 4, and 5. It also includes an introduction and conclusions of its own, as well as an annex.

In the first part, we present the development of an algorithm to compute the rotation interval of circle maps of degree one. The core of the algorithm is to compute the rotation number of non-decreasing circle maps of degree one that have a constant section. From this, the rotation interval of any circle map can be easily computed, since it is given by the rotation number of the upper and lower functions (sometimes called water functions), which are non-decreasing and have constant sections provided the original map is not non-decreasing. Hence, this algorithm fills the role of computing rotation intervals for cases that previously where out of reach. Mostly because the vast majority of existing algorithms rely on the differentiability of the map. The algorithm relies heavily in the relation between the existence of periodic points and the rotation number. Intuitively, it uses the constant section as a catching net for finding periodic points, which can then be used to compute rotation numbers.

The second part is an extension of the research started by Alsedà, Mondelo and Romero in [**AMR16**], and was later expanded in David Romero's PhD thesis [**RiS15**]. The main idea is to use the rich existing theory of wavelets as basis of function spaces to extract information from invariant objects, in particular attractors, in quasi-periodic forced skew products. That is families of maps on $\mathbb{S}^1 \times \mathbb{R}$ of the form

$$\begin{array}{rccc} \mathfrak{F}_{\sigma,\varepsilon}: & \mathbb{S}^1 \times \mathbb{R} & \longrightarrow & \mathbb{S}^1 \times \mathbb{R} \\ & (\theta, x) & \longmapsto & (R_\omega(\theta), F_{\sigma,\varepsilon}(\theta, x)), \end{array}$$

where $\omega \in \mathbb{R}\backslash\mathbb{Q}$ and $R_\omega(\theta) = \theta + \omega \mod 1$. In this case we consider that the map $\mathfrak{F}$ might have some parametric dependence on σ and ε, and part of our goal is to study how a change of parameters might affect the properties of the attractor by studying its wavelet expansion. We will mostly centre our attention in the computation of the regularity of the invariant object. This is mostly due to the fact that quasi-periodic forced skew products serve as toy models for finding strange nonchaotic attractors (or SNA). These are exactly what they name implies, attractors of a nonchaotic system that despite having regular dynamics have a very complicated geometry. The first SNA was published in 1984 in [**GOPY84**]. Since then, SNAs have been object of study, with some analytical results proving their existence [**Kel96, AM08**]. However, there exists many systems in which existence of SNA is suspected just from numerical approximations [**Kan84, NK96**].

There exist some results that allow us to compute regularity of functions from their wavelet expansion. We want to use this measure of regularity as a (flawed) way to detect wether or not an attractor might be strange. We will use the systems with known behaviour as a proof of adequacy of our algorithm, and then we will try to use the same method to try to understand the behaviour of systems with suspected strangeness, in particular the one found in [**NK96**].

For a more in depth explanation we refer the reader to the introduction found in each part.

Part 1

An algorithm to compute Rotation Numbers of circle maps

CHAPTER 1

An algorithm to compute Rotation Numbers for Circle Maps

1.1. Introduction

The rotation interval plays an important role in combinatorial dynamics. For example Misiurewicz's Theorem [**Mis82**] links the set of periods of a continuous lifting F of degree one to the set $M := \left\{n \in \mathbb{N} : \frac{k}{n} \in \operatorname{Rot}(F) \text{ for some integer } k\right\}$, where $\operatorname{Rot}(F)$ denotes the rotation interval of F. Moreover, it is natural to compute lower bounds of the topological entropy depending on the rotation interval [**ALMM88**]. In any case, the knowledge of the rotation interval of circle maps of degree one is of theoretical importance.

The rotation number was introduced by H. Poincaré to study the movement of celestial bodies [**Poi85**], and since then has been found to model a wide variety of physical and sociological processes. In the physical sense, it has been recently applied to climate science [**MSG20**]. In the sociological one, the application to voting theory [**JO19, MO15**] is specially surprising in this context.

The computation of the rotation number for invertible maps of degree one from $\mathbb{S}^1$ onto itself is well studied, and many very efficient algorithms exist for its computation [**Her79, Pav95, SV06, Vel88**]. However, there is a lack of an efficient algorithm for the non-invertible and non-differentiable case.

In this chapter, we propose a method that allows us to compute the rotation interval for the non-invertible case. Our algorithm is based on the fact that we can compute *exactly* the rotation number of a natural subclass of the class of continuous non-decreasing degree one circle maps that have a constant section and a *rational rotation number.* From this algorithm we get an efficient way to compute *exactly* the rotation interval of a large subset of the continuous non-invertible degree one circle maps by using the so called *upper* and *lower* maps, which, when different, always have a constant section. When dealing with maps outside the aforementioned class, the algorithm will return an arbitrarily precise rational aproximation of the rotation number.

To check the efficiency of our algorithm we will use it to compute some classical results such as a Devil's Staircase. When doing so, we will compare the efficiency of our algorithm with the performance of some other algorithms that have been traditionally used under the hypothesis of non-invertibility. On the other hand, we will also compute the rotation interval and Arnold tongues for a variety of maps, in the same comparing spirit. These maps include the Standard Map and some of its variants but have issues either with the differentiability, or even with the continuity. Of course these variants are not well suited for algorithms that strongly use differentiability.

The chapter is organised as follows. In Section 1.2 the theoretical background will be set. In Section 1.3 the algorithm will be presented, and, finally in Section 1.4 we will provide the mentioned examples of the use of the algorithm.

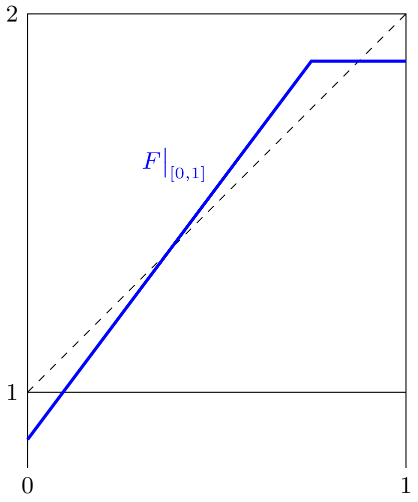


FIGURE 1.1. An example of a map from $\mathcal{L}_1$ which can be considered as a toy model for the elements of that class. The picture shows $F\big|_{[0,1]}$, and F is globally defined as $F(x) = F\big|_{[0,1]}(\{\!\{x\}\!\}) + \lfloor x \rfloor$.

1.2. A short Survey on Rotation Theory and the Computation of Rotation Numbers

We will start by recalling some results from the rotation theory for circle maps. To do this we will follow [**ALM00**].

The *floor function* (i.e. the function that returns the greatest integer less than or equal to the variable) will be denoted as $\lfloor \cdot \rfloor$. Also the *decimal part* of a real number $x \in \mathbb{R}$, defined as $x - \lfloor x \rfloor \in [0, 1)$ will be denoted by $\{\!\{x\}\!\}$.

In what follows $\mathbb{S}^1$ denotes the circle, which is defined as the set of all complex numbers of modulus one. Let $e \colon \mathbb{R} \longrightarrow \mathbb{S}^1$ be the natural projection from $\mathbb{R}$ to $\mathbb{S}^1$, which is defined by $e(x) := \exp(2\pi i x)$.

Let $f \colon \mathbb{S}^1 \longrightarrow \mathbb{S}^1$ be continuous map. A continuous map $F \colon \mathbb{R} \longrightarrow \mathbb{R}$ is a *lifting of* f if and only if $e(F(x)) = f(e(x))$ for every $x \in \mathbb{R}$. Note that the lifting of a circle map is not unique, and that any two liftings F and F' of the same continuous map $f \colon \mathbb{S}^1 \longrightarrow \mathbb{S}^1$ verify $F = F' + k$ for some $k \in \mathbb{Z}$.

For every continuous map $f \colon \mathbb{S}^1 \longrightarrow \mathbb{S}^1$ there exists an integer d such that

$$F(x + 1) = F(x) + d$$

for every lifting F of f and every $x \in \mathbb{R}$ (that is, the number d is independent of the choice of the lifting and the point $x \in \mathbb{R}$). We shall call this number d *the degree of* f. The degree of a map roughly corresponds to the number of times that the whole image of the map f covers $\mathbb{S}^1$ homotopically.

In this chapter we are interested studying maps of degree 1, since the rotation theory is well defined for the liftings of these maps.

We will denote the set of all liftings of maps of degree 1 by $\mathcal{L}_1$. Observe that to define a map from $\mathcal{L}_1$ it is enough to define $F\big|_{[0,1]}$ (see Figure 1.1) because F can be globally defined as $F(x) = F\big|_{[0,1]}(\{\!\{x\}\!\}) + \lfloor x \rfloor$ for every $x \in \mathbb{R}$.

REMARK 1.1. It is easy to see that, for every $F \in \mathcal{L}_1$, $F^n(x + k) = F^n(x) + k$ for every $n \in \mathbb{N}$, $x \in \mathbb{R}$ and $k \in \mathbb{Z}$. Consequently, $F^n \in \mathcal{L}_1$ for every $n \in \mathbb{N}$. ∎

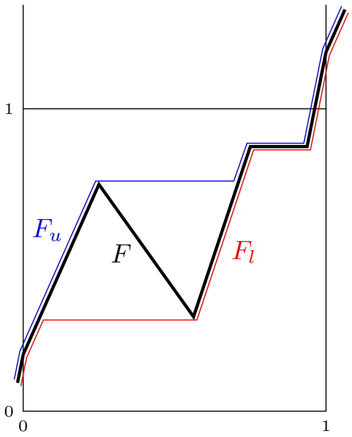


FIGURE 1.2. An example of a map $F \in \mathcal{L}_1$ with its lower map F_l in red and its upper map F_u in blue.

DEFINITION 1.2. Let $F \in \mathcal{L}_1$, and let $x \in \mathbb{R}$. We define the *rotation number of* x as

$$\rho_F(x) := \limsup_{n\to\infty} \frac{F^n(x) - x}{n}.$$

Observe (Remark 1.1) that, $\rho_F(x) = \rho_F(x+k)$ for every $k \in \mathbb{Z}$. The *rotation set of* F is defined as:

$$\operatorname{Rot}(F) = \{\rho_F(x) : x \in \mathbb{R}\} = \{\rho_F(x) : x \in [0,1]\}.$$

Ito [**Ito81**], proved that the rotation set is a closed interval of the real line. So, henceforth the set $\operatorname{Rot}(F)$ will be called the *rotation interval of* F. ■

PROPOSITION 1.3 (Proposition 3.7.11 in [**ALM00**]). *Let $F \in \mathcal{L}_1$ be non-decreasing. Then, for every $x \in \mathbb{R}$ the limit*

$$\lim_{n\to\infty} \frac{F^n(x) - x}{n}$$

exists and is independent of x.

For a non-decreasing map $F \in \mathcal{L}_1$, the number $\rho_F(x) = \lim_{n\to\infty} \frac{F^n(x)-x}{n}$ will be called the *rotation number of* F, and will be denoted by ρ_F.

Now, by using the notation from [**ALM00**], we will introduce the notion of upper and lower functions, that will be crucial to compute the rotation interval.

DEFINITION 1.4. Given $F \in \mathcal{L}_1$ we define the F*-upper map* F_u as

$$F_u(x) := \sup\{F(y) : y \leq x\}.$$

Similarly we will define the F*-lower map* as

$$F_l(x) := \inf\{F(y) : y \geq x\}.$$

An example of such functions is shown in Figure 1.2. ■

It is easy to see that $F_l, F_u \in \mathcal{L}_1$ are non decreasing, and $F_l(x) \leq F(x) \leq F_u(x)$ for every $x \in \mathbb{R}$.

The rationale behind introducing the upper and lower functions comes from the following result, stating that the rotation interval of a function $F \in \mathcal{L}_1$ is given by the rotation number of its upper and lower functions.

THEOREM 1.5 (Theorem 3.7.20 in [**ALM00**]). *Let $F \in \mathcal{L}_1$. Then,*

$$\operatorname{Rot}(F) = \left[\rho_{F_l}, \rho_{F_u}\right].$$

Note that this theorem makes indeed sense, since the upper and lower functions are non-decreasing and by Proposition 1.3 they have a single well defined rotation number.

Let $f\colon \mathbb{S}^1 \longrightarrow \mathbb{S}^1$ and let $z \in \mathbb{S}^1$. The *f-orbit of z* is defined to be the set

$$\operatorname{Orb}_f(z) := \{z, f(z), f^2(z), \dots, f^n(z), \dots\}.$$

We say that z is an *n-periodic point of f* if $\operatorname{Orb}_f(z)$ has cardinality n. Note that this is equivalent to $f^n(z) = z$ and $f^k(z) \neq z$ for every $k < n$. In this case the set $\operatorname{Orb}_f(z)$ will be called an *n-periodic orbit* (or, simply, a *periodic orbit*).

If we have a periodic orbit of a circle map, a natural question that might arise is how it behaves at a lifting level. This motivates the introduction of the notion of a *lifted cycle.*

Given a set $A \subset \mathbb{R}$ and $m \in \mathbb{Z}$ we will denote $A + m := \{x + m : x \in A\}$. Analogously, we set

$$A + \mathbb{Z} := \{x + m : x \in A,\ m \in \mathbb{Z}\}.$$

Definition 1.6. Let $f\colon \mathbb{S}^1 \longrightarrow \mathbb{S}^1$ be a continuous map and let F be a lifting of f. A set $P \subset \mathbb{R}$ is called a *lifted cycle of F* if $e(P)$ is a periodic orbit of f. Observe that, then $P = P + \mathbb{Z}$. The period of a lifted cycle is, by definition, the period of $e(P)$. Hence, when $e(P)$ is an n-periodic orbit of f, P is called an *n-lifted cycle*, and every point $x \in P$ will be called an *n-periodic* (mod 1) *point of F.* ∎

The relation between lifted orbits and rotation numbers is clarified by the next lemma.

Lemma 1.7 (Lemmas 3.7.2 and 3.7.3 in [**ALM00**]). *Let $F \in \mathcal{L}_1$. Then, x is an n-periodic* (mod 1) *point of F if and only if there exists $k \in \mathbb{Z}$ such that $F^n(x) = x + k$ but $F^j(x) - x \notin \mathbb{Z}$ for $j = 1, 2, \dots, n-1$. In this case,*

$$\rho_F(x) = \lim_{m\to\infty} \frac{F^m(x) - x}{m} = \frac{k}{n}.$$

Moreover, let P be a lifted n-cycle of F. Every point $x \in P$ is an n-periodic (mod 1) *point of F, and the above number k does not depend on x. Hence, for every $x \in P$ we have $\rho_F(P) := \rho_F(x) = \frac{k}{n}$.*

Now we can revisit Proposition 1.3:

Proposition 1.3 (Proposition 3.7.11 in [**ALM00**]). *Let $F \in \mathcal{L}_1$ be non-decreasing. Then, for every $x \in \mathbb{R}$ the limit*

$$\rho_F := \lim_{n\to\infty} \frac{F^n(x) - x}{n}$$

exists and is independent of x. Moreover, ρ_F is rational if and only if F has a lifted cycle.

In the next two subsections we will give a survey of two known algorithms that have been already used to compute rotation numbers of non-differentiable and non-invertible liftings from $\mathcal{L}_1$. The first one (Algorithm 1) stems automatically from the definition of rotation number (Definition 1.2); the other one (Algorithm 2) is due to Simó et al. [**JS09**].

1.2.1. Algorithm 1: computing the rotation interval from the definition of rotation number.

The first algorithm to compute ρ_F consists in using Proposition 1.3 and the following approximation:

$$\begin{aligned}\rho_{_F} &= \lim_{m\to\infty} \frac{F^m(x) - x}{m}\\ &\approx \left.\frac{F^n(x) - x}{n}\right|_{x=0}\\ &= \frac{F^n(0)}{n},\end{aligned}$$

with $n \in \mathbb{N}$ large enough. However, *a priori* we do not know how good the convergence is. In Lemma 1.8 we will show that the error is of order $1/n$. The implementation of the computation of this approximation to the rotation number can be found in the side algorithm pseudocode.

Algorithm 1
Direct Algorithm pseudocode

```
procedure ROTATION_NUMBER($F$, error)
    $n \leftarrow$ CEIL$\left(\frac{1}{\texttt{error}}\right)$
    $x \leftarrow 0$
    $k \leftarrow 0$
    for $i \leftarrow 1, n$ do
        $x \leftarrow F(x)$
        $s \leftarrow$ FLOOR$(x)$
        $k \leftarrow k + s$                  $\triangleright\ k = \lfloor F^n(0) \rfloor$
        $x \leftarrow x - s$   $\triangleright\ x = \{\{F^n(0)\}\} = F^n(0) - k$
    end for
    return $\frac{k+x}{n}$
end procedure
```

Since the maps from $\mathcal{L}_1$ are defined so that

$$F(x) = F\big|_{[0,1]}\big(\{\{x\}\}\big) + \lfloor x \rfloor,$$

we need to evaluate the function FLOOR$(\cdot) = \lfloor \cdot \rfloor$ once per iterate. So, for clarity and efficiency, it seems advisable to split $F^n(0)$ as $\{\{F^n(0)\}\} + \lfloor F^n(0) \rfloor$. The next lemma clarifies the computation error as a function of the number of iterates. In particular it explicitly gives the necessary number of iterates to obtain the rotation number woth a given a desired tolerance.

For every non-decreasing lifting $F \in \mathcal{L}_1$, and every $n \in \mathbb{N}$ we set (see Figure 1.3)

$$\ell_F(n) := \min_{x\in\mathbb{R}} \lfloor F^n(x) - x \rfloor = \min_{x\in[0,1]} \lfloor F^n(x) - x \rfloor .$$

The second equality holds because F has degree 1, and hence $\ell_F(n)$ is well defined. See Figure 1.3 for an intuitive representation of what $\ell_F(n)$ means.

LEMMA 1.8. *For every non-decreasing lifting $F \in \mathcal{L}_1$ and $n \in \mathbb{N}$ we have*

(a) either $F^n(z) = z + \ell_F(n) + 1$ for some $z \in \mathbb{R}$, or
$x + \ell_F(n) \le F^n(x) < x + \ell_F(n) + 1$ for every $x \in \mathbb{R}$;
(b) $\frac{\ell_F(n)}{n} \le \rho_{_F} \le \frac{\ell_F(n)+1}{n}$; and
(c) $\left|\rho_{_F} - \frac{F^n(x)-x}{n}\right| < \frac{1}{n}$ for every $x \in \mathbb{R}$.

PROOF. We will prove the whole lemma by considering two alternative cases. Assume first that $F^n(z) = z + \ell_F(n) + 1$ for some $z \in \mathbb{R}$. Then (a) holds trivially, and Proposition 1.3 and Lemma 1.7 imply that $\rho_{_F} = \frac{\ell_F(n)+1}{n}$. So, Statement (b) also holds in this case. Now observe that from the definition of $\ell_F(n)$ we have

$$\ell_F(n) \le \lfloor F^n(x) - x \rfloor \le F^n(x) - x \tag{1.1}$$

for every $x \in \mathbb{R}$. Moreover, there exists $k = k(x) \in \mathbb{Z}$ such that $x \in [z+k, z+k+1)$ and, since F is non-decreasing, so is F^n. Thus,

$$\begin{aligned}F^n(x) - x \le F^n(z+k+1) - x = F^n(z) + k + 1 - x =&\\ \ell_F(n) + 1 + (z+k+1-x) < \ell_F(n) + 2,&\end{aligned}$$

by Remark 1.1. Consequently,

$$\rho_{_F} - \frac{1}{n} = \frac{\ell_F(n)}{n} \le \frac{F^n(x) - x}{n} < \rho_{_F} + \frac{1}{n};$$

which proves (c) in this case.

Now we consider the case

$$F^n(x) \neq x + \ell_F(n) + 1$$

for every $x \in \mathbb{R}$. In view of the definition of $\ell_F(n)$, we cannot have

$$F^n(x) - x > \ell_F(n) + 1$$

for every $x \in \mathbb{R}$. Hence, by the continuity of $F^n(x) - x$ and (1.1),

$$\ell_F(n) \leq F^n(x) - x < \ell_F(n) + 1 \tag{1.2}$$

for every $x \in \mathbb{R}$. This proves (a).

Now we prove (b). We consider the functions: $x \longmapsto \ell_F(n) + x$, F^n, and $x \longmapsto \ell_F(n) + 1 + x$. They are all non-decreasing and, by Remark 1.1, they belong to $\mathcal{L}_1$. Hence, by Proposition 1.3, [**ALM00**, Lemma 3.7.19] and (1.2),

$$\ell_F(n) = \rho_{x \mapsto \ell_F(n)+x} \leq \rho_{F^n} \leq$$
$$\rho_{x \mapsto \ell_F(n)+1+x} = \ell_F(n) + 1.$$

Consequently,

$$\frac{\ell_F(n)}{n} \leq \rho_F = \frac{\rho_{F^n}}{n} \leq \frac{\ell_F(n)+1}{n},$$

and (b) holds. Moreover, (1.2) is equivalent to

$$\frac{\ell_F(n)}{n} \leq \frac{F^n(x) - x}{n} \leq \frac{\ell_F(n)+1}{n},$$

which proves (c).

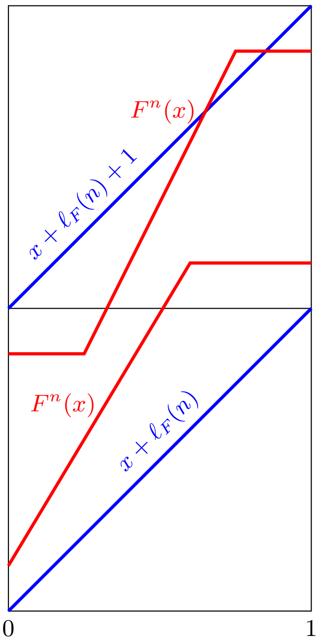


FIGURE 1.3. Plot of $x + \ell_F(n)$ $x + \ell_F(n) + 1$, and $F^n(x)$ for two arbitrary non-decreasing maps $F \in \mathcal{L}_1$ that fit in the two cases of the lemma.

□

1.2.2. Algorithm 2: the Simó et al. algorithm to compute the rotation interval. First of all, it should be noted that even though the authors propose an algorithm to compute the rotation interval for a general map $F \in \mathcal{L}_1$, we will only use it for non decreasing maps. *A priori* this algorithm is radically different from Algorithm 1 and it gives an estimate of ρ_F by providing and upper and a lower bound of the rotation number (rotation interval in the original paper) of F. Moreover, it is implicitly assumed that $\rho_F \in [0, 1]$ (in particular that $F(0) \in [0, 1)$ — this can be achieved by replacing the lifting F by the lifting $G := F - \lfloor F(0) \rfloor$, if necessary). The algorithm goes as follows:

(Alg. 2-1) Decide the number of iterates n in function of a given tolerance.

(Alg. 2-2) For $i = 0, 1, 2, \ldots, n$ compute $k_i = \lfloor F^i(x_0) \rfloor$ and $\alpha_i = F^i(x_0) - k_i$ (i.e. α_i is the fractionary part of $F^i(x_0)$).

(Alg. 2-3) Sort the values of k_i and α_i so that $\alpha_{i_0} < \alpha_{i_1} < \cdots < \alpha_{i_n}$ (this can be achieved efficiently with the help of an index vector).

(Alg. 2-4) Initialise $\rho_{\min} = 0$ and $\rho_{\max} = 1$.

(Alg. 2-5) For $j = 0, 1, 2, \ldots, n-1$ set $\rho_j = \frac{k_{i_{j+1}} - k_{i_j}}{i_{j+1} - i_j}$, and
- if $i_{j+1} > i_j$ set $\rho_{\min} = \max\{\rho_{\min}, \rho_j\}$; otherwise,
- if $i_{j+1} < i_j$ set $\rho_{\max} = \min\{\rho_{\max}, \rho_j\}$.

(Alg. 2-6) Return $\rho_{\max}$ and $\rho_{\min}$ as upper and lower bounds of the rotation number of F, respectively.

The real issue in this algorithm consists in dealing with the error. If the rotation number ρ_F satisfies a Diophantine condition $\left|\rho_F - \frac{p}{q}\right| \leq cq^{-\nu}$, with $c > 0$ and $\nu \geq 2$,

Algorithm 2 Simó et al. ([**JS09**]) Algorithm in pseudocode

```
procedure ROTATION_NUMBER($F, n$)
    index[] $\leftarrow$
    $x \leftarrow 0$
    $\rho_{\min} \leftarrow 0$
    $\rho_{\max} \leftarrow 1$
    for $i \leftarrow 0, n$ do
        $x \leftarrow F(x)$
        $k_i \leftarrow$ FLOOR($x$)
        $\alpha_i \leftarrow x - k_i$
        index[$i$] $\leftarrow i$
    end for
    sort $\alpha$[index[$i$]] by rearranging index[]
    for $i \leftarrow 0, n-1$ do
        $\rho_{\text{aux}} \leftarrow \frac{k_{\texttt{index}[i+1]} - k_{\texttt{index}[i]}}{\texttt{index}[i+1] - \texttt{index}[i]}$
        if index[$i+1$] $>$ index[$i$] then
            $\rho_{\min} \leftarrow \max\{\rho_{\min}, \rho_{\text{aux}}\}$
        else
            $\rho_{\max} \leftarrow \min\{\rho_{\max}, \rho_{\text{aux}}\}$
        end if
    end for
    return $\rho_{\min}, \rho_{\max}$
end procedure
```

then the error verifies

$$\varepsilon < \frac{1}{(cn^{\nu})^{\frac{1}{\nu-1}}}.$$

Note that this error depends strongly on the chosen number n of iterates, and that *n must be chosen before knowing what the rotation number could possibly be.* Hence Algorithm 2 it is not well suited to compute *unknown* rotation numbers of $\mathcal{L}_1$ maps. However, it is excellent in continuation methods where the current rotation number gives a good estimate of the next one.

REMARK 1.9. The original aim of the algorithm is to determine the existence of closed invariant curves on dynamical systems on the plane rather than the computation of rotation numbers of a given map of the circle. The rationale of the algorithm is that if, after computing $\rho_{\min}$ and $\rho_{\max}$, we find that $\rho_{\min} > \rho_{\max}$ then the computed orbit cannot lay on a closed invariant curve. This explains most of the limitations we have encountered, such as the lack of an *a priori* estimate of the error, or the fact that the algorithm is suited only for rotation numbers $\rho \in [0, 1]$. ∎

1.3. An algorithm to compute rotation numbers of non-decreasing maps with a constant section

The *diameter of an interval K* which, by definition is equal to the absolute value of the difference between their endpoints, will be denoted as $\operatorname{diam}(K)$. We say that an interval K is a *non-degenerate subinterval* if $\operatorname{diam}(K) > 0$. In particular, a non-degenerate interval is different from a point, i.e it has non-empty interior.

A *constant section of a lifting F* of a circle map is a closed non-degenerate subinterval K of $\mathbb{R}$ such that $F\big|_K$ is constant. In the special case when $F \in \mathcal{L}_1$, we have that $F(x+1) = F(x) + 1 \neq F(x)$ for every $x \in \mathbb{R}$. Hence, $\operatorname{diam}(K) < 1$.

The algorithm we propose is partly based on Lemma 1.8. However, it is the following simple proposition which allows us to compute the *exact* rotation number of a non-decreasing lifting from $\mathcal{L}_1$ that has a constant section, provided that $F^n(K) \cap (K + \mathbb{Z}) \neq \emptyset$. In this sense, Proposition 1.10 has a completely different approach of the problem. Instead of trying to *(costly) estimate* the rotation number as Algorithm 3 and Lemma 1.8 do, it seeks to find the exact rotation number. The contribution of Lemma 1.8 is still important, since it will allow us to have a fail safe in case Proposition 1.10 cannot be applied.

Proposition 1.10. *Let $F \in \mathcal{L}_1$ be non-decreasing and have a constant section K. Assume that there exists $n \in \mathbb{N}$ such that $F^n(K) \cap (K + \mathbb{Z}) \neq \emptyset$, and that n is minimal with this property. Then, there exists $\xi \in \mathbb{R}$ such that $F^n(K) = \{\xi\} \subset K + m$ with $m = \lfloor \xi - \min K \rfloor \in \mathbb{Z}$, ξ is an n-periodic* (mod 1) *point of F, and $\rho_{_F} = \frac{m}{n}$.*

Proof. Since K is a constant section of F, $F(K)$ contains a unique point, and hence there exists $\xi \in \mathbb{R}$ such that $F^n(K) = \{\xi\}$. Then, the fact that $F^n(K) \cap (K + \mathbb{Z}) \neq \emptyset$ implies that $\xi \in K + m$ with $m = \lfloor \xi - \min K \rfloor \in \mathbb{Z}$.

Set $\widetilde{\xi} := \xi - m \in K$. Then, $\left\{F^n\big(\widetilde{\xi}\big)\right\} = F^n(K) = \left\{\widetilde{\xi} + m\right\}$. Moreover, the minimality of n implies that $F^j\big(\widetilde{\xi}\big) - \widetilde{\xi} \notin \mathbb{Z}$ for $j = 1, 2, \dots, n-1$. So, Lemma 1.7 tells us that $\widetilde{\xi}$ (and hence ξ) is an n-periodic (mod 1) point of F. Thus, $\rho_{_F} = \frac{m}{n}$ by Proposition 1.3. □

As already said, Proposition 1.10 is a tool to compute *exactly* the rotation numbers of non-decreasing liftings $F \in \mathcal{L}_1$ which have a constant section and have a lifted cycle intersecting the constant section (and hence having rational rotation number). In the next subsection we shall investigate how restrictive are these conditions, when dealing with computation of rotation numbers.

1.3.1. On the genericity of Proposition 1.10.

First observe that the fact that Proposition 1.10 only allows the computation of rotation numbers of non-decreasing liftings $F \in \mathcal{L}_1$ which have a constant section is not restrictive at all. Indeed, if we want to compute rotation intervals of *non-invertible* continuous circle maps of degree one, Theorem 1.5 tells us that this is exactly what we want.

Clearly, one of the real restrictions that cannot be overcome in the above method to compute *exact* rotation numbers is that it only works for maps having a rational rotation number. Hence, for maps with non-rational rotation number we can only hope to get a rational approximation like the one given by Algorithm 1, which can be archieved with arbitrary precision. However, as stated by Theorem 1.16, it is not easy to find maps with irrational rotation number, even harder to do so with floating point aproximation.

On the other hand, we also have the formal restriction that Proposition 1.10 requires that the map F has a lifted cycle intersecting the constant section (indeed this is a consequence of the condition $F^n(K) \cap (K + \mathbb{Z}) \neq \emptyset$). A natural question is whether this restriction is just formal or it is a real one. In the next example we will see that the restriction is not superfluous since there exist maps which do not satisfy it.

Consequently, Proposition 1.10 is useless in computing the rotation numbers of non-decreasing liftings in $\mathcal{L}_1$ which have a constant section and either irrational rotation number or rational rotation number but do not have any lifted cycle intersecting the constant section. The only reasonable solution to these problems is to use an iterative algorithm to estimate the rotation number with a prescribed error, such as Algorithm 1, Algorithm 2 or others.

EXAMPLE 1.11. *There exist non-decreasing liftings in $\mathcal{L}_1$ which have a constant section and rational rotation number but do not have any lifted cycle intersecting the constant section:* Let $F \in \mathcal{L}_1$ be the map such that $F(x) = F\big|_{[0,1]}\big(\{\!\{x\}\!\}\big) + \lfloor x \rfloor$ for every $x \in \mathbb{R}$, and let

$$F\big|_{[0,1]}(x) := \begin{cases} x + 0.2 & \text{if } x \in [0, 0.1], \\ \frac{x}{2} + 0.25 & \text{if } x \in [0.1, 0.3], \\ 7x - 1.7 & \text{if } x \in [0.3, 0.4], \\ \frac{x}{4} + 1 & \text{if } x \in [0.4, 0.8], \\ 1.2 & \text{if } x \in [0.8, 1]. \end{cases}$$

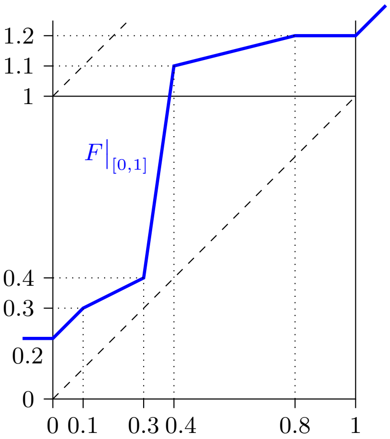


FIGURE 1.4. Example of a non-decreasing lifting in $\mathcal{L}_1$ with a constant section and rational rotation number which does not verify the assumptions of Proposition 1.10.

The map F is a non-decreasing lifting from $\mathcal{L}_1$, having a constant section $K = [0.8, 1]$ and rotation number $\frac{1}{3}$ given by the 3-lifted cycle $P = \{0.1, 0.3, 0.4\} + \mathbb{Z}$ (c.f. Lemma 1.7 and Proposition 1.3).

Now let us see that F does not have any lifted cycle intersecting the constant section. First, observe that

$$F^3(K) = F(F(F(K))) = F(F(\{1.2\})) = F(\{1.35\}) = \{1.75\} \not\subset K + \mathbb{Z}.$$

Hence, there is no lifted cycle of period 3 intersecting K. On the other hand, again by Lemma 1.7, we have that if x is an n-periodic (mod 1) point of F then there exists $k \in \mathbb{Z}$ such that $F^n(x) = x + k$ and

$$\frac{1}{3} = \rho_{F} = \lim_{m\to\infty} \frac{F^m(x) - x}{m} = \rho_{F}(x) = \frac{k}{n}.$$

Moreover, since F is non-decreasing, we know by [**ALM00**, Corollary 3.7.6] that n and k must be relatively prime. Thus, any lifted cycle of F has period 3, and from above this implies that there is no lifted cycle intersecting K. ■

1.3.2. Algorithm 3: A constant section based algorithm arising from Proposition 1.10. From the last paragraph of the previous subsection it becomes evident that Proposition 1.10 does not give a complete algorithm to compute rotation numbers of non-decreasing liftings in $\mathcal{L}_1$ which have a constant section. Such an algorithm must rather be a mix-up of Proposition 1.10, and Algorithm 1 to be used when we are not able to determine whether we are in the assumptions of that proposition. As we did for Algorithm 1, we will split $F^n(0)$ as $\{\!\{F^n(0)\}\!\} + \lfloor F^n(0) \rfloor$. The goal is twofold, on the one hand splitting helps minimizing the truncation errors. On the other hand, thanks to the splitting we can apply Proposition 1.10 more efficiently, since it requires the computation of m as an integer part. Note that here we are denoting the constant section by K and assuming that $0 \in K$, which will be justified later. Then, observe that the computations to be performed are exactly the same in both cases (meaning when we can use Proposition 1.10, and when alternatively we must end up by using Algorithm 1); except for the conditionals that check whether there exists $n \leq$ `max_iter` such that $F^n(K) \cap (K + \mathbb{Z}) \neq \emptyset$

Algorithm 3 Constant Section Based Algorithm

For a non-decreasing map $F \in \mathcal{L}_1$ parametrised so that $[-\texttt{tol}, \beta + \texttt{tol}]$ is a constant section of F

```
define tol ←                      ▷ Procedure parameter that bounds the rounding errors in the computation of $F^n(0)$
procedure ROTATION_NUMBER($F$, $\beta$, error)
    max_iter ← CEIL$\left(\frac{1}{\texttt{error}}\right)$      ▷ Maximum number of iterates allowed (to estimate the rotation number with the prescribed error when reached)
    $x \leftarrow 0$
    $m \leftarrow 0$
    for $n \leftarrow 1$, max_iter do
        $x \leftarrow F(x)$
        $s \leftarrow$ FLOOR$(x)$
        $m \leftarrow m + s$                            ▷ $m = \lfloor F^n(0) \rfloor$
        $x \leftarrow x - s$                            ▷ $x = \{\{F^n(0)\}\} = F^n(0) - m$
        if $x \le \beta$ then
            return $\frac{m}{n}$        ▷ Exact rotation number: Proposition 1.10 holds assuming that the rounding error of $F^n(0)$ is smaller than tol
        end if
    end for
    return $\frac{m+x}{\texttt{max\_iter}}$         ▷ We do not know whether we are in the assumptions of Proposition 1.10. So, we iteratively estimate the rotation number as in Algorithm 1. The error bound is given by Lemma 1.8
end procedure
```

is verified (that is, whether the assumptions of Proposition 1.10 are verified) before exhausting the `max_iter` iterates determined a priori.

In what follows $\widetilde{F^n(0)}$ will denote the computed value of $F^n(0)$ with rounding errors for $n = 1, 2, \dots, \texttt{max\_iter}$.

The algorithm goes as follows (see Algorithm 3 for a full implementation in pseudocode, and see the explanatory comments below):

(Alg. 3-1) Re-parametrize the lifting F so that it has a maximal (with respect to the inclusion relation) constant section of the form $[-\texttt{tol}, \beta + \texttt{tol}]$, where `tol` is the pre-defined rounding error bound.

(Alg. 3-2) Set the inputs of the algorithm:
- β as in step 1,
- F, the map from which we want the rotation number,
- `error`, the maximum error we want our approximation to have.

(Alg. 3-3) Decide the maximum number of iterates $\texttt{max\_iter} = \text{CEIL}\left(\frac{1}{\texttt{error}}\right)$ to perform in the worst case (i.e. when Proposition 1.10 does not work).

(Alg. 3-4) Initialize $x = 0$ and $m = 0$.

(Alg. 3-5) Compute iteratively $x = \{\{\widetilde{F^n(0)}\}\}$ and $m = \left\lfloor \widetilde{F^n(0)} \right\rfloor$ (so that $\widetilde{F^n(0)} = x + m$) for $n \le \texttt{max\_iter}$.

(Alg. 3-6) Check whether $x \le \beta$. On the affirmative we are in the assumptions of Proposition 1.10, and thus, $\rho_{_F} = \frac{m}{n}$. Then, the algorithm returns this value as the "exact" rotation number.

(Alg. 3-7) If we reach the maximum number of iterates (i.e. $n = \texttt{max\_iter}$) without being in the assumptions of Proposition 1.10 (i.e. with $x > \beta$ for every x) then, by Lemma 1.8, we have

$$\left|\rho_F - \frac{m+x}{\texttt{max\_iter}}\right| = \left|\rho_F - \frac{\widetilde{F^n(0)}}{\texttt{max\_iter}}\right| \approx \left|\rho_F - \frac{F^n(0)}{\texttt{max\_iter}}\right| < \frac{1}{\texttt{max\_iter}},$$

and the algorithm returns $\frac{m+x}{\texttt{max\_iter}}$ as an estimate of ρ_F with $\frac{1}{\texttt{max\_iter}}$ as the estimated error bound.

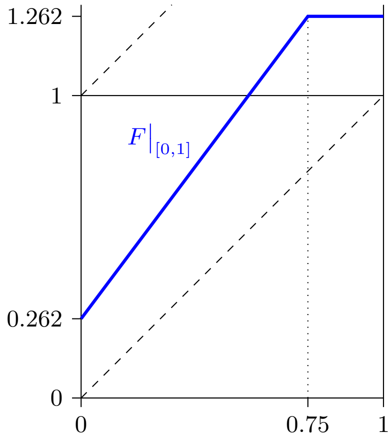
Remark 1.12. The fact that we can only *check whether the assumptions of Proposition 1.10 are verified before exhausting the* $\texttt{max\_iter} = \text{CEIL}\left(\frac{1}{\texttt{error}}\right)$ *iterates determined a priori* does not allow to take into account that F may have a lifted cycle intersecting the constant section but of very large period, i.e. with period larger than `max_iter`. In practice this problem is totally equivalent to the non-existence (or rather invisibility) of a lifted cycle intersecting the constant section, and it can be considered as a new (algorithmic) restriction to Proposition 1.10. It is solved in (Alg. 3-6) in the same manner as the two other problems related with the applicability of Proposition 1.10 that have already been discussed: by estimating the rotation number as in Algorithm 1. ∎

In the last part of this subsection we are going to discuss the rationale of (Alg. 3-2) (and, as a consequence of (Alg. 3-5)). The necessity of this tuning of the algorithm comes again from a challenge concerning the application of Proposition 1.10, which turns to be one of the most relevant restrictions in the use of that proposition. We will begin by discussing how we can efficiently check the condition $\xi = F^n(0) \in K + \mathbb{Z}$ (or equivalently $F^n(K) \cap (K+\mathbb{Z}) \neq \emptyset$) by taking into account that *the computation of* $F(x)$ *is done with rounding errors, and thus we do not know the exact values of* $F^n(0)$ *for* $n = 1, 2, \ldots,$ `max_iter`. The next example shows the problems arising in this situation.

Example 1.13. $\widetilde{F^n(0)} \in K + \mathbb{Z}$ *but* $F^n(K) \cap (K+\mathbb{Z}) = \emptyset$, *and this leads to a completely wrong estimate of* ρ_F.
Let $F \in \mathcal{L}_1$ be the map such that $F(x) = F\big|_{[0,1]}(\{\{x\}\}) + \lfloor x \rfloor$ for every $x \in \mathbb{R}$, and let

$$F\big|_{[0,1]}(x) := \begin{cases} \frac{4}{3}x + \mu & \text{if } x \in \left[0, \frac{3}{4}\right], \\ 1 + \mu & \text{if } x \in \left[\frac{3}{4}, 1\right], \end{cases}$$

with $\mu = \frac{819}{3124} - 10^{-16}$.



For this map F we have $K = \left[-\frac{3}{4}, 0\right]$ and (see Figure 1.5) the graph of F^5 lies above the graph of $x \longmapsto x+1$ and below the graph of $x \longmapsto x+2$, but very close to it at five F-preimages of $x = \frac{3}{4}$. On the other hand,

$$F^5(0) = 1.74999999999999887 \cdots \notin K + \mathbb{Z}$$

but the distance between $F^5(0)$ and $K + \mathbb{Z}$ is $\frac{7}{4} - F^5(0) \approx 1.138 \cdot 10^{-15}$. Should the computations be done with rounding errors of this last magnitude, we may have

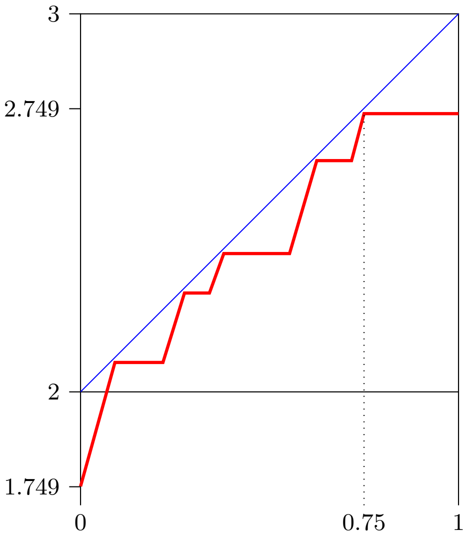


FIGURE 1.5. The graph of F^5. It lies below the graph of $x \longmapsto x{+}2$ but very close to it at five F-preimages of $x = \frac{3}{4}$.

$\widetilde{F^5(0)} \gtrapprox \frac{7}{4}$, and accept erroneously that $F^5(0) \in K + \mathbb{Z}$. This would lead to the conclusion that $\rho_{_F} = \frac{2}{5}$ but, as it can be checked numerically, $\rho_{_F} \approx 0.3983$ which is far from $\frac{2}{5}$.

At a first glance this seems to be paradoxical but, indeed, it can be viewed in the following way: The graph of F^5 does not intersect the diagonal (modulo 1) $x + 2$, but there is a map G close (at rounding errors distance) to F such that the graph of G^5 intersects that diagonal, and this gives a lifted periodic orbit of period 5 and rotation number $\frac{2}{5}$ for G. On the other hand, nothing is granted about the modulus of continuity of $\rho_{_F}$ as a function of F (notice that that everything here is continuous including the dependence of the rotation number of F on the parameter μ), and this example explicitly shows that it may be indeed very big. In short, close functions can have very different rotation numbers. ■

The most reasonable solution to the problem pointed out in the previous example consists in restricting the size of K depending of an a priori estimate of the rounding errors in computing $\widetilde{F^n(0)}$ for $n = 1, 2, \ldots, \texttt{max\_iter}$. Thus, we denote by `tol` an upper bound of these rounding errors, so that

$$\left| F^n(0) - \widetilde{F^n(0)} \right| \leq \texttt{tol} \qquad \text{holds for } n = 1, 2, \ldots, \texttt{max\_iter},$$

and, given a maximal (with respect to the inclusion relation) constant section K such that $0 \in K$ we write $K := [\alpha - \texttt{tol}, \beta + \texttt{tol}]$. Then observe that the condition $\widetilde{F^n(0)} \in [\alpha, \beta] + m$ for some $n \in \mathbb{N}$ and $m \in \mathbb{Z}$ implies $\xi = F^n(0) \in K + m$, and $\rho_{_F} = \frac{m}{n}$ by Proposition 1.10.

In practice, this "rounding errors free" version of the algorithm imposes a new restriction to the applicability of Proposition 1.10 (in the sense that it reduces even more the class of functions for which we can get the "exact rotation number"). However, as before, the rotation numbers of the maps in the assumptions of Proposition 1.10 for which we cannot compute the "exact rotation number" can be estimated as in Algorithm 1.

The computational efficiency of the algorithm strongly depends on how we check the condition $\widetilde{F^n(0)} \in K + \mathbb{Z}$. Taking into account the above considerations and improvements of the algorithm, this amounts checking whether $\alpha+\ell \leq \widetilde{F^n(0)} \leq \beta+\ell$ for some $\ell \in \mathbb{Z}$, and we have to do so by using $x = \{\{\widetilde{F^n(0)}\}\}$ and $m = \left\lfloor \widetilde{F^n(0)} \right\rfloor$ instead of $\widetilde{F^n(0)} = x+m$, which is the algorithmic available information. Checking whether $\alpha + \ell \leq \widetilde{F^n(0)} \leq \beta + \ell$ for some $\ell \in \mathbb{Z}$ is problematic since it requires at least two comparisons, and moreover in general $\ell \neq m$ (and thus we need some more computational effort to find the right value of ℓ). A very easy solution to this problem is to change the parametrization of F so that $\alpha = 0$. In this situation we have

$$m = m + \alpha \leq \widetilde{F^n(0)}, m + \beta < m + 1$$

because $\operatorname{diam}(K) < 1$, and $m = \left\lfloor \widetilde{F^n(0)} \right\rfloor$. Consequently, $\alpha + \ell \leq \widetilde{F^n(0)} \leq \beta + \ell$ for some $\ell \in \mathbb{Z}$ is equivalent to

$$\ell = m \quad \text{and} \quad x \leq \beta.$$

Thus, by "tuning" F so that $\alpha = 0$ we get that $\ell = m$ and we manage to determine whether $\widetilde{F^n(0)} \in [\alpha, \beta] + m$ just with one comparison ($x \leq \beta$).

To see how we can change the parametrization of F (that is step Alg. 3-1) so that $\alpha = 0$ consider the map $G(x) := F(x+\alpha) - \alpha$. Clearly, F and G are conjugate by the rotation of angle α: $x \longmapsto x+\alpha$. Then, obviously, G is a non-decreasing map in $\mathcal{L}_1$, has a constant section $[-\ \mathtt{tol}, \beta - \alpha +\ \mathtt{tol}]$, and $\rho_{_F} = \rho_{_G}$. So, every lifting can be replaced by one of its re-parametrizations with the same rotation number and constant section $[-\ \mathtt{tol}, \beta +\ \mathtt{tol}]$, where $\beta < 1 - 2\ \mathtt{tol}$.

1.4. Testing the Algorithm

In this section we will test the performance of Algorithm 3 by comparing it against Algorithms 1 and 2 when dealing with different usual computations concerning rotation intervals. First we will compare the efficiency of the three algorithms in computing and plotting Devil's Staircases. Afterwards we will plot rotation intervals and Arnold tongues for two bi-parametric families that mimic the standard map family. In the latter two cases, we will try to compare our algorithm with Algorithms 1 and 2 whenever possible.

1.4.1. Computing Devil's staircases. In this subsection we will perform the comparison of algorithms by computing and plotting the Devil's staircase for the parametric family $\{F_\mu\}_{\mu\in[0,1]} \subset \mathcal{L}_1$ defined as

Definition 1.14.

$$F_\mu(x) = F_\mu\big|_{[0,1]}\big(\{\{x\}\}\big) + \lfloor x \rfloor,$$

where (see Figure 1.1)

$$F_\mu\big|_{[0,1]}(x) = \begin{cases} \frac{4}{3}x + \mu & \text{if } x \leq \frac{3}{4} \\ \mu + 1 & \text{if } x > \frac{3}{4} \end{cases}. \tag{1.3}$$

■

Before doing this we shall remind the notion of a Devil's Staircase, and why typically exist for such families. To this end we will first recall and survey on the notion of *persistence of a rotation interval.*

Definition 1.15. Given a subclass $\mathcal{A}$ of $\mathcal{L}_1$, we say that $F \in \mathcal{A}$ has an $\mathcal{A}$-persistent rotation interval if there exists a neighbourhood U of F in $\mathcal{A}$ such that

$$\operatorname{Rot}(G) = \operatorname{Rot}(F)$$

for every $G \in U$. ■

TABLE 1.1. Performance of the three algorithms studied for a variety of problems. The cells marked with N/A in blue remark that Algorithm 2 does not work in general for $\rho \notin [0,1]$. The ones marked with N/A in red denote that the computation lasted more than a 100 processor hours and thus was terminated before it ended.

Problem	**Function Family**	**Time taken by algorithm (s)**		
		Classic	*Simó* et al.	*Proposed*
Devil's Staircase	F_μ (Def. 1.14)	2425.25	210.648	0.1413
Rotation Interval	Standard	354.868	N/A	3.2874
	PWLSM (Def. 1.20)	110.892	N/A	0.4737
	DSM (Def. 1.21)	63.588	N/A	0.2463
Arnol'd Tongues	Standard	N/A	N/A	14948.41
	PWLSM	N/A	N/A	9729.17
	DSM	N/A	N/A	4562.75

We can now state the *Persistence Theorem* (c.f. [**Mis89**]):

THEOREM 1.16 (Persistence Theorem). *Let $\mathcal{A}$ be a subclass of $\mathcal{L}_1$. Then the following statements hold:*

(a) The set of all maps with $\mathcal{A}$-persistent rotation interval is open and dense in $\mathcal{A}$ (in the topology of $\mathcal{A}$).
(b) If F has an $\mathcal{A}$-persistent rotation interval, then ρ_{F_l} and ρ_{F_u} are rational.

REMARK 1.17. If we apply Theorem 1.16 to our family $\{F_\mu\}_{\mu\in[0,1]}$ which verifies that the rotation number of F_μ exists for every $\mu \in [0,1]$, we have that the set of parameters $\mu \in [0,1]$ for which we have irrational rotation number has measure 0. Furthermore, for any $\kappa \in \mathbb{Q}$ such that there exists μ with $\rho_{F_\mu} = \kappa$, there exists an interval $[\alpha,\beta] \ni \mu$ such that for all $\eta \in [\alpha,\beta]$, $\rho_{F_\eta} = \kappa$. ■

The so-called Devil's staircase is the result of plotting the rotation number as a function of the parameter μ. By Theorem 1.16 we have that this plot will have constant sections for any rational rotation number, hence the "Staircase" in the name.

To test the algorithms, a μ-parametric grid computation of ρ_{F_μ} with μ ranging from 0 to 1 with a step of 10^{-5} has been done. For Algorithms 1 and 3 the `error` has been set to 10^{-6}. For Algorithm 3 the `tolerance` has been set to 10^{-10}. For Algorithm 2 we have arbitrarily set the number of iterates to 1000.

In Figure 1.6 we show a plot of the Devil's Staircase computed with Algorithm 3, and the plots of the differences between ρ_{F_μ} computed with Algorithms 3 and 1, and the differences between ρ_{F_μ} computed with Algorithms 3 and 2.

Table 1.1 shows the times[1] taken by each of the three algorithms in computing the whole Devil's staircase using the three algorithms studied.

We remark that in the computation of the Devil's Staircase, Algorithm 3 has been reduced to Algorithm 1 only for $\mu = 0$ and for $\mu = 1$, as one would expect, since these cases follow the pattern of Example 1.13.

As a part of the testing of the algorithms we have also considered the inverse problem: *Given a value $\rho \in \mathbb{R}\backslash\mathbb{Q}$ and a tolerance $\varepsilon > 0$ find the value $\mu = \mu(x)$ such that $\rho_{F_\mu} \in [\rho - \varepsilon, \rho + \varepsilon]$*. This problem has turned to be extremely ill-conditioned: by choosing ρ to be any irrational number. We have tried to use algebraic numbers

[1] The simulations have been done with an Intel® Core™ i7-3770 CPU @3.4GHz.

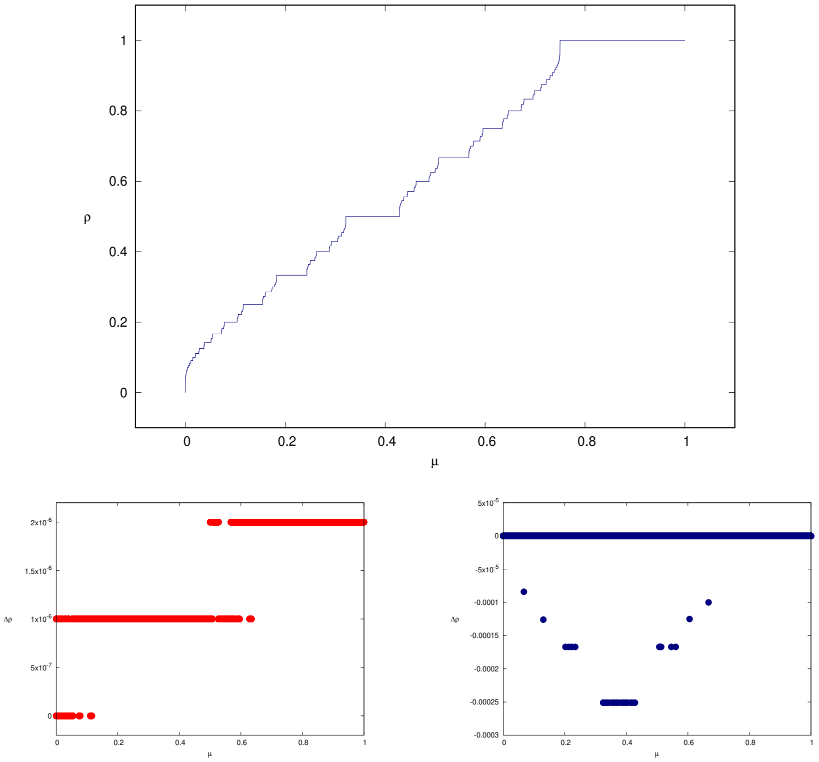


FIGURE 1.6. The Devil's Staircase associated to the family (1.3) computed with Algorithm 3 (upper picture). The lower pictures show the plots of the differences between the value of ρ_{F_μ} computed with Algorithm 3 and the value of ρ_{F_μ} computed with Algorithm 1 (left picture), and with the value of ρ_{F_μ} computed with Algorithm 2 (right picture).

such as the golden mean or $1/\sqrt{2}$ and transcendental numbers such as $\pi/4$ or $e/3$. In any studied case, the continuity module of the function $\mu \mapsto \rho_{F_\mu}$ around $\mu(\rho)$ was estimated to be at least 10^{25}, making any attempt to solve the problem unfeasible. Note that this is coherent with Theorem 1.16, since the values of μ that give non-rational values are nowhere dense.

1.4.2. Rotation intervals for standard-like maps. In this subsection we test our algorithm by efficiently computing the rotation intervals and some *Arnol'd tongues* for three bi-parametric families of maps: the standard map family and two piecewise-linear extensions of it; one continuous but not differentiable, and another one which is not even continuous.

We emphasize that the usual algorithms such as the ones from [**BS98, Pav95, SV06, Vel88**] cannot be used for these last two families while the one we propose here it works very well indeed.

First we will recall the notion of *Arnol'd tongue*.

DEFINITION 1.18 (Arnol'd Tongue [**Boy86**]). Let $\{F_{a,b}\}_{(a,b)\in\mathsf{P}}$ be a biparametric family of maps in $\mathcal{L}_1$ for which the rotation interval $\operatorname{Rot}(F_{a,b})$ is well defined for every possible point $(a,b) \in \mathsf{P}$ in the parameter set. Given a point $\varrho \in \mathbb{R}$ we define the *ϱ–Arnold Tongue of* $\{F_{a,b}\}_{(a,b)\in\mathsf{P}}$ as

$$\mathcal{T}_\varrho = \{(a,b) \in \mathsf{P} : \varrho \in \operatorname{Rot}(F_{a,b})\} \subset \mathsf{P}.$$

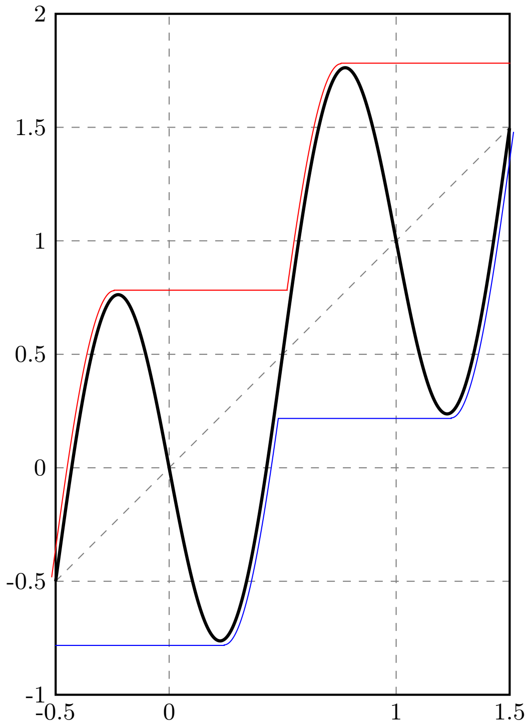


FIGURE 1.7. The standard map with $a = 2\pi$ and $\Omega = 0$, with its lower map in blue and its upper map in red.

■

Next we introduce each of the three families that we study and, for each of them we show the results and we explain the performance of the algorithm.

DEFINITION 1.19 (Standard Map). $\mathsf{S}_{\Omega,a} \in \mathcal{L}_1$ is defined as (see Figure 1.7):

$$\mathsf{S}_{\Omega,a}(x) := x + \Omega - \frac{a}{2\pi}\sin(2\pi x). \tag{1.4}$$

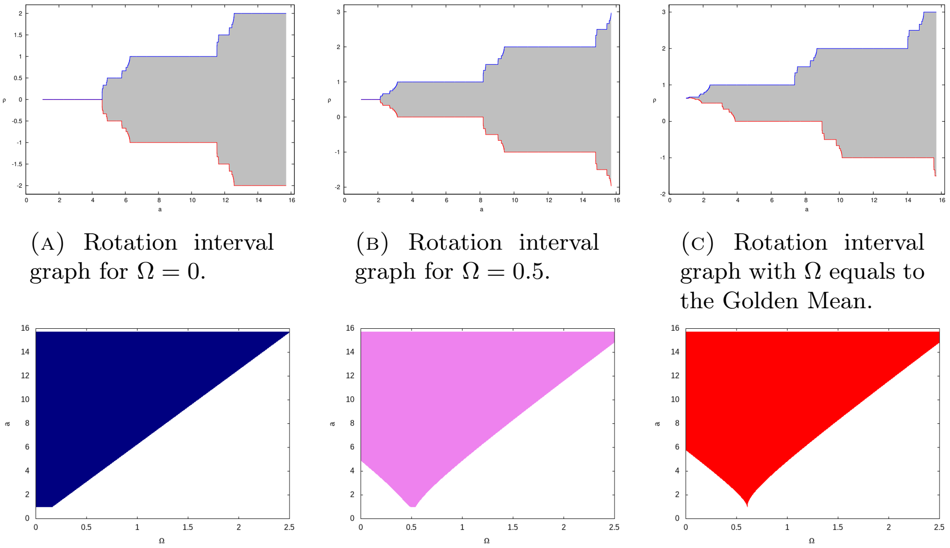
■

To compute the rotation intervals of $\mathsf{S}_{\Omega,a}$ we will use Theorem 1.5, together with Algorithm 3. To this end, first we will compute $\left(\mathsf{S}_{\Omega,a}\right)_l$ and $\left(\mathsf{S}_{\Omega,a}\right)_u$ (that is, the lower and upper maps of $\mathsf{S}_{\Omega,a}$), and then we will use Algorithm 3 to compute the rotation numbers $\rho_{(\mathsf{S}_{\Omega,a})_l}$ and $\rho_{(\mathsf{S}_{\Omega,a})_u}$ of these maps.

Note that $\mathsf{S}_{\Omega,a}$ is non-invertible for $a > 1$. Hence, in this case, $\left(\mathsf{S}_{\Omega,a}\right)_l$ and $\left(\mathsf{S}_{\Omega,a}\right)_u$ do not coincide and have constant sections. However, the characterization of these constants sections is not straightforward, since their endpoints have to be computed numerically. This is the reason why the computations of the rotation intervals and Arnol'd tongues for the standard map have been the slowest ones.

In Figure 1.8 we show some graphs of the rotation interval and Arnol'd tongues for the Standard Map. The graphs of the rotation intervals are plotted for three different values of Ω as a function of the parameter a.

DEFINITION 1.20 (Piecewise-linear standard map). We start by defining a convenience map



(A) Rotation interval graph for $\Omega = 0$.

(B) Rotation interval graph for $\Omega = 0.5$.

(C) Rotation interval graph with Ω equals to the Golden Mean.

(D) 0−Arnol'd tongue.

(E) $\mathcal{T}_\varrho$ Arnol'd tongue for $\varrho = 0.5$.

(F) $\mathcal{T}_\varrho$ for ϱ equal to the Golden Mean.

FIGURE 1.8. Graphs of the rotation interval and Arnol'd tongues for the Standard Map $\mathsf{S}_{\Omega,a}$. The graphs of the rotation intervals are plotted as a function of the parameter a.

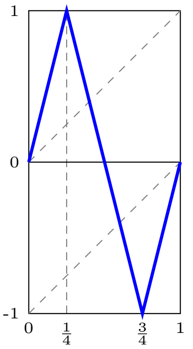
$\tau\colon [0,1] \longrightarrow [-1,1]$ as follows:

$$\tau(x) = \begin{cases} 4x & \text{when } x \in \left[0, \frac{1}{4}\right], \\ 2-4x & \text{when } x \in \left[\frac{1}{4}, \frac{3}{4}\right], \text{ and} \\ 4(x-1) & \text{when } x \in \left[\frac{3}{4}, 1\right]. \end{cases} \tag{1.5}$$

Then, the *piecewise-linear standard map* $\mathsf{T}_{\Omega,a} \in \mathcal{L}_1$ is defined by (see Figure 1.9):

$$\mathsf{T}_{\Omega,a}(x) = x + \Omega - \frac{a}{2\pi}\tau\big(\{\!\{x\}\!\}\big), \tag{1.6}$$

which corresponds to the standard map but using the τ wave function instead of the $\sin(2\pi x)$ function. ■



The upper and lower maps for this family are very easy to compute. Moreover, $\mathsf{T}_{\Omega,a}$ is non-increasing for $a > \frac{\pi}{2}$ and hence, in this case, the upper and lower maps do not coincide and have constant sections.

To compute the rotation intervals and Arnol'd Tongues of $\mathsf{T}_{\Omega,a}$ we proceed as for the Standard Map by using Theorem 1.5 and Algorithm 3.

In Figure 1.10 we show some graphs of the rotation interval and Arnol'd tongues for the piecewise-linear standard map. The graphs of the rotation intervals are plotted for three different values of Ω as a function of the parameter a.

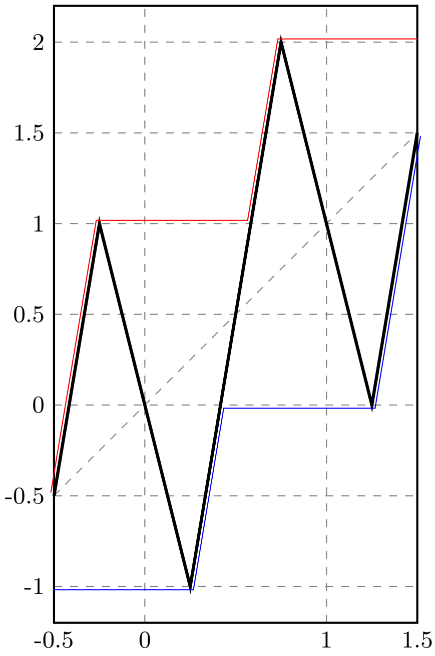
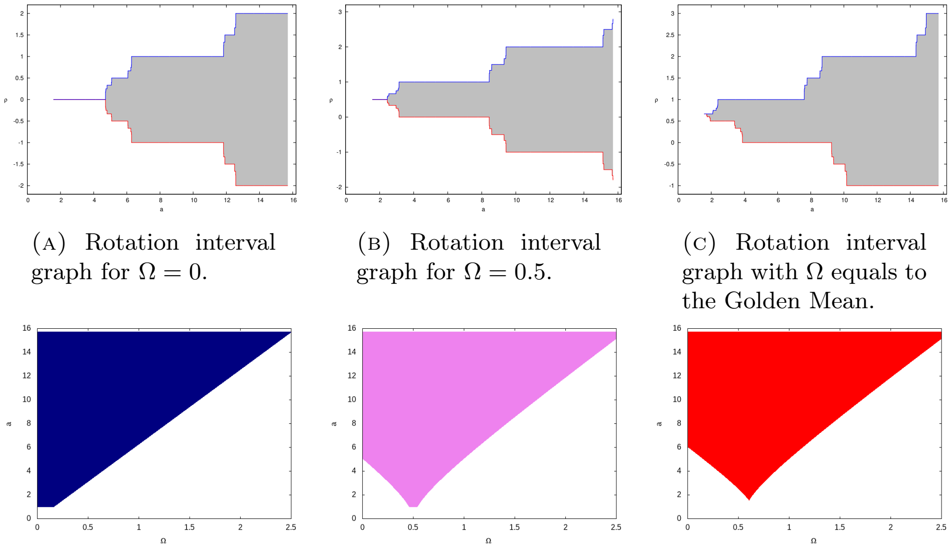


FIGURE 1.9. The piecewise-linear standard map $\mathsf{T}_{\Omega,a}$ with $a = \frac{5\pi}{2}$ and $\Omega = 0$. The lower map of $\mathsf{T}_{\Omega,a}$ is drawn in blue, and the upper map in red.



(A) Rotation interval graph for $\Omega = 0$.

(B) Rotation interval graph for $\Omega = 0.5$.

(C) Rotation interval graph with Ω equals to the Golden Mean.

(D) 0−Arnol'd tongue.

(E) $\mathcal{T}_{\varrho}$ Arnol'd tongue for $\varrho = 0.5$.

(F) $\mathcal{T}_{\varrho}$ for ϱ equal to the Golden Mean.

FIGURE 1.10. Graphs of the rotation interval and Arnol'd tongues for the piecewise-linear standard map $\mathsf{T}_{\Omega,a}$. The graphs of the rotation intervals are plotted as a function of the parameter a.

Definition 1.21 (The Discontinuous Standard Map). $\mathsf{D}_{\Omega,a} \in \mathcal{L}_1$ is defined as (see Figure 1.11):

$$\mathsf{D}_{\Omega,a}(x) := x + \Omega + \frac{a}{2\pi}\{\!\{x\}\!\}. \tag{1.7}$$

∎

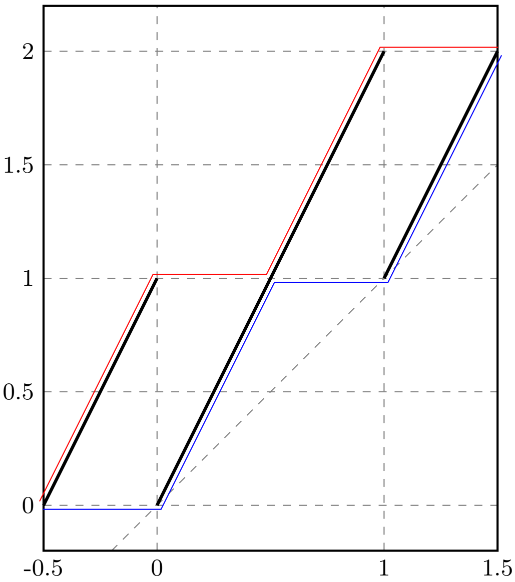


Figure 1.11. The discontinuous standard map with $a = 2\pi$ and $\Omega = 0$ with its lower map in blue and its upper map in red.

The map $\mathsf{D}_{\Omega,a}$, being discontinuous, belongs to the so called class of *old heavy maps* [**Mis86**] (the *old* part of the name stands for *degree one lifting* — that is, $\mathsf{D}_{\Omega,a} \in \mathcal{L}_1$). A map $F \in \mathcal{L}_1$ is called *heavy* if for any $x \in \mathbb{R}$,

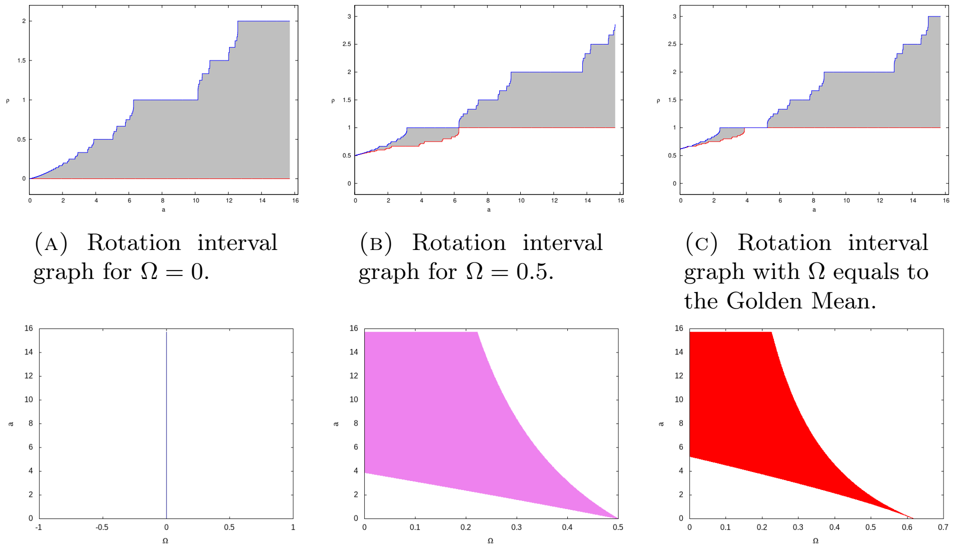
$$\lim_{y \searrow x^+} F(y) \le F(x) \le \lim_{y \nearrow x^-} F(y)$$

(in other words, the map "falls down" at all discontinuities).

Observe that for the class of *old heavy maps* the upper and lower maps in the sense of Definition 1.4 are well defined and continuous. Moreover, the whole family of *water functions* (c.f. [**ALM00**]) is well defined and continuous. So, the rotation interval of the *old heavy maps* is well defined [**Mis86**, Theorem A] and, moreover, Theorem 1.5 together with Algorithm 3 work for this class. Hence, to compute the rotation intervals and Arnol'd Tongues of $\mathsf{D}_{\Omega,a}$ we proceed again as for the Standard Map.

As for the piecewise-linear standard maps the upper and lower maps are very easy to compute, and have constant sections for $a \neq 0$.

In Figure 1.12 we show some graphs of the rotation interval and Arnol'd tongues for the discontinuous standard map. The graphs of the rotation intervals are plotted for three different values of Ω as a function of the parameter a.



(A) Rotation interval graph for $\Omega = 0$.

(B) Rotation interval graph for $\Omega = 0.5$.

(C) Rotation interval graph with Ω equals to the Golden Mean.

(D) $0-$Arnol'd tongue.

(E) $\mathcal{T}_\varrho$ Arnol'd tongue for $\varrho = 0.5$.

(F) $\mathcal{T}_\varrho$ for ϱ equal to the Golden Mean.

FIGURE 1.12. Graphs of the rotation interval and Arnol'd tongues for the discontinuous standard map $\mathsf{D}_{\Omega,a}$. The graphs of the rotation intervals are plotted as a function of the parameter a.

The times taken for all the computation related with the rotation intervals and the Arnol'd Tongues for each of the families studied using Algorithms 1, 2 and 3 can be found in Table 1.1.

1.5. Conclusions

The proposed algorithm clearly outperforms all the other tested algorithms, both in precision and speed even though the "exact" (and quick) part of the algorithm does not work for all the non-decreasing liftings in $\mathcal{L}_1$ which have a constant section (and hence the rotation number of these "bad" cases has to be computed with the much more inefficient classical algorithm). For all natural examples for which it has been tested, the computational speed and precision were unparalleled. Moreover, the set of functions becomes very general when one considers the fact that the upper and lower functions inherently have constant sections for any F that is not strictly increasing. Hence, the algorithm becomes a crucial tool to compute rotation intervals for general functions in $\mathcal{L}_1$ and, therefore, to find the set of periods of such maps [**ALM00**].

Moreover, a deeper study has been done on the dependence of the rotation number on the parameters. Our preliminary results have found that for irrational rotation numbers, the dependence of the parameters around them is extremely sensitive, with continuity module being at least 10^{25}. This agrees with Theorem 1.16, which says that non-persistent functions have measure zero.

Part 2

Computing regularities using Daubechies Wavelets

Introduction

As mentioned in the Preface, this part is the longer half of the thesis. It is an extension of the research started by Alsedà, Mondelo and Romero in [**AMR16**], and was later expanded in David Romero's PhD thesis [**RiS15**][2]. We will devote ourselves to the study of attractors on quasi-periodically forced skew products. That is, systems of the form

$$\begin{array}{rccc} \mathfrak{F}_{\sigma,\varepsilon}: & \mathbb{S}^1 \times \mathbb{R} & \longrightarrow & \mathbb{S}^1 \times \mathbb{R} \\ & (\theta, x) & \longmapsto & (R_\omega(\theta), F_{\sigma,\varepsilon}(\theta, x)), \end{array}$$

where $\omega \in \mathbb{R}\backslash\mathbb{Q}$ and $R_\omega(\theta) = \theta + \omega \mod 1$. Note that $\mathfrak{F}$ has a dependence on σ and ε, which can be used to study bifurcations within the system. One particular case is the change from a regular nonchaotic attractor to an strange one. What we call *Strange Non-chaotic attractors* (or SNAs) [**AC09**] are, with some nuance, exactly what they name implies, attractors of a nonchaotic system that despite having regular dynamics have a very complicated geometry. In particular, we would like to obtain a semi-analytic method that allows us to deal with the *strangeness* of these attractors. We would like to remark the semi-analicity of the method, since for SNAs there exist very few analytical results proving their existence. In fact, most of the current proofs of existence are done for a particular system [**GOPY84**] or for very limited families of systems [**Kel96, AM08, Bje09**]. This stems mainly from the fact that these systems are extremely different from one another. Moreover, there exist systems in which the strangeness is suspected but not proven, mainly because the approximation to the problem has been purely numerical [**Kan84, NK96, Har05**].

To this end, we will seek to obtain the *truncated wavelet expansion* of an attractor (for example [**AM08, Kel96, NK96**]) using a periodized Daubechies wavelet basis. That is, we want to approximate an attractor φ using a periodized wavelet basis of $\mathscr{L}^1(\mathbb{S}^1)$ $\psi^{\text{PER}}_{-j,n}$:

$$\varphi \approx d_0 + \sum_{j=0}^{N} \sum_{n=0}^{2^j-1} d_{j,n} \psi^{\text{PER}}_{-j,n}.$$

The reasons we might be interested in this expansion are manyfold. On one hand, in modern analysis, wavelet basis are considered building blocks of functions, allowing us to obtain information about the behaviour of φ via inclusion to various functional spaces [**Tri92, Tri06, Tri10, Jaf91**]. In particular, we hope to obtain information on the *strangeness* of φ through it belonging to some regularity space. We are well aware that this measure is not perfect or complete, but it serves as a good measure to understand whether one can expect the geometry to be complicated or not. Another reason to use wavelets is that they are very well suited for approximating functions with very complicated geometry, which is the case for SNAs.

[2] [**AMR16**] has a later publishing date due to the time it takes to publish a peer-review paper

In fact, the two reasons mention above answer the natural question that of why the need for such a complicated basis of $\mathscr{L}^1(\mathbb{S}^1)$ and not use the Fourier basis. For example, some of the attractors we will study are upper semi-continuous functions that are discontinuous on a dense set. In this situation, wavelets truly shine, since they are much more adaptable to sudden jumps in the functions they are approximating, unlike the sine or the cosine.

In Chapter 2, there is an introduction to the dynamical problem, showing some interesting examples of (suspected) strange non-chaotic attractors and the general dynamical tools we are going to use through the rest of the chapters. In particular we will give the definition of the invariance equation and give a method to compute the vertical Lyapunov exponent of hyperbolic attractors in quasi-periodically forced skew products.

In Chapter 3 we introduce the general theory of wavelets, sketching the obtention of the Daubechies wavelet family. Moreover, we will show how the wavelet expansion of a function might allow us to compute its regularity (or lack thereof in the case of strangeness). However, through some examples we will show the limitations of this framework as a measure for strangeness.

In Chapter 4 we use the results from the previous chapters to find a general numerical method that allows us to obtain truncated wavelet expansions of attractors of quasi-periodically forced skew products. The core of this chapter is the matricial form of the invariance equation, in which *the Wavelet Matrices* are used to simplify the calculations. The chapter ends with the application of the method to the systems presented in Chapter 2. The ones for which there exist analytical results are used as a proof of concept. Finally, we give some results for the Nishikawa-Kaneko system [**NK96**] that we believe might shed some light to the behaviour of its attractor.

Finally, in Chapter 5 we give a comprehensive explanation of how we have been able to generate the Wavelet Matrices from the previous chapter. In this sense we have been able to obtain very efficient computational implementations through a smart use of the Daubechies-Lagarias algorithm [**DL91, DL92**]. This is probably the most technical part of the thesis, since no amount of detail is spared.

CHAPTER 2

Setting up the Dynamical Problem

This chapter serves as an introduction to the dynamical problem we are interested in, *strange non-chaotic attractors* (SNAs) in *quasi-periodically forced skew products*. In particular, we will start by giving a formal definition of strange non-chaotic attractor, followed by some of the examples that we will study in this thesis. Furthermore, we will present some of the existing analytical results we know. Moreover, we will introduce the crucial notion of the invariance equation. The goal of this equation is to rephrase the existence of an invariant object of a Dynamical system in terms of an easy to use functional equation. Finally, we will give an algorithm to compute the Lyapunov exponent of the maps we are interested in.

2.1. Strange non-chaotic attractors

Strange non-chaotic attractors or SNAs first appeared as an object of study in 1984 in a paper by Grebogi, Ott, Pelikan and Yorke (or the GOPY paper)[**GOPY84**]. The same year Kaneko [**Kan84**] published his first fractalization of torus paper, in which he gave inconclusive numerical evidence of the existence of an SNA, even though he labeled the object *a fractal*. Some time later him and Nishikawa [**NK96**] would return to this problem with some more tools and understanding of SNAs, but the results are still considered inconclusive. Since then, SNAs have become an object of interest in the field of dynamical systems, with many (mostly numerical) papers appearing on the topic [**Kan84, NK96, Har05**]

The name of this objects is self describing, however each of the parts need some considerations. First of all there is *strangeness*, which is a reference to the geometry of the attractor. This is probably the hardest to define. The first instance of strangeness where reported by Ruelle and Takens in [**RT71**], where they detected chaotic attractors with very complicated geometry. In particular these attractors where locally the product a Cantor set and a two dimensional manifold. In 1984 two papers considering (possible) SNAs appear. In [**GOPY84**] strangeness is defined as "not being a finite set of points nor piecewise differentiable". On the other hand in [**Kan84**] (and later in [**NK96**]), the strangeness is argued in terms of the (perceived) fractal geometry of the object. Finally, the strangeness has also been argued in terms of the Hausdorff dimension of the attractor [**AC09**]. Through this thesis we will consider the first definition, since the other two imply the first one.

The second part of the name refers to the *nonchaoticity*, which still needs to be well defined, since multiple definitions of chaos and chaoticity exist. However, in the case of the study of SNAs, the standard is to consider systems in which almost all the orbits that tend to the attractor (i.e. orbits that have the attractor as the ω-limit) have non-positive Lyapunov exponents [**GOPY84**].

The final bit of the name to be precisely defined is the fact that we have an *attractor*. When it comes to attractors, the seminal paper is the one by Milnor in 1985 [**Mil85**]. Hence, our definitions will stem from there almost verbatim. In particular we will consider that all our SNA are minimal attractors.

Thus, following the definition given by Alsedà and Costa [**AC09**], we can finally give a formal definition of an SNA.

Definition 2.1 (Strange Non-Chaotic Attractor [**AC09**]). Let $F : U \to U$ define a discrete dynamical system. A closed set $\mathcal{A} \subset U$ is *Strange Nonchaotic Attractor* if:

(a) $\mathcal{A}$ is a *minimal attractor* in the sense of Milnor [**Mil85**]: the *realm of attraction* $\rho(\mathcal{A}) := \{x \in U : \omega(x) \subset \mathcal{A}\}$ has positive Lebesgue measure $\mu(\mathcal{A})$ and there is no strictly smaller closed set $\mathcal{A}' \subsetneq \mathcal{A}$ such that $\mu\big(\rho(\mathcal{A}')\big) > 0$

(b) $\mathcal{A}$ is *strange*, that is $\mathcal{A}$ is not a finite set of points nor a piecewise differentiable manifold.

(c) the set of points in the realm of attraction of $\mathcal{A}$, $\rho(\mathcal{A})$, whose maximal upper Lyapunov exponent $\lambda_{\max}(x) > 0$ has zero Lebesgue measure (*non chaoticity*).

■

As hinted in the paragraphs leading to this definition, this is not the only definition one can find of a strange non-chaotic attractor. However we feel it is the most general that still only includes interesting objects.

Both the attractors studied in the GOPY paper and by Nishikawa and Kaneko, arise in bi-parametric families of *Quasi-Periodically Forced Skew Products*, which are maps of the form

$$\begin{array}{rccc} \mathfrak{F}_{\sigma,\varepsilon}: & \mathbb{S}^1 \times \mathbb{R} & \longrightarrow & \mathbb{S}^1 \times \mathbb{R} \\ & (\theta, x) & \longmapsto & (R_\omega(\theta), F_{\sigma,\varepsilon}(\theta, x)), \end{array} \tag{2.1}$$

where $R_\omega(\theta) = \theta + \omega \mod 1$, with $\omega \in \mathbb{R}\backslash\mathbb{Q}$ is an irrational rotation, and $F_{\sigma,\varepsilon} : \mathbb{S}^1 \times \mathbb{R} \to \mathbb{R}$ with $\sigma, \varepsilon \in \mathbb{R}^+$. A priory we do not impose any conditions on $F_{\sigma,\varepsilon}$. However, depending on each system some restrictions might apply.

For this types of systems Stark [**Sta97**] proved that whenever we have an attractor $\mathcal{A}$, it is always the graph of a multivalued function

$$\varphi : \mathbb{S}^1 \to \mathbb{R} \tag{2.2}$$

Hence, for us Definition 2.1(c) can be rewritten as:

$$\varphi(\theta) : \mathbb{S}^1 \to \mathbb{R} \text{ is not a piecewise differentiable map.}$$

In fact, one of the main goals of this thesis is to obtain the wavelet expansion of φ when it is (the closure of) a map. From this wavelet expansion one can infer some information about the strangeness of φ.

2.1.1. The invariance equation.

The invariance equation will be the main tool we will be using to obtain the wavelet expansion of $\varphi(\theta)$. It stems from rephrasing the existence of an invariant object (in particular an attractor) as finding a fixed point of a functional equation. To this end, we will use the *Transfer Operator* (see e.g [**AM08, AM15, Kel96**]) of system (2.1) to obtain the invariance equation. Let $\mathscr{P}$ be the space of all functions (not necessarily continuous) from $\mathbb{S}^1$ to $\mathbb{R}$. Then for $\varphi \in \mathscr{P}$ one can define the following functional operator $\mathbf{T}\colon \mathscr{P} \longrightarrow \mathscr{P}$, called *Transfer Operator*, as

$$\mathbf{T}(\varphi)(\theta) = F_{\sigma,\varepsilon}(R_\omega^{-1}(\theta), \varphi(R_\omega^{-1}(\theta))).$$

Where R_ω and $F_{\sigma,\varepsilon}$ are as in Equation (2.1). See Figure 2.1 for a more visual definition.

Remark 2.2. Observe that the functional version of the system (2.1) in $\mathscr{P}$ is the one defined by the Transfer Operator. Thus, inferred from Figure 2.1, iterates of $\mathfrak{F}_{\sigma,\varepsilon}$ correspond to iterating $\mathbf{T}$ at a functional level. Hence,

$$\mathbf{T}(\varphi)(\theta) = \pi_x \left(\mathfrak{F}_{\sigma,\varepsilon}(R_\omega^{-1}(\theta), \varphi(R_\omega^{-1}(\theta)))\right),$$

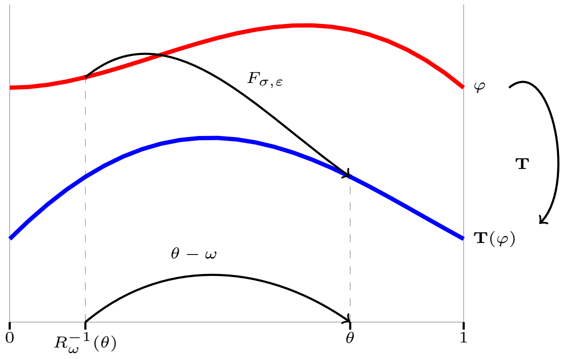


FIGURE 2.1. A sketch of the notion of the Transfer Operator.

where $\pi_x\colon \mathbb{S}^1 \times \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ denotes the projection with respect to the second component. ∎

In view of the above Remark, the graph of a function $\varphi\colon \mathbb{S}^1 \longrightarrow \mathbb{R}$ is invariant for the System (2.1) if and only if φ is a fixed *point* of $\mathbf{T}$, that is,

$$F_{\sigma,\varepsilon}\Big(R_\omega^{-1}(\theta), \varphi\big(R_\omega^{-1}(\theta)\big)\Big) = \mathbf{T}(\varphi)(\theta) = \varphi(\theta).$$

The above expression is, precisely, the invariance equation and will be our main tool to obtain a truncated wavelet expansion of φ. However, throughout the rest of this thesis we will use a trivial rewrite of the equation that will ease our computations:

$$F_{\sigma,\varepsilon}\big(\theta, \varphi(\theta)\big) = \varphi\big(R_\omega(\theta)\big). \tag{2.3}$$

REMARK 2.3. The Transfer Operator is a differentiable operator provided that $F_{\sigma,\varepsilon} \in C^{n+1}(\mathbb{S}^1 \times \mathbb{R}, \mathbb{R})$, where $\mathcal{C}^{n+1}(\mathbb{S}^1 \times \mathbb{R}, \mathbb{R})$ is the space of continuous and $n+1$ times differentiable functions (see e.g. [**dlLO99**]). ∎

2.1.2. Some examples of (possible) SNAs. In this subsection we will present some examples of (alleged) Strange Non-chaotic attractors. The existence of an SNA has been analytically proven for the first three examples. The fourth one is still undecided, with numerical estimations suggesting contradictory information [**NK96, Har05**].

All the systems studied add some restrictions to System (2.1). In particular, there are two main simplifications that are almost universally used. They concern simplifications on $F_{\sigma,\varepsilon}$ from Equation (2.1):

- *Multiplicative forcing:* $F_{\sigma,\varepsilon} := f_\sigma(x) \cdot g_\varepsilon(\theta)$
- *Additive forcing:* $F_{\sigma,\varepsilon} := f_\sigma(x) + g_\varepsilon(\theta)$

The former is the most studied starting with [**GOPY84**]. Some general analytical results actually exist for the multiplicative forcing. Most remarkably by Keller [**Kel96**] and by Alsedà and Misiurewicz [**AM08, AM15**]. Summarizing these two results is the main goal of the next subsection.

The systems that we will introduce in this subsection fall in one of these two categories. The first three correspond to systems with a multiplicative forcing, while the last one has an additive forcing. In this thesis we will use the Keller and the Alsedà-Misiurewicz systems to test the robustness of our method in order to apply it to the Nishikawa-Kaneko system. In all cases, $\omega = \frac{\sqrt{5}-1}{2}$, i.e., the rotation is given by the golden mean.

We are well aware that there exist more results regarding the existence of SNAs. When it comes to analytic results, the results by Bjerklöv [**Bje09**] are remarkable. He is able to prove the existence of SNAs for a general family with multiplicative forcing. Regarding numerical results, we would like to emphasize the results By Heagy and Hammel in [**HH94**], which through very thorough semi-analytical methods give mechanisms of creation of SNAs. However, for the sake of brevity we have decided to focus only on the following systems.

2.1.2.1. *The GOPY model* [**GOPY84**]. What we call the GOPY model is the first of two proven SNA published in the literature. In the paper the authors show the existence of two Strange Nonchaotic attractors, one on the cylinder and another one on a three dimensional space. We will focus on the former. The system is as follows:

$$\begin{array}{rccc} \mathfrak{F}_\sigma : & \mathbb{S}^1 \times \mathbb{R} & \longrightarrow & \mathbb{S}^1 \times \mathbb{R} \\ & (\theta, x) & \longmapsto & \big(R_\omega(\theta), 2\sigma \tanh(x)\cos(2\pi\theta)\big). \end{array} \tag{2.4}$$

In this case, $F_\sigma = f_\sigma(x) \cdot g(\theta)$, where $f_\sigma = 2\sigma\tanh(x)$ and $g(\theta) = \cos(2\pi\theta)$. The system is imagined as a Quasi-Periodically Forced Skew Product, its quasi-periodicity stemming from it being a nonlinear oscillator driven by two incommensurate frequencies.

This system has some properties that are very easy to check. First of all, since $f(0,\theta) = 0$, the circle $x \equiv 0$ is invariant. Moreover, the orbits are always bounded, because $|2\sigma\tanh(x)| < 2\sigma$ and $|\cos(2\pi\theta)| \leq 1$, hence $F_\sigma(x,\theta) < 2\sigma$. Therefore, irrespective of the initial condition, after one iterate the orbit of any point will fall within the band $x \in [-2\sigma, 2\sigma]$. Finally, it is easy to see that the vertical Lyapunov exponent for $x \equiv 0$ is exactly $\log\big(|\sigma|\big)$. Hence, if $\sigma > 1$, $x \equiv 0$ is repelling.

The existence of the attractor follows precisely from these last two facts.

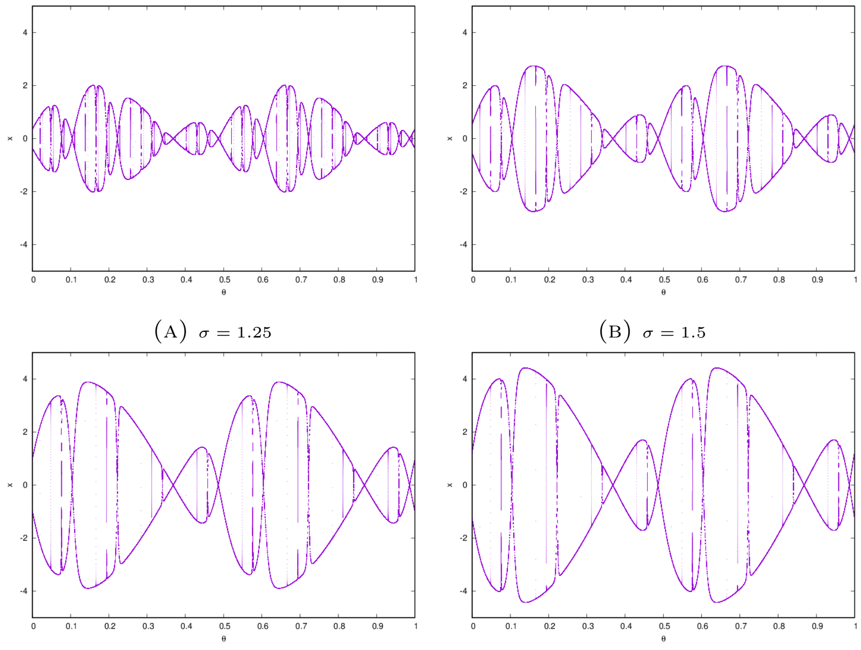
Its nonchaoticity is proven by seeing that the Lyapunov exponent on the attractor is negative by doing some clever bounds on its value.

Finally, its strangeness is shown using a simple analysis of the system. It can be seen that the attractor must be zero on a dense set of points whilst being different from zero on another dense set. That is, since $\cos(2\pi\theta) = 0$ for $\theta = \frac{1}{4}$ and $\theta = \frac{3}{4}$, $f\left(\,\cdot\,, \frac{1}{4}\right) = 0$ (same for $f\left(\,\cdot\,, \frac{3}{4}\right) = 0$). Therefore, since $x \equiv 0$ is invariant it follows that $F_{\sigma,\varepsilon}\Big(\,\cdot\,, R_\omega^k\left(\frac{1}{4}\right)\Big) = 0$ for any $k \in \mathbb{Z}$ (same for $\theta = \frac{1}{4}$). Hence, one can see that the attractor is equal to zero on a dense subset of $\mathbb{S}^1$. On the other hand, it is easy to see that the attractor is different from zero in a dense set of $\mathbb{S}^1$, so the strangeness follows. In Figure 2.2 one can see plots of the strange attractor for different values of σ.

2.1.2.2. *The Keller-GOPY model.* What we call the Keller-GOPY model is a particular example of the families of systems that Keller studies in his paper [**Kel96**]:

$$\begin{array}{rccc} \mathfrak{F}_{\sigma,\varepsilon} : & \mathbb{S}^1 \times \mathbb{R} & \longrightarrow & \mathbb{S}^1 \times \mathbb{R} \\ & (\theta, x) & \longmapsto & \big(R_\omega(\theta), 2\sigma \tanh(x)\,|\varepsilon + \cos(2\pi\theta)|\,\big). \end{array} \tag{2.5}$$

In particular, the model studied is very similar to the GOPY model, but it forces the cosine to be positive and the parameter ε guarantees that $g_\varepsilon(\theta) = 0$ for some θ if an only if $\varepsilon = 0$. In this way we ensure that what happened in the GOPY model, where the fact that $\cos(2\pi\theta) = 0$ for $\theta = \frac{1}{4}$ and $\frac{3}{4}$ implies strangeness, can be controlled by the value of ε. Therefore, one would obtain an Strange attractor provided $\varepsilon = 0$. In this case we say that the attractor is *pinched.* In [**Kel96**], the author gives some general results for a whole class of quasi-periodically forced skew products with multiplicative forcing. Mostly the conditions on these systems is GOPY-like, that is, f_σ must be $\mathcal{C}^1$, monotonous with $f_\sigma(0) = 0$ and $g_\varepsilon(\theta) = 0$ if and only if $\varepsilon = 0$. Some cases of the attractor for this system can be found in Figure 2.3



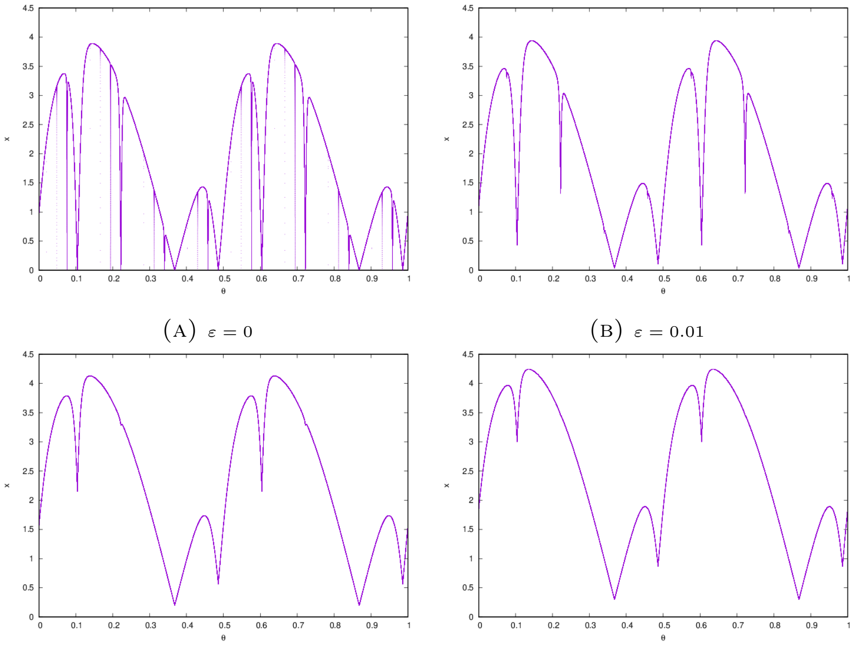
(A) $\sigma = 1.25$

(B) $\sigma = 1.5$

(C) $\sigma = 2$

(D) $\sigma = 2.25$

FIGURE 2.2. Plot of the attractor for the GOPY model for various values of σ.



(A) $\varepsilon = 0$

(B) $\varepsilon = 0.01$

(C) $\varepsilon = 0.1$

(D) $\varepsilon = 0.25$

FIGURE 2.3. Plot of the attractor for the Keller-GOPY model for $\sigma = 2$ and different values of ε. A simple visual inspection already shows the complexity of the pinched case compared to the regular case with $\varepsilon > 0$

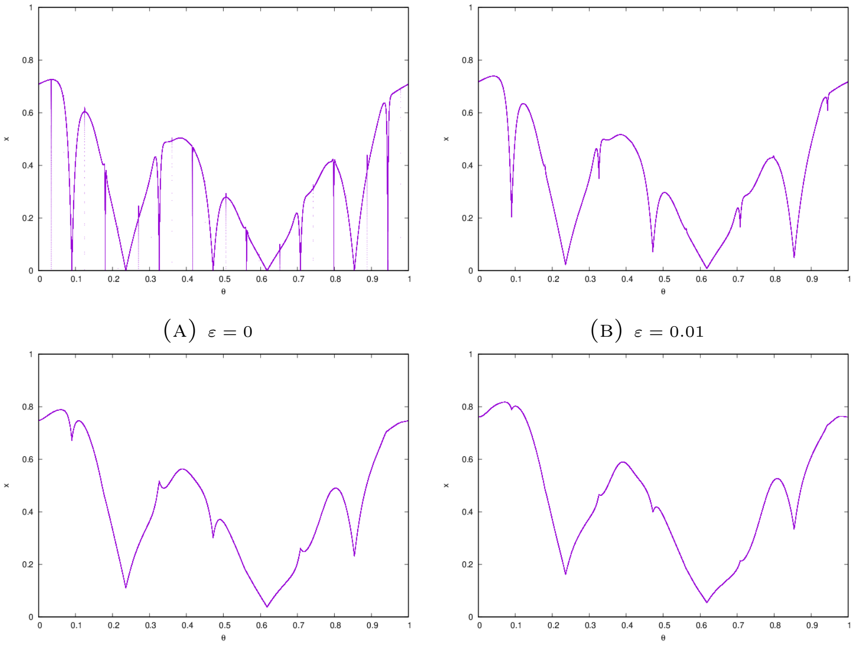
The proof of the existence of these SNAs is general and it is grounded in functional analysis theory. Mostly the author shows that if one iterates the transfer operator, one obtains a series of decreasing functions (on the fibres) and hence one can prove the existence of an upper semi-continuous function as the limit of these decreasing succession of functions.

The study of these systems marks a step up from what was previously done in SNAs, because the proofs are generic and rely heavily in functional analysis. On the other hand, previous results relied heavily on the properties of the system itself (the GOPY case) or (sometimes inconclusive) numerical computations of some metrics of the attractor (the Nishikawa-Kaneko case).

2.1.2.3. *The Alsedà-Misiurewicz model* [**AM08**]. Alsedà and Misiurewicz presented this model

$$\begin{array}{rccc} \mathfrak{F}_{\sigma,\varepsilon}: & \mathbb{S}^1 \times \mathbb{R} & \longrightarrow & \mathbb{S}^1 \times \mathbb{R} \\ & (\theta, x) & \longmapsto & \big(R_\omega(\theta), 4x(1-x)(\varepsilon + \sigma\theta(1-\theta))\big), \end{array} \tag{2.6}$$

as a way to extend the results from Keller [**Kel96**] to a more general setting. In particular they sought to obtain similar results for the case when f_σ is a unimodal $\mathcal{C}^1$ map. The restrictions on the system being a multiplicative forcing and on g_ε still remain. In particular the authors pay a particular attention to the system we will study, becoming the *leitmotiv* of the whole paper. Various instances of the attractor can be found in Figure 2.4. Note that for this system we also have a pinching condition for $\varepsilon = 0$ which leads to the attractor being strange.



(A) $\varepsilon = 0$

(B) $\varepsilon = 0.01$

(C) $\varepsilon = 0.1$

(D) $\varepsilon = 0.25$

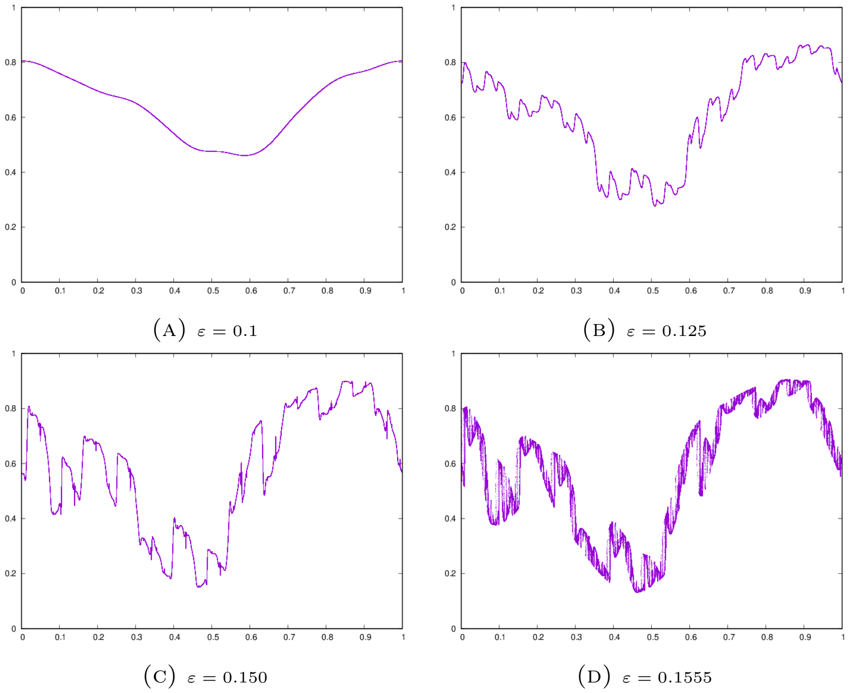
FIGURE 2.4. Plot of the attractor for the Alsedà Misiurewicz model for $\sigma = 3$ and different values of ε. Again, a simple visual inspection already shows the complexity of the pinched case.

2.1.2.4. *The Nishikawa-Kaneko model,* [**NK96**]. By the Nishikawa-Kaneko system we will mean the following system:

$$\begin{array}{rccc} \mathfrak{F}_{\sigma,\varepsilon}: & \mathbb{S}^1 \times \mathbb{R} & \longrightarrow & \mathbb{S}^1 \times \mathbb{R} \\ & (\theta, x) & \longmapsto & \big(R_\omega(\theta), \sigma x(1-x) + \varepsilon \sin(2\pi\theta)\big). \end{array} \tag{2.7}$$

This is the only system that we study that is additively forced and that does not have analytic results proving or disproving its strangeness. The first instance of this system was studied by Kaneko alone in [**Kan84**]. It was then followed by a more in depth study of these kind of systems in [**NK96**].

The authors claim that these kind of systems may have four different behaviours. For some values of ε the system is nonchaotic and it has a smooth attractor. Then, as ε increases, the attractor becomes what the authors call a *pre-fractal torus*, which would correspond to a smooth function that starts to have a complicated geometry (similar to what one would obtain by studying the truncated expansion of a Weierstrass function). By further increasing ε, the attractor become what the authors call a *fractal torus*, which would be classified as a nowhere differentiable continuous function (something like a Weierstrass function), or what we would call an SNA. Finally, there exists a critical value $\varepsilon^\star$ such that if $\varepsilon > \varepsilon^\star$, the system is chaotic and has a chaotic attractor. In Figure 2.5 one can find the plots of the attractor for $\sigma = 3$ and some values of ε, corresponding to the system studied in [**NK96**].



(A) $\varepsilon = 0.1$

(B) $\varepsilon = 0.125$

(C) $\varepsilon = 0.150$

(D) $\varepsilon = 0.1555$

FIGURE 2.5. Plot of the attractor for the Nishikawa-Kaneko model for $\sigma = 3$ and different values of ε. This are cases studied in [**NK96**]. According to the authors Figure C corresponds to a pre-fractal torus and Figure D to a fractal torus.

The arguments to justify the strangeness of the attractor in these systems is completely numerical, and it is based entirely on computational approximations.

The authors compute some metrics of the attractor, mainly its length and its number of extrema. These values are obtained using an increasingly fine mesh of points on the cylinder and arrive to the conclusion that both the length and the number of extrema grow exponentially, so the attractor should be considered a fractal. However, there are some very clear limitations to these methods, all of them acknowledged by the authors themselves. Mostly, the mesh they use has just below ten million points. In particular, it is shown that for some values of ε the length seems to increase exponentially for coarser meshes but then it stabilizes for finer ones. Hence, it could well be that for some even finer meshes the computed length of the attractor stabilizes and becomes finite. Something completely analogous happens with the number of extrema. They claim that they grow as the mesh gets finer, but the mesh cannot be done arbitrarily fine, at least numerically.

2.1.3. Existing analytical results. Finally, after introducing some examples of systems that have been studied as containing an SNA, we will dedicate this subsection to give a very brief survey of the main results that exist for Strange nonchaotic attractors in the case of quasi-periodically forced skew products. Mainly concerning Keller-GOPY and Alsedà-Misiurewicz type systems. However, we need some conditions on f_σ and g_ε:

- $g_\varepsilon \colon \mathbb{S}^1 \longrightarrow [0,\infty)$ continuous, hence bounded.
- $f_\sigma \colon [0,\infty) \longrightarrow [0,\infty)$ in $\mathcal{C}^1$, bounded and such that $f_\sigma(0) = 0$.

With these restrictions the behaviour of system (2.1) is well understood for f_σ either strictly monotone [**Kel96**] or unimodal [**AM08, AM15**].

First of all, we need to check the existence of an attractor.

Remark 2.4. Given a system as in Equation 2.1 satisfying the restrictions above, let us study the existence of invariant objects. Firstly, since $f_\sigma(0) = 0$, it follows that the circle $x \equiv 0$ is invariant. By contrast it is easy to see that all circles $x \equiv \varepsilon$, $\varepsilon > 0$ are not. Furthermore, a natural question is the stability of $x \equiv 0$. To this end, we will compute its vertical Lyapunov Exponent. Recall that the *vertical Lyapunov Exponent* at (θ_0, x_0) is defined as

$$\lambda_{x_0,\theta_0} := \limsup_{k\to\infty} \frac{1}{k} \log \left\| \begin{pmatrix} 1 & 0 \\ \dfrac{\partial x_k}{\partial \theta} & \dfrac{\partial x_k}{\partial x} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| = \limsup_{k\to\infty} \frac{1}{k} \log \left| \frac{\partial x_k}{\partial x} \right|, \tag{2.8}$$

where

$$x_k = F_{\sigma,\varepsilon}\big(R_\omega^{k-1}(\theta_0), x_{k-1}\big) = F_{\sigma,\varepsilon}\Big(R_\omega^{k-1}(\theta_0), F_{\sigma,\varepsilon}\big(R_\omega^{k-2}(\theta_0), x_{k-2}\big)\Big) = \dots$$

Hence, by the chain rule

$$\frac{\partial x_k}{\partial x} = \prod_{k=0}^{n-1} D_x F_{\sigma,\varepsilon}\big(R_\omega^k(\theta_0), x_k\big). \quad \text{(See for example [\textbf{Kel96}])} \tag{2.9}$$

The irrational rotation R_ω ensures that the system is ergodic, so it is natural to apply Birkhoff's Ergodic Theorem.

Theorem 2.5 (Birkhoff's Ergodic Theorem). *Let T be a measure-preserving transformation on a measure space (T, Σ, μ), and suppose f is a μ integrable function. Then if T is ergodic and μ is invariant,*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} f(T^k x) = \frac{1}{\mu(X)} \int_X f d\mu \quad a.e.$$

In other words, the time average of the system is equal to the space average of the function f.

Therefore, by Birkhoff's Ergodic Theorem we can rewrite the vertical Lyapunov Exponent at $x \equiv 0$ as

$$\lambda_{x=0}(f_\sigma, g_\varepsilon) = \int_{\mathbb{S}^1} \log \left| \left. \frac{\partial F_{\sigma,\varepsilon}(\theta, x)}{\partial x} \right|_{x=0} \right| d\theta = \log(f_\sigma'(0)) + \int_{\mathbb{S}^1} \log |g_\varepsilon(\theta)| \, d\theta.$$

Hence, if $f_\sigma'(0) > 1$, $\lambda_{x=0}(f_\sigma, g_\varepsilon)$ is positive. Therefore, $x \equiv 0$ is a repellor for system (2.1). Moreover, since f_σ and g_ε are bounded by construction, infinity is also repelling. Thus, under these assumptions, the systems must have an attractor between $x \equiv 0$ and infinity. That shall be our attractor $\big(\theta, \varphi(\theta)\big)$. Note that both the Keller-GOPY model and the Alsedà-Misiurewicz models satisfy these conditions. As we have already mentioned in the previous subsection, $\big(\theta, \varphi(\theta)\big)$ can be classified as a *Strange Non-Chaotic Attractor* for some values of ε and σ. ∎

The previous remark formalizes what we already noted in Subsection 2.1.2, that the strangeness stems from the fact that the circle $x \equiv 0$ is repelling in the systems studied but for some values of ε a dense set of points of the attractor lay within the repellor. Hence we have repelling points within an attracting set. To describe this phenomenon, we say that *the system is pinched* if there exists θ_0 such that $\varphi(\theta_0) = 0$, i.e our attractor least a point on the circle $x \equiv 0$. Recall that if a system is pinched, then $\varphi\big(R_\omega^n(\theta_0)\big)$ for $n \in \mathbb{N}$, since $x \equiv 0$ is invariant under rotation. Hence $\varphi(\theta) = 0$ on a dense set of $\theta \in \mathbb{S}^1$.

Finally, we can present the following result, which summarizes the main theorem by Keller in [**Kel96**]. In particular, the system we call Keller-GOPY falls within the hypothesis of the theorem. Hence, this result will be used further down the line to check the robustness of the method we propose of using regularity spaces to measure strangeness.

THEOREM 2.6 (Rephrasing of Theorem 1 in [**Kel96**]). *Let $\mathfrak{F}_{\sigma,\varepsilon}$ as in system* (2.1) *such that $F_{\sigma,\varepsilon}$:*

- *$F_{\sigma,\varepsilon}(\theta, x) = f_\sigma(x) g_\varepsilon(\theta)$, with*
- *$g_\varepsilon \colon \mathbb{S}^1 \longrightarrow [0, \infty)$ continuous, hence bounded.*
- *$f_\sigma \colon [0, \infty) \longrightarrow [0, \infty)$ in $\mathcal{C}^1$, bounded, $f_\sigma(0) = 0$ and monotone.*

Then, there exists an upper semi-continuous map $\varphi \colon \mathbb{S}^1 \longrightarrow [0, \infty)$ whose graph $\mathcal{K} = \{(\theta, \varphi(\theta)) : \theta \in \mathbb{S}^1\}$ is invariant under the System (2.1) *and contains an attractor. Moreover,*

(a) *if $f_\sigma'(0) > 1$ and $g_\varepsilon(\theta_0) = 0$ for some θ_0 then the set $\{\theta \colon \varphi(\theta) > 0\} \subset \mathcal{K}$ has full Lebesgue measure whereas the set $\{\theta \colon \varphi(\theta) = 0\} \subset \mathcal{K}$ is residual. Furthermore $(\theta, \varphi(\theta))$ is an SNA.*

(b) *if $f_\sigma'(0) > 1$ and $g_\varepsilon > 0$ then φ is positive and continuous. Furthermore if $g_\varepsilon \in \mathcal{C}^1$ then so is φ,*

(c) *if $f_\sigma'(0) \neq 1$ then $|x_n - \varphi(\theta_n)| \to 0$ exponentially fast for almost every θ and every $x > 0$.*

REMARK 2.7. We refer the reader to [**Kel96**] for a more comprehensive explanations on the proof of such Theorem. However, such proof is strongly based on the fact that f_σ is monotone on the fibres. Thus, the natural question is, what happens if f_σ has a different behaviour. Following this thread, in [**AM08**], the authors consider such function to be unimodal. Using similar ideas as the ones in [**Kel96**] (but introducing several operators and techniques) it is shown that an upper-semiconscious function invariant map $\varphi \colon \mathbb{S}^1 \longrightarrow \mathbb{R}$ which is strictly positive in a set of full Lebesgue measure but $\varphi = 0$ on a dense residual set (see [**AM08, AM15**]). In fact amalgamating the results from [**AM08, AM15**] one can write a version of Theorem 2.6 by replacing the need that f_σ is monotone by the requirement that f_σ is unimodal. ∎

2.2. Obtaining the Lyapunov Exponent

In Definition 2.1, we have defined nonchaoticity to mean a non-positive Lyapunov exponent on the attractor. Moreover, as we will see in Chapter 4, it will be crucial for us to obtain an evaluation of $\varphi(\theta)$ on a mesh of points, in particular dyadic points. To this end, we need a reliable way to compute the Lyapunov exponent with some control over the error. Theorem 2.6(c) guarantees that this exponent is computable and negative. For other cases, we rely on the particularities of the system we study.

First of all, from Equation 2.8 we can get the Lyapunov exponent along the invariant curve φ,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} \log(|\mathrm{D}_x F_{\sigma,\varepsilon}(\varphi(\theta_k)), \theta_k)|).$$

Using Equation 2.9 and defining

$$\log\left(\frac{\partial x_k}{\partial x}\right) := \log(|\mathrm{D}_x F_{\sigma,\varepsilon}(\varphi(\theta_k)), \theta_k)|),$$

we obtain the the n-th approximation of the Lyapunov by

$$\lambda_\varphi^{n)} := \frac{1}{n} \sum_{k=0}^{n-1} \log\left(\frac{\partial x_k}{\partial x}\right).$$

Thus, one can define the *one step*-increment of $\lambda_\varphi^{n)}$ by $\triangle_\varphi^{n)} := \lambda_\varphi^{n)} - \lambda_\varphi^{n-1)}$. Since $\triangle_\varphi^{n)}$ must tend to zero as n tends to infinity we can rephrase the last expression as follows

$$\lambda_\varphi^{n)} = \lambda_\varphi^{n-1)} + \triangle_\varphi^{n)}.$$

In addition, observe that

$$(n-1)\lambda_\varphi^{n-1)} = \sum_{k=0}^{n-2} \log\left(\frac{\partial x_k}{\partial x}\right).$$

Thus, we have that

$$\lambda_\varphi^{n)} = \frac{(n-1)\lambda_\varphi^{n-1)} + \log\left(\dfrac{\partial x_{n-1}}{\partial x}\right)}{n} = \lambda_\varphi^{n-1)} + \frac{\log\left(\dfrac{\partial x_{n-1}}{\partial x}\right) - \lambda_\varphi^{n-1)}}{n}.$$

We can rewrite the above equation using $\triangle_\varphi^{n)}$

$$\triangle_\varphi^{n)} := \frac{\log\left(\dfrac{\partial x_{n-1}}{\partial x}\right) - \lambda_\varphi^{n-1)}}{n}. \tag{2.10}$$

Hence we can put together the following algorithm to compute the vertical Lyapunov exponent of a hyperbolic attractor.

```
ALGORITHM 2.8. procedure LYAPUNOV($\mathfrak{F}_{\sigma,\varepsilon}$, $D_xF_{\sigma,\varepsilon}$, $x_0$, TRANSIENT, TOL)
  $x \leftarrow x_0$
  $\theta \leftarrow Random[0,1]$                          ▷ we set $\theta_0$ to be random
  for $i \leftarrow 1,$ TRANSIENT do        ▷ We use a large transient to ensure convergence
      $\big(\theta, x\big) = \mathfrak{F}_{\sigma,\varepsilon}\,\big(\theta, x\big)$
  end for
  $\triangle_\varphi = \log\Big(\big|D_xF_{\sigma,\varepsilon}(\theta,x)\big|\Big)$                ▷ Initialize $\triangle_\varphi$
  $\lambda_\varphi \leftarrow 0$                                            ▷ Initialize $\lambda_\varphi$
  for $n \leftarrow 2,$ MAX_ITER do        ▷ Predetermined maximum number of iterates
      $\lambda_\varphi \leftarrow \lambda_\varphi \triangle_\varphi$
      $\triangle_\varphi = \frac{\log\left(\left|D_xF_{\sigma,\varepsilon}(\theta,x)\right|\right)-\lambda_\varphi}{n}$
      if $\big|\triangle_\varphi\big| <$ TOL then              ▷ Check if Equation (2.10) holds
          $\lambda_\varphi \leftarrow \lambda_\varphi + \triangle_\varphi$
          Return $\lambda_\varphi$
      end if
  end for
  Return Lack of convergence
end procedure
```

■

CHAPTER 3

Using Daubechies Wavelets with more than 2 vanishing moments

This chapter contains the results we may need regarding wavelet theory. It is not a comprehensive survey on the existing theory, which is too vast and profound, but a summary of the main results that allow us to understand and obtain the results from Chapters 4 and 5.

First and foremost we will give a very short and schematic introduction of the construction of orthonormal families of wavelets. Presenting important concepts such as the *multy resolution analysis* construction, the *scaling function*, the *scaling filter* and the *mother wavelet*. Then, we will focus on Daubechies wavelets, seeing what their main properties are and how these properties play a crucial role in their construction.

Following the construction of the Daubechies wavelet family, we will proceed to understand how we modify wavelet basis on $\mathbb{R}$ to get basis for functions on $\mathbb{S}^1$. This gives rise to an orthonormal basis for $\mathscr{L}^1(\mathbb{S}^1)$ that is not a wavelet basis *per se*, in the sense that it is not generated by translations and dilations of a mother function. However, most of the important properties of wavelet basis are mantained.

Finally, we will present the main result regarding (Besov) regularity and the wavelet coefficients of a function. We are interested in these theorems because, as we shall see, we will use the regularity as an (imperfect) measure of strangeness.

Most results cited can be found in [**HW96, Mal98, Tri10, Tri06, Dau92**].

3.1. A survey on wavelets

As stated before, our main goal is to approximate functions on the cylinder $\varphi : \mathbb{S}^1 \to \mathbb{R}$ by means of a Daubechies wavelet basis. To this end, we shall start by shorty surveying the orthonormal wavelet theory on $\mathbb{R}$ and how it can be adapted to work on $\mathbb{S}^1$.

3.1.1. Wavelets as an orthonormal basis for $\mathscr{L}^2$.

In this section we will present some basic wavelet theory. That is, we will show how to obtain a family of functions $\{\psi_{j,n}\}_{j,n\in\mathbb{Z}} \in \mathscr{L}^2(\mathbb{R})$ such that they are a basis of $\mathscr{L}^2(\mathbb{R})$, and such that

(a) $\psi_{j,n}$ are translations and/or dylations of a mother function, called the mother wavelet, ψ:

$$\psi_{j,n} = 2^{-j/2}\psi\left(\frac{x-2^j n}{2^j}\right), \quad \text{and}$$

(b) they have the following orthogonality condition

$$\int_{\mathbb{R}} \psi_{j,n}(x)\psi_{k,m}(x)dx = \delta_{j,k}\delta_{n,m},$$

where $\delta_{\cdot,\cdot}$ denotes the Kroeneker delta.

Hence, since we have an orthogonal basis, any function $f \in \mathscr{L}^2(\mathbb{R})$ can be written as a linear combination of $\psi_{j,n}$'s. That is,

$$f(x) = \sum_{j\in\mathbb{Z}} \sum_{n\in\mathbb{Z}} d_{j,n}\psi_{j,n}(x), \quad \text{for } d_{j,n} \in \mathbb{R}. \tag{3.1}$$

In order to obtain such bases, we shall present the Multiresolution Analysis construction, which is the most general way of obtaining orthonormal families of wavelets [**Dau92**].

Definition 3.1. A sequence of closed subspaces $\{\mathcal{V}_j\}_{j\in\mathbb{Z}}$ of $\mathscr{L}^2(\mathbb{R})$ is called a *Multiresolution Analysis* (or simply a *MRA*) if it satisfies the following six properties:

(a) $\{0\} \subset \cdots \subset \mathcal{V}_1 \subset \mathcal{V}_0 \subset \mathcal{V}_{-1} \subset \cdots \subset \mathscr{L}^2(\mathbb{R})$.
(b) $\{0\} = \bigcap_{j\in\mathbb{Z}} \mathcal{V}_j$.
(c) $\operatorname{clos}\left(\bigcup_{j\in\mathbb{Z}} \mathcal{V}_j\right) = \mathscr{L}^2(\mathbb{R})$.
(d) There exists a function ϕ whose integer translations, $\phi(x-n)$, $n \in \mathbb{Z}$, form an orthonormal basis of $\mathcal{V}_0$. Such function is called the *scaling function.*
(e) For each $j \in \mathbb{Z}$ it follows that $f(x) \in \mathcal{V}_j$ if and only if the *dyadic translations* verify that $f(x - 2^j n) \in \mathcal{V}_j$ for each $n \in \mathbb{Z}$.
(f) For each $j \in \mathbb{Z}$ it follows that $f(x) \in \mathcal{V}_j$ if and only if the *dilation* verifies that $f(x/2) \in \mathcal{V}_{j+1}$.

■

Remark 3.2. Let $p_k : \mathscr{L}^2(\mathbb{R}) \to \mathcal{V}_k$ be the projection from $\mathscr{L}^2(\mathbb{R})$ to $\mathcal{V}_k$. Then

$$\lim_{k\to\infty} p_k f = f$$

In fact, one should think of each $\mathcal{V}_k$ as a sequence of spaces on which finer approximations of f might be achieved. ■

Remark 3.3. Once an MRA is fixed, $\mathcal{V}_j$ has an orthonormal basis $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$, for every j, where

$$\phi_{j,n}(x) = 2^{-j/2}\phi\left(\frac{x-2^j n}{2^j}\right).$$

■

Now, define the subspace $\mathcal{W}_j$ as the orthogonal complement of $\mathcal{V}_j$ on $\mathcal{V}_{j-1}$, that is,

$$\mathcal{V}_{j-1} = \mathcal{W}_j \oplus \mathcal{V}_j. \tag{3.2}$$

In this sense, one can think of $\mathcal{W}_j$ as a sequence of increasignly coarser spaces in which to approximate f. Therefore, by the inclusion of the spaces $\mathcal{V}_j$ we have

$$\mathscr{L}^2(\mathbb{R}) = \operatorname{clos}\left(\bigoplus_{j\in\mathbb{Z}} \mathcal{W}_j\right) = \operatorname{clos}\left(\mathcal{V}_0 \oplus \bigoplus_{j=0}^{\infty} \mathcal{W}_{-j}\right). \tag{3.3}$$

Definition 3.4. Let $\{\mathcal{V}_j\}_{j\in\mathbb{Z}}$ be a Multiresolution Analysis and let $\phi(x)$ be its scaling function. Define

$$h[n] := \left\langle \frac{1}{\sqrt{2}}\phi\left(\frac{x}{2}\right), \phi(x-n)\right\rangle, \quad \text{for } n \in \mathbb{Z}, \tag{3.4}$$

where $\langle\,\cdot\,,\cdot\rangle$ denotes the usual $\mathscr{L}^2(\mathbb{R})$ scalar product. The sequence $h[n]$ is called the *scaling filter* (or the *low pass filter*) of the Multiresolution Analysis. We define the support of h, denoted by $\operatorname{supp}(h)$, as the minimum subset $\mathfrak{I}$ of $\mathbb{Z}$ such that $\mathfrak{I} = \{\ell, \ell+1, \ldots, \ell'\}$ is a set of consecutive integers and

$$h[n] = 0 \text{ for every } n \in \mathbb{Z}\backslash\mathfrak{I}.$$

■

Now we can define the so-called *mother wavelet*, which by dylations and translations will generate a basis of $\mathscr{L}^2(\mathbb{R})$.

Definition 3.5. The *mother wavelet* $\psi \in \mathcal{W}_0$ is defined to be the function whose Fourier transform is

$$\widehat{\psi}(\xi) = \frac{1}{\sqrt{2}} e^{-i\xi} \widehat{h}^*(\xi + \pi)\widehat{\phi}(\xi) \tag{3.5}$$

where $\widehat{h}^*(\xi)$ is the complex conjugate of

$$\widehat{h}(\xi) = \sum_{n\in\mathbb{Z}} h[n]e^{-in\xi}, \tag{3.6}$$

with $\widehat{h}(0) = \sqrt{2}$. ■

The following theorem (see [**Mal98**, Theorem 7.3]) ensures that we can contruct a wavelet basis from the mother wavelet.

Theorem 3.6 (Mallat, Meyer). *The mother wavelet given by Equation* (3.5) *verifies that, for each integer j, the family $\{\psi_{j,n}\}_{n\in\mathbb{Z}}$ is an orthonormal basis of $\mathcal{W}_j$, where:*

$$\psi_{j,n}(x) = 2^{-j/2}\psi\left(\frac{x - 2^j n}{2^j}\right). \tag{3.7}$$

As a consequence, the family $\{\psi_{j,n}\}_{(j,n)\in\mathbb{Z}\times\mathbb{Z}}$ is an orthonormal basis of $\mathscr{L}^2(\mathbb{R})$.

It follows that one needs ϕ and h to determine a wavelet basis. Taking into account (3.3) and the theorem above, every map $f \in \mathscr{L}^2(\mathbb{R})$ can be written as

$$\begin{aligned} f(x) &= \sum_{j\in\mathbb{Z}}\sum_{n\in\mathbb{Z}} \langle f, \psi_{j,n}\rangle\, \psi_{j,n}(x) \\ &= \sum_{n\in\mathbb{Z}} \langle f, \phi_{0,n}\rangle\, \phi_{0,n}(x) + \sum_{j=0}^{\infty}\sum_{n\in\mathbb{Z}} \langle f, \psi_{-j,n}\rangle\, \psi_{-j,n}(x). \end{aligned}$$

Of course depending on the choice of the scaling function, the mother wavelet ψ can be dramatically different. There are several wavelets families, such as Daubechies, Symlets, Morlet, Coiflets, or Shannon (see e.g. [**Mal98**]). Each family has its own properties that make them suited for particular applications. Through this paper, we will focus in *Daubechies wavelets*. This choice will be clear later on the thesis, but it mainly rests on two facts: they are the best suited to work on compact supports (see Lemma 3.24, for example) and they can be used to compute the regularity of functions, as it can be seen in Theorem 3.35.

3.1.2. Constructing a wavelet basis from a scaling filter. In practice, however, we will rarely construct a whole MRA in order to obtain a wavelet. We would like to be able to construct a wavelet basis just by choosing suitable $h[n]$. To this end, we present the following result from Mallat and Meyer:

Theorem 3.7. *(Mallat, Meyer, Theorem 7.2 in* [**Mal09**]*) Let $\phi \in \mathcal{L}^2(\mathbb{R})$ be an integrable scalling function. The fourier series of $h[n] = \langle 2^{-1/2}\phi(t/2), \phi(t-n)\rangle$ satisfies for all $\omega \in \mathbb{R}$*

$$|\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2 \tag{3.8}$$

and

$$\hat{h}(0) = \sqrt{2}. \tag{3.9}$$

Conversely, if $\hat{h}(\omega)$ is 2π-periodic and continuously differentiable in a neighbourhood of $\omega = 0$, if it satisfies the above two conditions, and if

$$\inf_{\omega \in [-\pi/2, \pi/2]} |\hat{h}(\omega)| > 0$$

then

$$\hat{\phi}(\omega) = \prod_{p=1}^{\infty} \frac{\hat{h}(2^{-p}\omega)}{\sqrt{2}}$$

is the Fourier transform of a scaling funcion $\phi \in \mathcal{L}^2(\mathbb{R})$.

Remark 3.8. Some properties regarding how h, ϕ and ψ relate to one another are derived from Theorem 3.7. In particular:

(a) Relating h and ϕ with ψ we get

$$\frac{1}{2}\psi\left(\frac{x}{2}\right) = \sum_{-\infty}^{\infty} (-1)^{1-n} h[1-n] \psi(t-n) \tag{3.10}$$

Note that in the literature $(-1)^{1-n}h[1-n]$ is usually denoted by $g[n]$ and can also be constructed as

$$g[n] = \left\langle \psi\left(\frac{x}{2}\right), \phi(t-n) \right\rangle$$

(b) We have

$$\sum_{n\in\mathbb{Z}} h[2n] = \frac{\sqrt{2}}{2} = \sum_{n\in\mathbb{Z}} h[2n+1] \tag{3.11}$$

■

Theorem 3.7 gives us the necessary conditions on h to obtain a scaling function ϕ that generates a MRA. From this ϕ we obtain the mother wavelet ψ by Definition 3.1.1. Hence, a suitable h become a backdoor to obtain mother wavelets without the need to construct entire MRAs. Following this line of reasoning, we might be interested in how properties of ψ and h may relate. One such property is compact support. The definition of support for wavelets is exactly what one would expect.

Definition 3.9. Let $\psi(x)$ be a wavelet derived from a Multiresolution Analysis $\{\mathcal{V}_j\}_{j\in\mathbb{Z}}$. We define the *support* of $\psi(x)$, and we will denote it by $\text{supp}(\psi)$, as the minimum interval of the real line such that $\psi(x) = 0$ for every x which does not belong to the support of $\psi(x)$. The size of $\text{supp}(\psi)$ is defined as the length of such interval. ■

Now we have a theorem that tells us how the support of ψ, ϕ and h relate to one another.

Theorem 3.10 (Compact Support, Theorem 7.5, [**Mal09**]). *The scaling function ϕ has compact support if and only if h has compact support. Moreover their supports are equal. If the support of h and ϕ is $[N_1, N_2]$, then the support of ψ is exactly $[(N_1 - N_2 + 1)/2, (N_2 - N_1 + 1)/2]$, hence compact.*

In the following section we will give an overview of the construction of Daubechies wavelets from a carefully chosen h.

3.2. Daubechies Wavelets

The main properties of Daubechies wavelets Daubechies wavelets have the following core properties:

(a) They are an orthonormal basis of $\mathscr{L}^2(\mathbb{R})$.
(b) All the coefficients of h are real.

(c) They have p vanishing moments.
(d) They have minimal compact support, in the sense of Theorem 3.13.

In particular, the interest in compact support will be clear in Chapters 4, and specially 5. This stems from the fact that the smaller the support the more efficient the computations become and the less momory we will require to compute the wavelet expansion of the attractor.

The properties (a) and (b) are pretty straightforward, property (d) has been introduced in the previous subsection. Now we shall introduce the third one.

Definition 3.11. Let $\psi(x)$ be a wavelet from a Multiresolution Analysis $\{\mathcal{V}_j\}_{j\in\mathbb{Z}}$. We say that $\psi(x)$ has *$p-$Vanishing Moments* if the integer p is the greatest number such that

$$\int_{\mathbb{R}} x^k \psi(x)\ dx = 0 \textit{ for } 0 \leq k < p.$$

■

Remark 3.12. Notice that Definition 3.11 means that $\psi(x)$ is orthogonal to any polynomial of degree $p-1$. ■

Now, we can see how vanishing moments and compact support relate to one another.

Theorem 3.13 (Daubechies, Theorem 7.7, [**Mal09**]). *Let ψ be an orthonormal mother wavelet with p-vanishing moments, then h has support size at least* $2p-1$.

Precisely, this theorem tells us what it is meant for minimal compact support. Fixing a number of vanishing moments forces a lower bound for the support of the wavelet. In particular, Daubechies wavelets are obtained when size of the support is made to be equal to $2p-1$. The process by which these wavelets are obtained is constructive and a bit involved, hence a more detailed contruction of Daubechies wavelets can be found in the following subsection.

3.2.1. Construction the Daubechies Wavelet Family. In this subsection we want to give a more detailed construction of the Daubechies family of wavelets. We will present a mix between Chapter 7 in [**Mal09**], and Chapter 6 in [**Dau92**], taking and mixing results from both. Recall that the core properties of Daubechies wavelets is that they are an orthonormal basis of $\mathscr{L}^2(\mathbb{R})$, all the coefficients $h[n]$ are real, they have minimal compact support, and p–vanishing moments. We believe it is interesting to see the whole construction of the Daubechies Wavelet Family, since it clearly shows that its core properties forces it to stray away from computational efficiency. Naively one could just decide on $2p$ real coefficients (from 0 to $2p-1$) and plug them in one after the other in h and set all the other coefficients to zero. However, Theorem 3.7 tells us that not all such sequences may work. We need to find a way to generate the coefficients such that the requirements for Daubechies wavelets are met whilst complying with Theorem 3.7.

First though, we will introduce a technical Theorem, which presents an easier way to compute the number of vanishing moments of a mother wavelet other than computing integrals against monic polynomials.

Theorem 3.14 (Theorem 7.4, [**Mal09**]). *Let ψ and ϕ be a wavelet and a scaling function that generate an orthogonal basis. Suppose that $|\psi(t)| = \mathcal{O}\big((1+t^2)^{-p/2-1}\big)$ and $|\phi(t)| = \mathcal{O}\big((1+t^2)^{-p/2-1}\big)$. Then the following are equivalent:*

(i.) The wavelet ψ has p vanishing moments.
(ii.) $\hat{\psi}(\omega)$ and its first $p-1$ derivatives are zero at $\omega = 0$.
(iii.) $\hat{h}(\omega)$ and its first $p-1$ derivatives are zero at $\omega = \pi$.

REMARK 3.15. Note that the $\mathcal{O}$ conditions are trivially met for compactly supported wavelets. ■

Now we can start our overview on the construction of Daubechies wavelets. Firstly, we want the filter h to have real coefficients, hence $\hat{h}$ will be a trigonometric polynomial

$$\hat{h}(\omega) = \sum_{n=0}^{N-1} h[n]e^{-in\omega}. \tag{3.12}$$

If we want ψ to have p vanishing moments, Theorem 3.14 tells us that $\hat{h}$ has a zero of order p at $\omega = \pi$. Since we want to minimize number of coefficients of the triogonometric polynomial (i.e minimizing N in equation (3.12)) we will factor out

$$(1 + e^{-i\omega})^p$$

from $\hat{h}(\omega)$, which is the minimal polynomial with p zeroes at $\omega = \pi$. Hence we can write

$$\hat{h}(\omega) = \sqrt{2}\left(\frac{1 + e^{-i\omega}}{2}\right)^p \chi(e^{-i\omega}) \tag{3.13}$$

Now our objective is to find a suitable polynomial $\chi(e^{-i\omega})$ of minimum degree m which will give us a valid h. In particular, by Theorem 3.7 we require

$$|\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2. \tag{3.14}$$

Hence, h has $N = m + p + 1$ nonzero coefficients. Theorem 3.17 shows us that m is at least $p - 1$. However, let us recall first the Bezout Theorem for polynomials.

THEOREM 3.16 (Bezout, Theorem 6.1.1[**Dau92**]). *Let $Q_1(y)$ and $Q_2(y)$ be two polynomials with no common zeroes of degrees n_1 and n_2 respectively. Then, there exist two unique polyomials $P_1(y)$ and $P_2(y)$ of degrees $n_2 - 1$ and $n_1 - 1$ such that*

$$P_1(y)Q_1(y) + P_2(y)Q_2(y) = 1$$

We will need the theorem above for the prove of the following theorem.

THEOREM 3.17 (Daubechies, Theorem 7.7, [**Mal09**]). *A low pass filter h such that $\hat{h}(\omega)$ has p zeroes at $\omega = \pi$ has at least $2p$ non-zero coefficients.*

PROOF. The proof is constructive, and it will give us $h[n]$ for a given p. Since $h[n]$ is real, $|\hat{h}(\omega)|^2$ is an even function. Hence, it can be written as a polynomial in $\cos(\omega)$. Therefore, in the notation of Equation (3.13), $|\chi(e^{-i\omega})|^2$ is a polynomial in $\cos(\omega)$ as well. Since

$$\sin^2\left(\frac{\omega}{2}\right) = \frac{1 - \cos(\omega)}{2}$$

we can rewrite $|\chi|^2$ as a polynomial $P(\sin^2(\omega/2))$, giving us

$$|\hat{h}(\omega)|^2 = 2\cos\left(\frac{\omega}{2}\right)^{2p} P\left(\sin^2\frac{\omega}{2}\right)$$

Setting $y = \sin^2(\omega/2)$ we can rewrite Equation (3.14) as

$$(1 - y)^p P(y) + y^p P(1 - y) = 1. \tag{3.15}$$

Thus, minimizing the number of non-zero terms of $\hat{h}(\omega)$ corresponds to finding a solution to $P(y)$ of minimal degree. This solution can be obtained with Theorem 3.16, since there exist two unique polynomials P_1 and P_2 of degree $p - 1$ such that

$$(1 - y)^p P_1(y) + y^p P_2(y) = 1$$

Moreover, if we subtitute $1 - y$ for y we obtain

$$y^p P_1(1 - y) + (1 - y)^p P_2(1 - y) = 1.$$

By uniqueness of P_1 and P_2, we get $P_2(y) = P_1(1-y)$. Hence, it follows that $P_1(y)$ is a solution of Equation (3.15). In this case, we can obtain an explicit form for P_1, since

$$\begin{aligned} P_1(y) =&(1-y)^{-p}[1 - y^p P_1(1-y)] \\ =&\sum_{k=0}^{p-1} \binom{p+k-1}{k} y^k + \mathcal{O}(y^p) \end{aligned}$$

were this follows from a Taylor expansion of the first p coefficients for $(1-y)^{-p}$. Since we know that the degree of P_1 is $p-1$, we obtain that

$$P_1(y) = \sum_{k=0}^{p-1} \binom{p+k-1}{k} y^k.$$

Hence, by setting $P(y) = P_1(y)$ we obtain an explicit solution to Equation (3.15). This is the unique with lowest degree, but by no means the unique one. For any other higher degree solution $\widetilde{P}$, we have

$$(1-y)^p[\widetilde{P}(y) - P(y)] + y^p[\widetilde{P}(1-y) - P(1-y)] = 0. \tag{3.16}$$

Hence $\widetilde{P} - P$ is divisible by y^p,

$$\widetilde{P}(y) - P(y) = y^p \widetilde{P}(y).$$

Moreover,

$$\widetilde{P}(y) + \widetilde{P}(1-y) = 0,$$

in other words, $\widetilde{P}$ is antisimetric with respect to $\frac{1}{2}$. In any case, since we are interested in minimizing the number of coefficients, we will stick to the polynomial with minimal degree, $P(y)$.

However, we are interested in obtaining information about $\hat{h}(\omega)$ and so far we have been working with $|\hat{h}(\omega)|^2$, hence we need to find a the polynomial χ from Equation (3.13) such that $|\chi(e^{-i\omega})|^2 = P(\sin^2(\omega/2))$, something like a square root. Moreover, another requirement is that the coefficients of χ should be real. So χ should be of the form

$$\chi(e^{-i\omega}) = \sum_{k=0}^{m} r_k e^{-ik\omega} = r_0 \prod_{k=0}^{m} (1 - a_k e^{-i\omega}) \tag{3.17}$$

where a_k are the roots of χ. Note, moreover, that since the coefficients of χ are real, the conjugate of $\chi(e^{-i\omega})$ is simply $\chi(e^{i\omega})$. Hence,

$$|\chi(e^{-i\omega})|^2 = \chi(e^{-i\omega})\chi(e^{i\omega}) = P(\sin^2(\omega/2)) = P\left(\frac{2 - e^{i\omega} - e^{-i\omega}}{4}\right) := Q(e^{-i\omega})$$

To solve this factorization it is helpful to extend the polynomials to $\mathbb{C}$ by setting $z := e^{-i\omega}$. Hence,

$$Q(z) = \chi(z)\chi(z^{-1}) = r_0^2 \prod_{k=0}^{m} (1 - a_k z)(1 - a_k z^{-1}) = \prod_{k=0}^{m} \left(1 + a_k^2 - a_k(z + z^{-1})\right)$$

which is a polynomial in $z + z^{-1}$. Putting all of this together we know that if c_k is a root of Q, then so it is its conjugate $\bar{c}_k$ and $\frac{1}{c_k}$. This automatically implies that $1/\bar{c}_k$ is also a root. Hence, to build R we only need to be cautious with our root-picking. For each pair $(c_k, 1/c_k)$ we might take the root a_k of R to be the one of the two that satisfies $|a_k| \leq 1$ in addition to pick $\bar{a}_k$ so that the coefficients are real. This procedure gives us a polynomial of minimum degree $p-1$. Moreover, we can take $r_0^2 = Q(0) = P(1/2) = 2^{p-1}$. Therefore, our polynomial $\widehat{h}(\omega)$ has degree $p + p - 1 = 2p - 1$, hence $2p$ different coefficients. Since by Equation (3.6) we

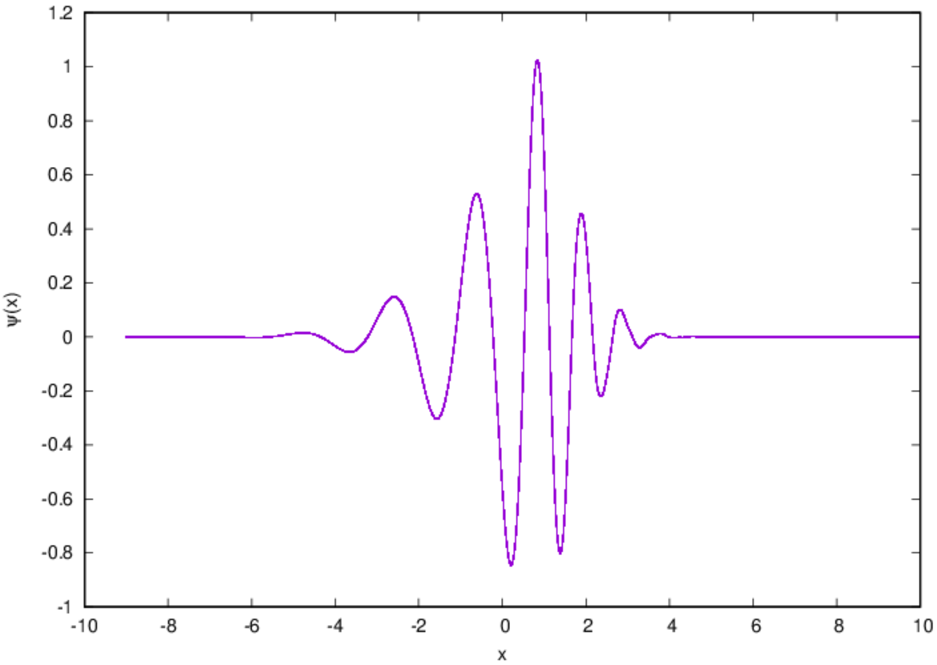


FIGURE 3.1. Daubechies mother wavelet with 10 vanishing moments.

have that the coefficients of $\widehat{h}$ correspond to the elements of h, we have constructed h such that it satisfies Theorem 3.7, has real coefficients, and has support to be exactly [0,2p-1], which is the Daubechies Filter. □

REMARK 3.18. By the construction above and Theorem 3.10, we have that the Daubechies mother wavelet has support exactly $\text{supp}(\psi) = [-p+1, p]$. Moreover, any $\psi_{j,n}$ will have support

$$\text{supp}(\psi) = [2^j(1-p+n), 2^j(p+n)].$$

This is important because when we will define the periodized wavelets in Subsection 3.3 we will see that the fact that the size of the support decreases as j becomes negative will have a tremendous importance. ■

It is crucial to remark again that Daubechies wavelets with $p > 1$ vanishing moments do not have a closed expression. As it has been seen in their construction, the only goal is obtaining wavelets that satisfy certain conditions regarding compactness and their vanishing moments. As we will see, this will pose a trade-off because the number of vanishing moments play an important role when using wavelets as basis of regularity spaces (see Section 3.4). However, as we shall see in Chapter 5, the more vanishing moments a wavelet has, the harder it becomes to evaluate it at a given point. Hence, we feel that a good compromise between vanishing moments and realistic computational requirements are Daubechies wavelets with 10 vanishing moments (see Figure 3.1).

Until now, we have been presenting results for wavelets defined on $\mathbb{R}$. However, we are interested in $\mathscr{L}^2$ functions defined on $\mathbb{S}^1$. In the following subsection, we will translate the $\mathbb{R}$-language of $\psi(x)$, to the $\mathbb{S}^1$-language of $\psi^{\text{PER}}(\theta)$.

3.3. Wavelets on the unit circle

Since we are interested in quasi-periodically forced skew products, our natural scenario will be to study functions $\varphi : \mathbb{S}^1 \to \mathbb{R}$, where $\mathbb{S}^1 = \mathbb{R}/\mathbb{Z}$. Hence, we need to adapt the Daubechies wavelets family to obtain a good orthonormal basis for $\mathscr{L}^2(\mathbb{S}^1)$. It turns out that only a slight adaptation on the family defined on $\mathbb{R}$ is enough for it to work on $\mathbb{S}^1$. We call this modification the *perioditization of the wavelet family.*

In Section 4.5 from [**HW96**] wavelets are constructed to be an orthonormal basis of $\mathscr{L}^2(\mathbb{S}^1)$ using the "*periodization*" given by

$$\psi^{\text{PER}}_{j,n}(\theta) = \sum_{l\in\mathbb{Z}} \psi_{j,n}(\theta + l) = 2^{-j/2} \sum_{l\in\mathbb{Z}} \psi\left(\frac{(\theta + l) - 2^j n}{2^j}\right). \tag{3.18}$$

The following theorem shows the importance of these periodized wavelets.

Theorem 3.19 (Theorem 5.9, [**HW96**]). *Let $\psi(x)$ be the mother wavelet of an orthonormal wavelet basis from a $\mathbb{R}$-MRA. Then the system given by*

$$\{1, \psi^{\text{PER}}_{-j,n} \text{ with } j \geq 0 \text{ and } n = 0, 1, \ldots, 2^j - 1\}$$

is an orthonormal basis of $\mathscr{L}^2(\mathbb{S}^1)$.

Remark 3.20. Note that in this case, for a fixed j we no longer need to have infinitely many values of n to complete the basis. Instead, for each j-level we have exactly 2^j elements of the basis, namely $\psi^{\text{PER}}_{-j,0}, \psi^{\text{PER}}_{-j,1}, \ldots, \psi^{\text{PER}}_{-j,2^j-1}$. Hence, we can define

$$\mathcal{V}^{\text{PER}}_0 := \langle\!\langle 1 \rangle\!\rangle \quad \text{and} \quad \mathcal{W}^{\text{PER}}_{-j} := \langle\!\langle \psi^{\text{PER}}_{-j,0}, \psi^{\text{PER}}_{-j,1}, \ldots, \psi^{\text{PER}}_{-j,2^j-1} \rangle\!\rangle,$$

where $\langle\!\langle f_1, f_2, \ldots, f_n \rangle\!\rangle$ denotes the space generated by $f_1, f_2, \ldots, f_n$. By Theorem 3.19 it follows that the expression on $\mathbb{S}^1$ equivalent to equation (3.3) becomes

$$f = a_0 + \sum_{j=0}^{\infty} \sum_{n=0}^{2^j-1} \left\langle f, \psi^{\text{PER}}_{-j,n} \right\rangle \psi^{\text{PER}}_{-j,n} \in \mathcal{V}_0 \oplus \bigoplus_{j=0}^{\infty} \mathcal{W}^{\text{PER}}_{-j}. \tag{3.19}$$

And if now we are interested in obtaining a truncated expression for (3.19) we only need to choose a suitable ν and hence we have

$$f \approx a_0 + \sum_{j=0}^{\nu} \sum_{n=0}^{2^j-1} \left\langle f, \psi^{\text{PER}}_{-j,n} \right\rangle \psi^{\text{PER}}_{-j,n}. \tag{3.20}$$

This means that to truncate a wavelet expansion, we can simply choose our cut for a certain maximum j, which we will call ν, and automatically our expansion will have $2^{\nu+1}$ elements. ∎

Remark 3.21. Note that even if we are using the wavelet notation and refer to them as *periodized wavelets*, $\left\{\psi^{\text{PER}}_{j,n}\right\}$ aren't in fact a wavelet basis *per se.* As we have seen, they are a basis of $\mathscr{L}^2(\mathbb{S}^1)$, but this basis is not generated from a single mother wavelet. Note that if we were consider that ψ^{PER} is a sort of *periodized mother wavelet*, we would obtain

$$\begin{aligned} 2^{-j/2}\psi^{\text{PER}}\left(\frac{x - 2^j n}{2^j}\right) &= 2^{-j/2} \sum_{\ell\in\mathbb{Z}} \psi\left(\frac{x - 2^j n}{2^j} + l\right) \\ &= 2^{-j/2} \sum_{\ell\in\mathbb{Z}} \psi\left(\frac{x}{2^j} + l - n\right) \neq \psi^{\text{PER}}_{j,n}(x) \end{aligned}$$

In fact, in this case the 1-periodicity of $2^{-j/2}\psi^{\text{PER}}\left(\frac{x-2^j n}{2^j}\right)$ is not guaranteed. ∎

The following proposition shows us the similarities between $\psi^{\text{PER}}_{-j,n}$ and $\psi^{\text{PER}}_{-j,m}$. That is, as in the case of $\mathbb{R}$-wavelets, n corresponds to a simple translation, even though in this case it is of size $\frac{n}{2^j}$. Moreover, one needs to keep in mind that these translations are $\mod 1$. For the sake of simplicity we have introduced the following notation:

$$\mathsf{mod}\big(n+k,\, 2^j\big) := n+k \mod 2^j.$$

Proposition 3.22. *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments. Then, for every $\theta \in \big[0,1)$ and any $\alpha \in [0,1)$, $j \in \{1,\dots,\nu\}$, and $n \in \{0,1,\dots,2^j-1\}$,*

$$\psi^{\text{PER}}_{-j,n}\big(R_\alpha(\theta)\big) = \psi^{\text{PER}}_{-j,\mathsf{mod}\left(n+k,\,2^j\right)}\left(R_\alpha\left(\theta+\tfrac{k}{2^j}\right)\right)$$

for every $k = 1,2,\dots,2^j-1$.

Proof. We have $\mathsf{mod}\big(n+k,\,2^j\big) = n+k-r\cdot 2^j$ where $r = \big\lfloor \frac{n+k}{2^j}\big\rfloor \in \{0,1\}$. Then, from Equation (3.18) it follows that,

$$\psi^{\text{PER}}_{-j,\mathsf{mod}\left(n+k,\,2^j\right)}\left(R_\alpha\left(\theta+\tfrac{k}{2^j}\right)\right) = \psi^{\text{PER}}_{-j,n+k-r\cdot 2^j}\left(R_\alpha\left(\theta+\tfrac{k}{2^j}\right)\right) =$$

$$2^{j/2}\sum_{l\in\mathbb{Z}}\psi\left(\frac{(\theta+k\cdot 2^{-j}+\alpha+l)-2^{-j}(n+k-r\cdot 2^j)}{2^{-j}}\right) =$$

$$2^{j/2}\sum_{l\in\mathbb{Z}}\psi\left(\frac{(\theta+r+\alpha+l)-2^{-j}n}{2^{-j}}\right) =$$

$$2^{j/2}\sum_{l\in\mathbb{Z}}\psi\left(\frac{(\theta+\alpha+l)-2^{-j}n}{2^{-j}}\right) = \psi^{\text{PER}}_{-j,n}\big(R_\alpha(\theta)\big).$$

□

Remark 3.23. Note that in the above proposition α might have any value in $[0,1)$, in particular it can be 0 or it can be an irrational rotation given by ω. This proposition will be crucial in what follows, since it indicates that fixing $j \in \mathbb{N}$, the values of $\psi^{\text{PER}}_{-j,n}$ on $\mathbb{S}^1$ are the same for all n, just shifted around in intervals of length $\frac{1}{2^j}$. Figure 3.2 shows this behaviour. ■

Finally, since we are interested mostly in compactly supported wavelets, it would be expected that the infinite sum over l in Equation (3.18) should be finite. Hence, from Definition 3.9 Equation (3.18) we get the following proposition.

Lemma 3.24. *Let $\psi(x)$ be a mother wavelet for an orthonormal wavelet family from a $\mathbb{R}$-MRA with compact support. Let $\operatorname{supp}(\psi) = [a,b]$. For given j and n let us define*

$$l_{\min}(\theta) = \big\lceil 2^j(a+n)-\theta\big\rceil \qquad l_{\max}(\theta) = \big\lfloor 2^j(b+n)-\theta\big\rfloor$$

where $\theta \in [0,1)$ (i.e. $\theta \in \mathbb{S}^1$). Then,

$$\psi^{\text{PER}}_{j,n}(\theta) = \sum_{l=l_{\min}(\theta)}^{l_{\max}(\theta)} \psi_{j,n}(\theta+l).$$

Proof. Given $\theta \in [0,1)$, from equation (3.18) we have

$$\psi^{\text{PER}}_{j,n}(\theta) = \sum_{l\in\mathbb{Z}}\psi_{j,n}(\theta+l).$$

Since $\operatorname{supp}(\psi) = [a,b]$ from Equation 3.7 it follows that $\operatorname{supp}\big(\psi_{j,n}\big) = \big[2^j(a+n), 2^j(b+n)\big]$. Hence, in the sum above whenever $\theta+l < 2^j(a+n)$ or $\theta+l > 2^j(b+n)$ $\psi_{-j,n}(\theta+l) = 0$. Since $l \in \mathbb{Z}$, the result follows. □

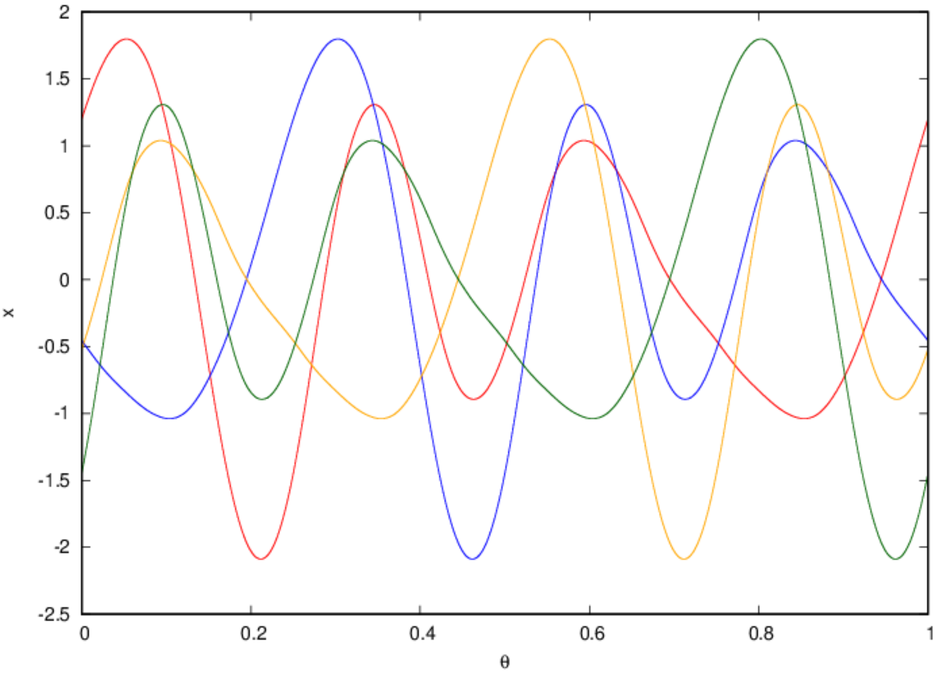


FIGURE 3.2. Plot of the periodized Daubechies wavelets with 10 vanishing moments corresponding to $j = -2$. In red $\psi^{\text{PER}}_{-2,0}$, in blue $\psi^{\text{PER}}_{-2,0}$, in orange $\psi^{\text{PER}}_{-2,0}$ and in green $\psi^{\text{PER}}_{-2,0}$. Notice the fact that all of them are just a shift $\mod 1$ of $\psi^{\text{PER}}_{-2,0}$ of length $\frac{n}{4}$.

3.4. Wavelets as a (flawed) way to measure strangeness

In this section we will see how one can use the wavelet basis as a way to measure the strangeness of an attractor, seeing also the limitations of these techniques.

The strangeness will be measured by computing and estimate of the *regularity* of functions, in particular the regularity in *Besov spaces*. This has probably been one of the most studied topic in analysis in the 20th century, and even whole series of books have been written about this topic [**Tri83, Tri92, Tri06**]. The theory is way more complex and profound than anything we are going to use, but nevertheless they are a useful tool, though flawed, for our purposes.

In general, smoothness (or regularity) is measured through the inclusion to some functional space. The classic example is the chain

$$\mathcal{C}^\infty(\mathbb{R}) \subset \cdots \subset \mathcal{C}^m(\mathbb{R}) \subset \cdots \subset \mathcal{C}^2(\mathbb{R}) \subset \mathcal{C}^1(\mathbb{R}) \subset \mathcal{C}^0(\mathbb{R}) \tag{3.21}$$

However, this regularity chain is not complete by any means. For example between $\mathcal{C}^1$ and $\mathcal{C}^0$ one can find the Lipschitz continuous functions, which clearly deserve a space by themselves. Another example of a class of functions that occupy this space in particular are the absolutely continuous functions. Therefore, over the 20th century there has been an effort to try and generalize the notion of regularity of functions through big families of regularity spaces, not necessarily equivalent to one another but that give structured classifications of functions. One such families are the Besov spaces $\mathscr{B}^s_{p,q}$.

3.4.1. Basic notions. In this subsection we will present a variety of normed functional spaces that will lead to the definition of $\mathscr{B}^s_{p,q}$ for $s > 0$, $0 < p < \infty$ and $0 < q \leq \infty$. Finally, we will explore some properties of these spaces that will justify our interest for $\mathscr{B}^s_{\infty,\infty}$. All the results stem from **[Tri83]**. To simplify the notation, assume that all of these spaces are defined on $\mathbb{R}$, i.e., are of dimension one. Let us start with the simplest of spaces and build up from there.

(i) Starting from the basics, we need to consider $\mathcal{C}^m(\mathbb{R})$, which we will endow with the following norm

$$\|f\|_{\mathcal{C}^m} = \sum_{k\leq m} \left\| f^{(k)}(x) \right\|_{\mathscr{L}_\infty}, \tag{3.22}$$

where $f^{(n)}$ denotes the *nth* derivative of f. Then,

$$\mathcal{C}^m(\mathbb{R}) := \left\{ f \in \mathcal{C}^0 : \|f\|_{\mathcal{C}^m} \text{ is well defined and finite} \right\}.$$

(ii) The classic generalization of the Lipschitz spaces, the *Hölder spaces*, usually denoted by $\mathcal{C}^s(\mathbb{R})$ with $s > 0$, $s \notin \mathbb{Z}$. They can be understood as filling the gaps in the chain from equation (3.21). However, $\mathcal{C}^s(\mathbb{R})$ is not quite general enough, since they are not well defined when $s \in \mathbb{Z}$. For these spaces, we define the norm

$$\|f\|_{\mathcal{C}^s} = \|f\|_{\mathcal{C}^{\lfloor s \rfloor}} + \sup_{x\neq y} \frac{\left| f^{(\lfloor s \rfloor)}(x) - f^{(\lfloor s \rfloor)}(y) \right|}{|x-y|^{\{\!\{s\}\!\}}}, \tag{3.23}$$

where $\lfloor s \rfloor$ denotes the integer part of s and $\{\!\{s\}\!\}$ denotes its fractional part. Note that it roughly corresponds to checking for the fractional Lipschitz condition on a chosen derivative. Hence,

$$\mathcal{C}^s(\mathbb{R}) := \left\{ f \in \mathcal{C}^{\lfloor s \rfloor} : \|f\|_{\mathcal{C}^s} \text{ is well defined and finite} \right\}$$

(iii) A closely related family of spaces are the *Zygmund spaces* $\mathscr{C}^s$ (sometimes called Hölder-Zygmund spaces). These spaces were devised to be able to use a single family of spaces irrespective of the fact that s is or is not an integer. This spaces use the second differences $\Delta_h^2 f^2 = f(x+2h) - 2f(x+h) + f(x)$ instead of the simple difference. Moreover, the decomposition of s into an integer and a fractional part is done slightly different:

$$s = [s] + \{\!\{s\}\!\}^+$$

were $[s] \in \mathbb{Z}$ and $0 < \{\!\{s\}\!\}^+ \leq 1$ (instead of $0 \leq \{\!\{s\}\!\} < 1$). In particular one can write $[s] = -\lfloor 1 - s \rfloor$. Now we have all the ingredients to define the norm of the Zygmund spaces:

$$\|f\|_{\mathscr{C}^s} = \|f\|_{\mathcal{C}^{[s]}} + \sup_{h\neq 0} \frac{\Delta_h^2 f^{([s])}}{h^{\{\!\{s\}\!\}^+}}. \tag{3.24}$$

As before, the space is defined as the set of functions with finite norm.

$$\mathscr{C}^s(\mathbb{R}) := \left\{ f \in \mathcal{C}^{[s]} : \|f\|_{\mathscr{C}^s} \text{ is well defined and finite} \right\}$$

(iv) The *Sobolev spaces* $W_p^m(\mathbb{R})$ come into play when we are we are mostly interested in the norm of the (distributional) derivatives of f:

$$\|f\|_{W_p^m} = \sum_{k\leq m} \left\| f^{(k)} \right\|_{\mathscr{L}^p}. \tag{3.25}$$

Hence,

$$W_p^m(\mathbb{R}) := \left\{ f \in \mathscr{L}^p : \|f\|_{W_p^m} \text{ is well defined and finite} \right\}$$

.

(v) Finally we can introduce the Besov spaces $\mathscr{B}^s_{p,q}(\mathbb{R})$ for $0 < p < \infty$, $0 < q \leq \infty$ and $s > 0$, with norm

$$\|f\|_{\mathscr{B}^s_{p,q}} = \|f\|_{W^{[s]}_p} + \left(\int_{\mathbb{R}} |h|^{-q\{\{s\}\}^+} \left\|\Delta_h^2 f^{([s])}\right\|^q_{\mathscr{L}^p} \frac{dh}{|h|}\right)^{\frac{1}{q}} . \tag{3.26}$$

If $q = \infty$ we change the integral and the q exponents by the maximum over $\mathbb{R}$. Therefore, we can define

$$\mathscr{B}^s_{p,q}(\mathbb{R}) := \left\{ f \in W^{[s]}_p(\mathbb{R}) : \|f\|_{\mathscr{B}^s_{p,q}} \text{ is well defined and finite} \right\}.$$

The derivative in these spaces is in the distributional sense.

Remark 3.25. Note that for all the spaces above we have a parameter (either m for (i) and (iv) or s for (ii), (iii) and (v)) that computes how regular a function in this space is. Hence, they are examples of regularity spaces because through this parameters one can construct spectra on which to measure regularities. In particular one can talk that a function f has Hölder-Zygmund regularity s if $f \in \mathscr{C}^s$, or p, q-Besov regularity s if $f \in \mathscr{B}^s_{p,q}$. ∎

Remark 3.26. Note that the definitions of all the spaces above can be easily changed to be defined on $\mathbb{S}^1$ instead of $\mathbb{R}$. Since the regularity will be measured through the inclusion to Besov spaces, we might want to choose p and q so that our measurements are as refined as possible. In particular, since $\mathbb{S}^1$ is compact, we have a chain of inclusions in the $\mathscr{L}^p$ spaces

$$\mathscr{L}^\infty(\mathbb{S}^1) \subset \ldots \subset \mathscr{L}^m(\mathbb{S}^1) \subset \ldots \subset \mathscr{L}^2(\mathbb{S}^1) \subset \mathscr{L}^1(\mathbb{S}^1)$$

Consequently, when dealing with $\left\|\Delta_h^2 f^{([s])}\right\|_{\mathscr{L}^p}$ inside the integral in Equation 3.26, we have chosen p to be as large as possible. In particular, when we study the more general construction of the Besov spaces, we will see that one can use $p = \infty$ without any problems. ∎

We are well aware that the following proposition is a bit too general for the concepts we have introduced so far. Mostly because it deals with $s \in \mathbb{R}$, not only for $s > 0$. However, we feel that it is a good addition to this more intuitive section. The general definition of $\mathscr{B}^s_{p,q}$ can be find in the next subsection.

Proposition 3.27 (Proposition 2.3.2.2 in **[Tri83]**). *For $p \in \mathbb{N}$ we have:*

(a) Let $0 < q_0 \leq q_1 \leq \infty$ and let $s \in \mathbb{R}$. Then

$$\mathscr{B}^s_{p,q_0} \subset \mathscr{B}^s_{p,q_1}$$

(b) Let q_0, q_1 be any two natural numbers, let $s \in \mathbb{R}$, and let $\varepsilon > 0$. Then

$$\mathscr{B}^{s+\varepsilon}_{p,q_0} \subset \mathscr{B}^s_{p,q_1}$$

What this proposition shows us is that fixing p all Besov spaces are more or less equivalent (being slightly larger as $q \to \infty$). Hence, for simplicity, in the upcoming sections we will end up considering $q = \infty$. In the next subsection we will give a formal definition of the Besov spaces $\mathscr{B}^s_{p,q}$ that is valid for $1 < p, q \leq \infty$ and $s \in \mathbb{R}$.

3.4.2. General definitions. Despite having direct definitions of the Besov spaces $\mathscr{B}^s_{p,q}$ for $s > 0$, $0 < p < \infty$ and $0 < q \leq \infty$, which might help with the intuitive study of a particular function, we will present an equivalent, though much more theoretical definition of these spaces. This definition, moreover, allows us to extend the regularity to $s \leq 0$ and/or $p = \infty$. Finally, this more theoretical construction is crucial when it comes to the study of how the wavelet expansion and regularity relate to one another.

The space of all real valued rapidly decreasing infinitely differentiable functions is called the *(real) Schwartz space* and it is denoted by $\mathcal{S}(\mathbb{R})$. The topological dual of $\mathcal{S}(\mathbb{R})$ is the space of *tempered distributions* which is denoted by $\mathcal{S}'(\mathbb{R})$. As previously stated in this chapter, $f \in \mathcal{S}'(\mathbb{R})$, $\widehat{f}(\xi)$ denotes the *Fourier transform* of f and $f^{\vee}(x)$ stands for the *inverse Fourier transform* in the sense of distributions. Recall, also, that the essential supremum is defined as

$$\operatorname*{ess\,sup}_{x\in\mathbb{R}} f(x) = \inf\{a \in \mathbb{R} : \mu(\{x \in \mathbb{R} : f(x) > a\}) = 0\},$$

where μ is a measure (in our case the usual Lebesgue measure). Finally, let $\varphi_0 \in \mathcal{S}(\mathbb{R})$ be such that

$$\varphi_0(x) := \begin{cases} 1 & \text{if } |x| \leq 1 \\ 0 & \text{if } |x| \geq 3/2 \end{cases}$$

and set

$$\varphi_j(x) := \varphi_0(2^{-j}x) - \varphi_0(2^{-j+1}x)$$

for $j \in \mathbb{N}$. The familiy $\{\varphi_j\}_{j=0}^{\infty}$ is called a *Dyadic Resolution of Unity* in $\mathbb{R}$. Note that we can have multiple dyadic resolutions, depending on how φ_0 is chosen to behave on $1 < |x| < 3/2$.

Definition 3.28. Let $\{\varphi_j\}_{j=0}^{\infty}$ be a Dyadic Resolution of Unity and $s \in \mathbb{R}$. For $f \in \mathcal{S}'(\mathbb{R})$ we define the quasi-norm

$$\|f\|_{p,q,s} := \left(\sum_{j=0}^{\infty} 2^{jsq} \left(\|(\varphi_j \widehat{f})^{\vee}\|_{\mathscr{L}_p} \right)^q \right)^{1/q} .$$

Where $\|\cdot\|_p$ denotes the standard $\mathcal{L}^p$ norm. For the case $q = \infty$ the usual modification of taking the essential supremum instead of powers of q is done. Then, we define the *Generalized Besov Spaces* by

$$\mathscr{B}^s_{p,q}(\mathbb{R}) := \{f \in \mathcal{S}'(\mathbb{R}) \colon \|f\|_{p,q,s} < \infty\}.$$

■

As it can be seen in [**Tri06**, Remark 2 of Section 2.3], the spaces $\mathscr{B}^s_{p,q}(\mathbb{R})$ and the quasi-norm are, in fact, independent of the chosen dyadic resolution of unity φ. However, the choice of the values p and q does give us different spaces.

Remark 3.29. When $p = q = \infty$ it can be proven that $\mathscr{B}^s_{\infty,\infty}(\mathbb{S}^1) = \mathscr{C}^s(\mathbb{S}^1)$ whenever $s > 0$. What the Besov spaces allow is the generalization of the Hl̈der-Zygmund regularity to $s \leq 0$. Of particular interest for us will be the case $s = 0$. ■

In view of the above remark we define the notion of regularity of a map.

Definition 3.30. We say that a map f has *Besov regularity* $s \in \mathbb{R}$ if $f \in \mathscr{B}^s_{\infty,\infty}(\mathbb{R})$, that is,

$$\|f\|_{\infty,\infty,s} = \sup_{j\geq 0} 2^{js} \left(\operatorname*{ess\,sup}_{x\in\mathbb{R}} \left|(\varphi_j \widehat{f})^{\vee}(x)\right| \right) < \infty.$$

■

Recall that our natural framework is the unit circle. As presented in [**AMR16**, Lemma 3.8] Definition 3.30 can be rewritten as follows:

Definition 3.31. We say that a circle map f has *Besov regularity* $s \in \mathbb{R}$ if the map f belongs to $\mathscr{B}^s_{\infty,\infty}(\mathbb{S}^1)$. ■

The next subsection includes a brief note on the limitations of the Besov approach for our study. It is self-evident that a map φ with regularity $s > 1$ is almost everywhere differentiability and hence its graph has very regular geometry. However, we can have very different functions in the geometrical sense that have the exact same regularity when $0 \leq s < 1$. That is why we claim that the regularity is a *flawed* measure of strangeness.

3.4.3. Limitations of the Besov approach. The title of this subsection already points the flawed nature of this approach. This mostly stems that strangeness as in Definition 2.1 is very difficult to quantify, despite its simplicity. Functions that cannot be written as piecewise differential functions range from upper semicontinuous functions as in [**Kel96**] to something like the Weierstrass functions. The situation becomes way more complicated when we consider that the attractor might be a multi-valued function [**Sta99**], which by definition cannot be included into any function space.

In particular, we believe the following two examples illustrate the limitations of the Besov approach.

Example 3.32. We will see that we can get two continuous functions $f, g : \mathbb{S}^1 \to \mathbb{R}$ with the same regularity. The first one is not differentiable at any point, while the second one is differentiable everywhere except on $\theta = 0$. First of all, consider the Weierstrass function

$$f(x) = \sum_{n=1}^{\infty} a^n \cos(b^n x),$$

which has Besov regularity $-\log_b a$ whenever $a \cdot b > 1$ [**Har16**]. On the other hand

$$g(x) = x^{-\log_b a}$$

has Besov regularity also equal to $-\log_b a$. Hence we have two functions f and g with the same regularity. However, f would fall within the definition of strangeness from Chapter 2, while, g would not be considered strange, since it is differentiable everywhere outside 0. A figure of two functions stemming from this example side by side can be seen in Figure 3.3. ∎

Example 3.33. As in the previous example, we will show two functions that have Regularity equal to zero, but that they are have essentially different geometrical complexity. On one hand, an upper semicontinuous and discontinuous in almost any point, like the attractor from the Keller-GOPY model in the pinched case (see Figure 3.4). This type of functions have regularity 0, due to the following result:

Proposition 3.34 (Proposition V.4.6 in [**Ste70**]). *Efery $f \in \mathscr{B}^s_{\infty,\infty}$ with $s \in (0,1)$ may be modified on a set of zero measure so that it becomes continuous.*

On the other let us consider the function

$$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ \frac{1}{\log(x)} & \text{if } 0 < x \leq \frac{1}{2} \\ \frac{-1}{\log(2)} & \text{if } \frac{1}{2} < x \leq 1 \end{cases}, \tag{3.27}$$

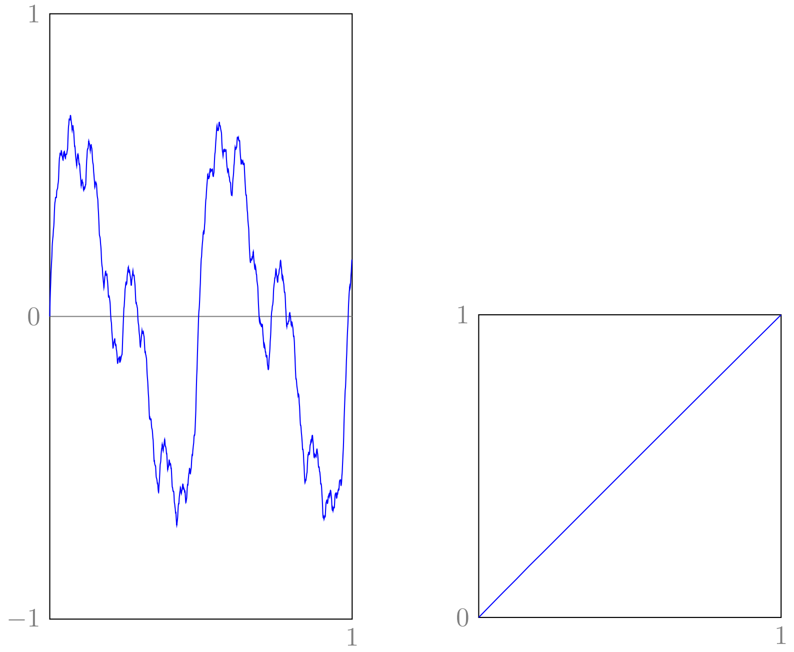
which is continuous (and in fact $\mathcal{C}^\infty$ in $(0, 1/2) \cup (1/2, 1)$) but that has regularity zero. See Figure 3.4 for the graph of both functions.

Let us prove that $f \in \mathscr{B}^0_{\infty,\infty}$ but $f \notin \mathscr{B}^s_{\infty,\infty}$ with $s > 0$. The fact that $f \in \mathscr{B}^0_{\infty,\infty}$ follows directly from the fact that f is continuous. We will prove that $f \notin \mathscr{B}^s_{\infty,\infty}$ with $s > 0$ by contradiction. Suppose that f has positive regularity $\alpha > 0$. Then we would have that there exists $C > 0$ such that

$$|0 - f(x)| = \left|0 - \frac{1}{\log(x)}\right| \leq C|0 - x|^\alpha$$

for $x \in [0, 1]$. This implies that the inequality

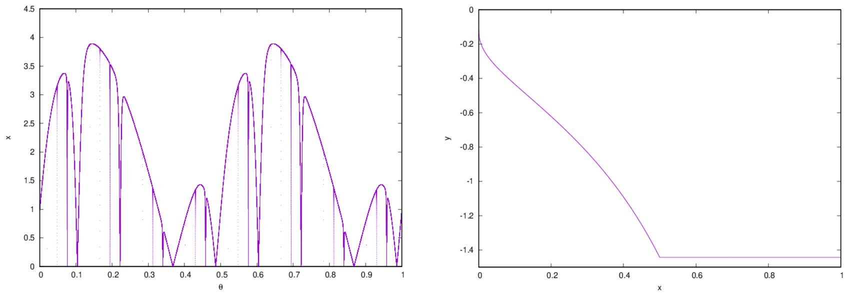
$$C|x|^\alpha |\log(x)| \geq 1,$$



(A) Weierstrass function with $a = 0.5$ and $b = 2.01$.

(B) $f(x) = x^{-\log_{2.01}(0.5)}$ $\approx x^{0.99285590}$

FIGURE 3.3. Two very different functions with Besov regularity $s = -\log_{2.01}(0.5) \approx 0.992855904$. (A) is not differentiable at any point while (B) is differentiable everywhere except at 0.



(A) Keller-GOPY attractor in the pinched case.

(B) Graph of function defined in Equation (3.27).

FIGURE 3.4. Two very different functions with Besov regularity $s = 0$. Note the very complicated geometry in (A) and the relatively regular behaviour of (B).

should hold, which cannot happen since $\lim_{x\to 0^+} |x|^{\alpha}|\log(x)| = 0$ for all $\alpha > 0$. Hence, we have a function that is upper semi-continuous and has very complicated geometry and a function that is differentiable on $(0, 1/2) \cup (1/2, 1)$, both with regularity zero. ∎

Despite these limitations, a crucial point is that if $s > 1$ we have almost everywhere differentiability ensured. Hence, not everything is futile. If we are able to measure regularities bigger than 1 then we can discard strangeness right away.

3.4.4. Wavelet coefficients and regularity. Next, we want to show how this notion of regularity and the wavelet coefficients of a given function are related. This is thoroughly explained in Section 4 of [**AMR16**], but we will present a short summary. As previously mentioned we will be using Daubechies wavelets. The reasoning behind this choice is twofold. On one side, the Daubechies wavelets family are orthonormal bases on $\mathscr{L}^2(\mathbb{R})$. On the other hand, depending on the number of vanishing moments, they are good bases of the functional spaces $\mathscr{B}^s_{\infty,\infty}(\mathbb{R})$. Specifically, given a function φ they allow us to determine the Besov regularity s of φ (see e.g. [**AMR16**] and the references therein). Let us rephrase [**Tri06**, Theorem 1.64] in the spirit of Theorem 4.2 in [**AMR16**] for our particular needs. For $t \in \mathbb{R}$ we set

$$\mathrm{R}(t) = \begin{cases} t - \frac{1}{2} & \text{if } t > \frac{1}{2}, \\ t + \frac{1}{2} & \text{if } t < -\frac{1}{2} \end{cases} \tag{3.28}$$

Using this function we can present the main result concerning wavelet coefficients and regularity in a concise manner.

Theorem 3.35 (Theorem 1.64 [**Tri06**]). *Let $f \in \mathscr{L}^2(\mathbb{S}^1)$ and let ψ be a mother Daubechies wavelet with more than $\max(\mathrm{R}(\tau), \frac{5}{2} - \mathrm{R}(\tau))$ vanishing moments for some $\tau \in \mathbb{R} \setminus \left[-\frac{1}{2}, \frac{1}{2}\right]$ and for each j define $\mathcal{D}_j = \{0, 1, \dots, 2^j - 1\}$. Then, $f \in \mathscr{B}^{\mathrm{R}(\tau)}_{\infty,\infty}(\mathbb{S}^1)$ if and only if there exists $C > 0$ such that*

$$\sup_{n \in \mathcal{D}_j} \left|\left\langle f, \psi^{\mathrm{PER}}_{j,n} \right\rangle\right| \leq C 2^{\tau j}$$

for all $j \leq 0$. Furthermore, if ψ has more than 2 vanishing moments, then $f \in \mathscr{B}^0_{\infty,\infty}(\mathbb{R})$ if and only if either the sequence $\left\{2^{-\tau j} \sup_{n \in \mathcal{D}_j} \left|\left\langle f, \psi^{\mathrm{PER}}_{j,n} \right\rangle\right|\right\}_{j=0}^{-\infty}$ is unbounded for every $\tau \in \mathbb{R}$ or there exist $C > 0$ and $\tau \in \left[-\frac{1}{2}, \frac{1}{2}\right]$ such that

$$\sup_{n \in \mathcal{D}_j} \left|\left\langle f, \psi^{\mathrm{PER}}_{j,n} \right\rangle\right| \leq C 2^{\tau j}$$

for $j \leq 0$.

It is important to notice that the condition on the vanishing moments limits the choice of the wavelets used. As can be seen in Figure 3.5, the more vanishing moments the wavelet has, the bigger range of regularities that can be computed.

Remark 3.36. Note that the crux of Theorem 3.35 is that the regularity lies in the decay of the value of the coefficients that correspond to higher frequencies (i.e. j large). That is, if the maximum of the absolute value of the coefficients in each j-level is large, then the regularity shall be small. This is coherent with our intuition, since if the higher frequencies have a more prominent role, the function φ should be more complicated (at least locally, see Section 4.3 in [**Mal98**]). This is similar to what happens when one computes the Fourier expansion of the square wave. To obtain a good approximation of the signal one needs to compute the coefficients of very high frequencies. In Figures 4.5 and 4.10 one can see an example of this phenomenon. ∎

Remark 3.37. In view of [**AMR16**, Remark 4.3] and the above results, the coefficients $\sup_{n \in \mathcal{D}_j} |\langle f, \psi_{j,n} \rangle|$ decay approximately exponentially

$$s_j := \log_2 \left(\sup_{n \in \mathcal{D}_j} |\langle f, \psi_{j,n} \rangle| \right) \simeq \tau j + \log_2(C). \tag{3.29}$$

Thus to compute an approximation of the Besov regularity s, one can make a regression to estimate the slope τ from the graph of the pairs (j, s_j) and compute s

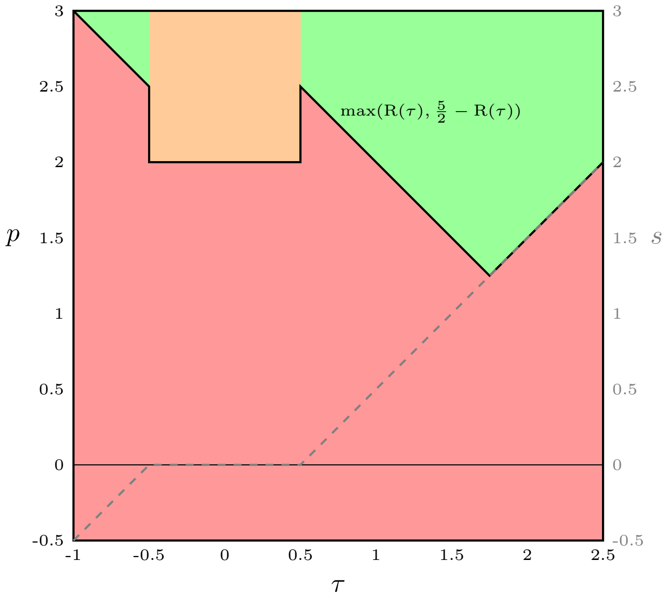


FIGURE 3.5. Relation between s and the number of vanishing moments given by Theorem 3.35, and the regularity computed. In red the areas where Theorem 3.35 cannot be applied. In green where the first part of the theorem migh be applied. The orange area corresponds to values of τ that give regularity zero. The grey line corresponds to the regularity computed as a function of τ.

from the slope of the model. However, note that this computed regularity will never be exact, since Theorem 3.35 has an inequality, while when doing a linear fit, the obtained line is always below some of the (j, s_j). In any case, this has worked rather well for the Keller-GOPY and the Alsedà-Misiurewicz systems. See Figures 4.8 and 4.12 as examples. The situation is rather different for the Nishikawa-Kaneko case, where the linear correlation is non-existent and needs to be studied more carefully (See 4.16).

The value of C plays a role in some transitions to regularity zero. In particular, when the sequence $\left\{2^{-\tau j}\sup_{n\in\mathbb{D}_j}\left|\left\langle f,\psi^{\text{PER}}_{j,n}\right\rangle\right|\right\}_{j=0}^{-\infty}$ is unbounded. This case might happen, for example, when the coefficients corresponding to lower frequencies decay much slower than the ones for higher frequencies. See Figure 4.16 for examples with systems with increasingly larger values of C, despite the fact that their regularity still could be very large. ■

In view of the above remark, we can establish a strategy to use the wavelet coefficients to give an estimate of the regularity of a function. This is done in Section 4 of [**AMR16**]. The following is a rephrase of these techniques but tailoring them for our particular needs.

ALGORITHM 3.38 (Method to estimate regularities). Let $\mathscr{L}^2(\mathbb{S}^1)$ and choose a Daubechies wavelet basis on the circle $\psi^{\text{PER}}_{-j,n}$ with $p \geq 2$ vanishing moments. Consider an approximation of the wavelet expansion with $N = 2^{\nu+1}$ coefficients

$$f = a_0 + \sum_{j=0}^{\nu}\sum_{n=0}^{2^j-1}\left\langle f,\psi^{\text{PER}}_{-j,n}\right\rangle\psi^{\text{PER}}_{-j,n}. \tag{3.30}$$

To estimate the Besov regularity s of the map f, perform the following steps:

Step 1. By using the coefficients $\langle f, \psi^{\mathrm{PER}}_{-j,n}\rangle$ from the given approximation, in view of Theorem 3.35, compute

$$s_{-j} := \log_2 \left(\sup_{0\leq n\leq 2^j-1} \left|\langle f, \psi^{\mathrm{PER}}_{-j,n}\rangle\right| \right)$$

for $j = 0, \ldots, \nu$.

Step 2. In view of equation (3.29), make a linear regression to estimate the slope τ of the graph of the pairs $(-j, s_{-j})$ with $j = 0, \ldots, \nu - 1$. Then, when there is evidence of linear correlation between the variables $-j$ and s_{-j} set $s = \mathrm{R}(\tau)$, defined as in equation (3.28).

Step 3.

- If the number of vanishing moments p for ψ satisfies $p > \max(s, 5/2 - s)$ and $p \geq 2$ then f has regularity s.
- Otherwise we need to repeat Steps1 $-$ 2 with a Daubechies wavelet having more vanishing moments until $p > \max(s, 5/2 - s)$.

■

Remark 3.39. Note that since we are dealing with *truncated* wavelet expansions and doing linear fits, it may make sense to remove some pairs $(-j, s_{-j})$ out of the linear fit, as it was done in [**AMR16**] (even though with the methods presented the disparity is much smaller). In general we have detected s_0, s_{-1} and $s_{-\nu}$ are the ones that fluctuate the most when changing ν. We believe that it stems from the fact that the frequencies not being represented in the expansion get compensated in the very low frequency coefficients (if it oscillates fast enough it may as well look constant) and in the coefficients corresponding to ν. Hence, for the examples in Section 4.3 we have chosen to omit them from the linear correlation computations. ■

In the same way as [**AMR16**, Remark4.5], we want to mention that for our computations we need to make ν as large as possible. Some of the main reasons being:

(a) The truncation error of the approximation (3.30) depends on the regularity of f. Thus, since it is not reasonable to assume that the regularity is known, we need to fix the value of ν as the biggest our computing resources and time can support.
(b) In view of **Step 2** and to increase the reliability of the regressions on ν samples, clearly ν must be as big as possible.
(c) Related with the above point, among all samples $(-j, s_{-j})$ with $j = 0, \ldots, \nu - 1$, one should only use the pairs $(-j, s_{-j})$ where the linear dependence is clearly seen. Thus one has another restriction to the number of samples available and hence $\nu > 0$ must be as large as possible.
(d) As it can be seen in Section 4.4, in the Nishikawa-Kaneko attractor there are two possible linear correlations when considering $(-j, s_{-j})$ for some values of ε . One spans the coefficients corresponding to lower values of j (and hence, lower frequencies). The other one, can be found in the ones corresponding to higher frequencies. If one only considered the correlation spanning the lower frequency coefficients, one would obtain very low values of the regularity. However, the correlation concerning higher frequencies results in very high values of the regularity. Hence, to be sure that the obtained results are acceptable, ν should be very large. If it were too

small, the second correlation might be hidden and hence the regularity computed would not be optimal.

In view of the above, we are ready to compute an estimate of the regularity of a certain function $f \in \mathscr{L}^2(\mathbb{S}^1)$. It only remains to find the truncated wavelet expansion of the attractor φ from Chapter 2. This is, precisely, one of the main topics of the following Chapter, where we will obtain the wavelet coefficients using the invariance equation from Section 2.1.1.

CHAPTER 4

Solving the Invariance Equation with Wavelets

In this chapter we will seek to solve the invariance Equation (2.3),

$$F_{\sigma,\varepsilon}(\theta, \varphi(\theta)) = \varphi\big(R_\omega(\theta)\big),$$

using wavelets. Recall that what we are trying to do is to find a map $\varphi : \mathbb{S}^1 \to \mathbb{R}$ such that its graph is the attractor of the system. As we discussed in Chapter 2 this is equivalent to φ being a solution to the invariance equation which is a functional equation in which its fixed points are invariant graphs of System (2.1). Moreover, recall that Theorem 3.19 tells us that the so called periodized wavelet basis for $\mathscr{L}^2(\mathbb{S}^1)$ has the peculiarity that is of the form

$$\{1, \psi^{\text{PER}}_{j,n} \text{ with } j \leq 0 \text{ and } n = 0, 1, \ldots, 2^{-j} - 1\}.$$

That is, fixing $j > 0$ the number of different functions $\psi^{\text{PER}}_{-j,n}$ is finite. Hence, we will seek to solve the invariance equation by finding coefficients $d_{-j,n} \in \mathbb{R}$ such that

$$\varphi(\theta) = d_0 + \sum_{j=0}^{\infty} \sum_{n=0}^{2^j-1} d_{-j,n} \psi^{\text{PER}}_{-j,n}(\theta),$$

where $d_{-j,n} = \big\langle \varphi, \psi^{\text{PER}}_{-j,n} \big\rangle$, that is

$$d_{-j,n} = \int_{\mathbb{S}^1} \varphi(\theta) \psi^{\text{PER}}_{-j,n}(\theta) d\theta.$$

It comes without saying that it is impossible to find infinitely many wavelet coefficients numerically, so we will truncate the expression to get a *truncated wavelet expansion*. The natural way of doing this is to fix a maximum value of j, that we shall call ν, and then obtaining all the coefficients for $j = 0, \ldots, \nu$ and, for each j, for $n = 0, 1, \ldots, 2^j - 1$. In other words, we want to obtain an expression as

$$\breve{\varphi}(\theta) := a_0 + \sum_{j=0}^{\nu} \sum_{n=0}^{2^j-1} d_{-j,n} \psi^{\text{PER}}_{-j,n}(\theta). \tag{4.1}$$

Note that in this case the expansion has exactly $2^{\nu+1}$ elements.

Finally, we can state the goal of this chapter. We want to find the coefficients $\{d_{-j,n}\}_{j=0}^{\nu}$ such that

$$\left| F_{\sigma,\varepsilon}\left(\theta, a_0 + \sum_{j=0}^{\nu} \sum_{n=0}^{2^j-1} d_{-j,n} \psi^{\text{PER}}_{-j,n}(\theta)\right) - \varphi\left(a_0 + \sum_{j=0}^{\nu} \sum_{n=0}^{2^j-1} d_{-j,n} \psi^{\text{PER}}_{-j,n}\big(R_\omega(\theta)\big)\right)\right| < \texttt{tol}, \tag{4.2}$$

for every $\theta \in \mathbb{S}^1$, with `tol` being a reasonable error chosen beforehand.

Note that this procedure can eventually lead to the reconstruction of the attractor without having to compute it by forward iteration, as it is otherwise usual for numerical studies of attractors. This in principle might allow an improved study of attractors with very slow convergence, or the study of bifurcations in which the attractor might dissappear but the invariant function might not.

This chapter will start with a section in which we will give a detailed account of how we have solved Equation 4.2 using Newton's method.

4.1. Setting up Newton

After the the dynamical explanation of the problem in Chapter 2 and the more or less analysis-centric wavelet introduction in Chapter 3, we will use the results of both sections to produce an algorithm that will allow us to compute truncated wavelet expansions of functions such that their graphs $\big(\theta, \varphi(\theta)\big)$ are attractors of System (2.1).

Remark 4.1. To de-clutter the indices, from now on we will work with the single index $\ell = \ell(j,n) = 2^j + n$. Hence, Equation (4.2) now can be rewritten as

$$\breve{\varphi}(\theta) = a_0 + \sum_{j=0}^{\nu} \sum_{n=0}^{2^j-1} d_{-j,n} \psi_{-j,n}^{\text{PER}}(\theta) = d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\text{PER}}(\theta), \tag{4.3}$$

where $N = 2^{\nu+1}$. More precisely, we can consider the coefficients d_0 and $\{d_\ell\}_\ell$ as the unknowns of the functional equation (4.2), obtaining

$$d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\text{PER}}(R_\omega(\theta)) - F_{\sigma,\varepsilon}\left(\theta, d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\text{PER}}(\theta)\right) \approx 0. \tag{4.4}$$

Hence, we will strive to solve N dimensional vector of unknowns

$$\begin{aligned} \mathbf{D} :&= (d_0, d_1, \dots, d_{N-1}) \\ &= (d_0, d_{0,0}, \dots, d_{-\nu, 2^\nu - 1}) \quad \text{(in the bi-indexed notation)}. \end{aligned} \tag{4.5}$$

■

Since our goal is to find $\mathbf{D}$, the straightforward approach would be to obtain a non-linear system of N independent equations with $\mathbf{D}$ as the vector of unknowns, and solve it directly by using Newton's method. This approach is quite standard when one is dealing with the Fourier basis, for example in celestial mechanics. That is, if we fix θ, Equation 4.4 gives us one equation with N unknowns $d_0, d_1, \dots d_{N-1}$. Hence, if we can get N such equations, we should be able to find a solution for $\mathbf{D}$. As it is usual, we will discretize θ into N dyadic points $\theta_i = {}^i\!/\!{}_N \in \mathbb{S}^1$. For each θ_i we have then an instance of Equation (4.4). Hence, by defining the function $\mathbf{F}_{\sigma,\varepsilon}(\mathbf{D}) \colon \mathbb{R}^N \longrightarrow \mathbb{R}^N$

$$\mathbf{F}_{\sigma,\varepsilon}(\mathbf{D}) := \tag{4.6}$$
$$\begin{pmatrix} d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\text{PER}}(R_\omega(\theta_0)) - F_{\sigma,\varepsilon}\left(\theta_0, d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\text{PER}}(\theta_0)\right) \\ d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\text{PER}}(R_\omega(\theta_1)) - F_{\sigma,\varepsilon}\left(\theta_1, d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\text{PER}}(\theta_1)\right) \\ \vdots \\ d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\text{PER}}(R_\omega(\theta_{N-1})) - F_{\sigma,\varepsilon}\left(\theta_{N-1}, d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\text{PER}}(\theta_{N-1})\right) \end{pmatrix},$$

we have converted the problem of solving the invariance equation using wavelets into a root finding one. Namely we want to solve

$$\mathbf{F}_{\sigma,\varepsilon}(\widetilde{\mathbf{D}}) \approx \vec{0} \in \mathbb{R}^N,$$

to obtain an approximation $\widetilde{\mathbf{D}}$ of $\mathbf{D}$.

Hence, applying Newton's method at each step we want to compute

$$\mathbf{D}_{\mathsf{n}+1} = \mathbf{D}_{\mathsf{n}} - \mathbf{J}\mathbf{F}_{\sigma,\varepsilon}(\mathbf{D}_{\mathsf{n}})^{-1} \mathbf{F}_{\sigma,\varepsilon}(\mathbf{D}_{\mathsf{n}}), \tag{4.7}$$

where $\mathbf{D}_n$ is the nth approximation of $\mathbf{D}$ by the method. To avoid inverting $\mathbf{JF}(\mathbf{D}_n)$, is is standard to change the iteration to solving the following linear system system of equations

$$\mathbf{JF}_{\sigma,\varepsilon}(\mathbf{D}_\mathsf{n})\big(\mathbf{D}_{\mathsf{n}+1}-\mathbf{D}_\mathsf{n}\big) = -\mathbf{F}_{\sigma,\varepsilon}(\mathbf{D}_\mathsf{n}). \tag{4.8}$$

for $(\mathbf{D}_{\mathsf{n}+1}-\mathbf{D}_\mathsf{n})$ at each Newton step .

At it is evident from Equation (4.6), a direct implementation of this path leads us to evaluating periodized Daubechies wavelets $\psi^{\text{PER}}_{-j,n}(\theta)$ at each newton step. As explained in Subsection 3.2 this is not an easy task. In fact Chapter 5 of this thesis is dedicated solely to this problem. Hence another approach is necessary.

To this end, we will use linear algebra and some cleverly defined matrices that will allow us to precompute all the wavelet evaluations that we need and store them in an ordered manner. The main motivation behind the construction of these matrices is the realization that the truncated wavelet expansion evaluated at θ can be written as an scalar product of two vectors

$$\breve{\varphi}(\theta) = d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi^{\text{PER}}_\ell(\theta) = \big(1, \psi^{\text{PER}}_1(\theta), \psi^{\text{PER}}_2(\theta), \dots, \psi^{\text{PER}}_{N-1}(\theta)\big) \cdot \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{N-1}. \end{pmatrix} \tag{4.9}$$

An analogous result can be obtained for the truncated expression for $\breve{\varphi}(R_\omega(\theta))$. From Equation (4.9) we can see that instead of evaluating periodized wavelets at each Newton step, we can pre-compute them and use linear algebra whenever we need to use the evaluation. This is one of the crucial points of this method, since it allows us to orderly store and use the value of a wavelet evaluated at a point whenever we might need it. To exploit this properties, we have defined to matrices Ψ and Ψ_R where all the needed $\psi^{\text{PER}}_\ell(\theta_i)$ and $\psi^{\text{PER}}_\ell(R_\omega(\theta_i))$ will be stored.

Definition 4.2. Let $\psi(x)$ be an orthonormal wavelet from a $\mathbb{R}$-MRA:

(a) We define the *wavelet matrix* Ψ as a square matrix whose column 0 is constant 1 and the column $\ell \in \{1, 2, \dots, N-1\}$ is $\psi^{\text{PER}}_\ell(\theta_i)$ with $i = 0, 1, \dots, N-1$. Equivalently, the ith row of the column will be of the form

$$\big(1, \psi^{\text{PER}}_1(\theta_i), \psi^{\text{PER}}_2(\theta_i), \dots, \psi^{\text{PER}}_{N-1}(\theta_i)\big)$$

(b) We define the *rotated wavelet matrix* Ψ_R as a square matrix whose column 0 is constant 1 and the column $\ell \in \{1, 2, \dots, N-1\}$ is $\psi^{\text{PER}}_\ell(R_\omega(\theta_i))$ with $i = 0, 1, \dots, N-1$. Equivalently, the ith row of the column will be of the form

$$\big(1, \psi^{\text{PER}}_1(R_\omega(\theta_i)), \psi^{\text{PER}}_2(R_\omega(\theta_i)), \dots, \psi^{\text{PER}}_{N-1}(R_\omega(\theta_i))\big)$$

■

Remark 4.3. Let Ψ be the wavelet matrix for a Daubechies wavelet of p vanishing moments. We can write the block of size $N \times 2^j$ corresponding to the wavelets $\psi^{\text{PER}}_{-j,n}$ with the same j as

$$\begin{pmatrix}
\psi^{\text{PER}}_{-j,0}(0) & \psi^{\text{PER}}_{-j,1}(0) & \psi^{\text{PER}}_{-j,3}(0) & \cdots & \psi^{\text{PER}}_{-j,2^j-1}(0) \\
\psi^{\text{PER}}_{-j,0}\left(\frac{1}{N}\right) & \psi^{\text{PER}}_{-j,1}\left(\frac{1}{N}\right) & \psi^{\text{PER}}_{-j,3}\left(\frac{1}{N}\right) & \cdots & \psi^{\text{PER}}_{-j,2^j-1}\left(\frac{1}{N}\right) \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\psi^{\text{PER}}_{-j,0}\left(\frac{N-2}{N}\right) & \psi^{\text{PER}}_{-j,1}\left(\frac{N-2}{N}\right) & \psi^{\text{PER}}_{-j,3}\left(\frac{N-2}{N}\right) & \cdots & \psi^{\text{PER}}_{-j,2^j-1}\left(\frac{N-2}{N}\right) \\
\psi^{\text{PER}}_{-j,0}\left(\frac{N-1}{N}\right) & \psi^{\text{PER}}_{-j,1}\left(\frac{N-1}{N}\right) & \psi^{\text{PER}}_{-j,3}\left(\frac{N-1}{N}\right) & \cdots & \psi^{\text{PER}}_{-j,2^j-1}\left(\frac{N-1}{N}\right)
\end{pmatrix}.$$

For the matrix Ψ_R the result is completely analogous. As we will see later we can apply Proposition 3.22 to find self-similarities within these types of blocks to reduce the amount of memory required to store Ψ and Ψ_R. ∎

REMARK 4.4. Using the matrices defined above, we obtain

$$\Psi \cdot \mathbf{D} = \begin{pmatrix} d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\mathrm{PER}}(0) \\ d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\mathrm{PER}}\left(\frac{1}{N}\right) \\ \cdots \\ d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\mathrm{PER}}\left(\frac{N-1}{N}\right) \end{pmatrix} = \begin{pmatrix} \breve{\varphi}(0) \\ \breve{\varphi}\left(\frac{1}{N}\right) \\ \cdots \\ \breve{\varphi}\left(\frac{N-1}{N}\right) \end{pmatrix} =: \mathbf{\Phi}. \tag{4.10}$$

The vector $\mathbf{\Phi}$ contains the approximation of the attractor evaluated at all the dyadic points. Similarly,

$$\Psi \cdot \mathbf{D} = \begin{pmatrix} d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\mathrm{PER}}(R_\omega(0)) \\ d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\mathrm{PER}}\Big(R_\omega\left(\frac{1}{N}\right)\Big) \\ \cdots \\ d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\mathrm{PER}}\Big(R_\omega\left(\frac{N-1}{N}\right)\Big) \end{pmatrix} = \begin{pmatrix} \breve{\varphi}(R_\omega(0)) \\ \breve{\varphi}\Big(R_\omega\left(\frac{1}{N}\right)\Big) \\ \cdots \\ \breve{\varphi}\Big(R_\omega\left(\frac{N-1}{N}\right)\Big) \end{pmatrix} =: \mathbf{\Phi}_R. \tag{4.11}$$

Form the vectors $\mathbf{\Phi}$ and $\mathbf{\Phi}_R$ we can recover the graph of the attractor. We will use this property in Sections 4.3 and 4.4 to check the quality of the results obtained. ∎

Our goal now is to write Equation 4.8 using Ψ and Ψ_R. Let us start by defining the vector $\wp \in \mathbb{R}^N$

$$[\wp]_i := F_{\sigma,\varepsilon}\left(\theta_i, [\Psi\mathbf{D}]_i\right).$$

where $[\Psi\mathbf{D}]_i$ is the $i-$ th component of the vector $\Psi \cdot \mathbf{D} \in \mathbb{R}^N$ and $[\Psi_R\mathbf{D}]_i$ is the $i-$ th component of the vector $\Psi_R \cdot \mathbf{D} \in \mathbb{R}^N$, respectively.

In view of the above definitions and remark, we can re-write Equation 4.6 to obtain an expression $\mathbf{F}_{\sigma,\varepsilon}(\mathbf{D})$ using the wavelet matrices:

$$\mathbf{F}_{\sigma,\varepsilon}(\mathbf{D}) = \Psi_R\mathbf{D} - \wp. \tag{4.12}$$

This equation give us the invariance equation evaluated at all θ_i at once. From here it follows easily that the Jacobian Matrix of this systems can be written as

$$\mathbf{JF}_{\sigma,\varepsilon} = \Psi_R - \mathit{\Delta}_{\sigma,\varepsilon}\Psi,$$

where $\mathit{\Delta}_{\sigma,\varepsilon}$ is the *derivative matrix* defined as a diagonal square matrix given by

$$(\mathit{\Delta}_{\sigma,\varepsilon})_{i,\ell} = \begin{cases} \mathrm{D}_x F_{\sigma,\varepsilon}\left(\theta_i, [\Psi\mathbf{D}]_i\right) & \text{if } i = \ell, \\ 0 & \text{otherwise.} \end{cases}$$

If rewrite Equation (4.8) using the wavelet matrices we obtain

$$\wp_{\mathsf{n}} - \Psi_R\mathbf{D}_{\mathsf{n}} = -\mathbf{F}_{\sigma,\varepsilon}(\mathbf{D}_{\mathsf{n}}) = \mathbf{JF}_{\sigma,\varepsilon}(\mathbf{D}_{\mathsf{n}})(\mathrm{X}_{\mathsf{n}}) = (\Psi_R - \mathit{\Delta}_{\sigma,\varepsilon}^{(\mathsf{n})}\Psi)\mathrm{X}_{\mathsf{n}},$$

where $\mathit{\Delta}_{\sigma,\varepsilon}^{(\mathsf{n})}$ stands for the derivative matrix at each step, and $\mathrm{X}_{\mathsf{n}} := \mathbf{D}_{\mathsf{n}} - \mathbf{D}_{\mathsf{n}-1}$ stands for the unknown at each Newton step. Therefore, at each Newton iterate we have to solve the following $N \times N$ linear system of equations

$$\big(\Psi_R - \mathit{\Delta}_{\sigma,\varepsilon}^{(\mathsf{n})}\Psi\big)\mathrm{X}_{\mathsf{n}} = \wp_{\mathsf{n}} - \Psi_R. \tag{4.13}$$

The above algebraic form implies that the matrices Ψ and Ψ_R only need to be (pre)computed once in the whole iteration. For example, all the computations in Sections 4.3 and 4.4 have been done with the same pre-computed wavelet matrices stored in a binary file.

To conclude the rephrasing of the problem we need to explain how one can get the initial seed $\mathbf{D}_{\mathsf{0}}$. In this cases, it is usual to use the Birkhoff Ergodic Theorem (see Theorem 2.5). Indeed, assuming that the dynamical system given by Equation (2.1) verifies that $\varphi(\cdot)\psi^{\text{PER}}(\cdot) \in \mathscr{L}^1(\mathbb{S}^1)$ then, by the Birkhoff Ergodic Theorem, it follows that

$$d_\ell := \int_{\operatorname{supp}\left(\psi_\ell^{\text{PER}}\right)} \varphi(\theta)\psi_\ell^{\text{PER}}(\theta)\, d\theta = \lim_{k\to\infty} \frac{1}{k} \sum_{i=0}^{k-1} \overbrace{\varphi \cdot \psi_\ell^{\text{PER}} \cdot \chi_{\operatorname{supp}\left(\psi_\ell^{\text{PER}}\right)}}^{f \text{ of Birk. Erg. Thm.}} (R_\omega^i(\theta)),$$

where $\chi_{\operatorname{supp}\psi_\ell^{\text{PER}}}$ is the characteristic function for the support of ψ_ℓ^{PER}. However, the convergence of this method is very slow, and it requires a lot of evaluations on irrational points, which defeats the purpose of the method. However, it turns out that the trapezoidal rule because gives us a good seed. That is, taking $k = N$ our initial seed will be given by

$$\left[\mathbf{D}_0\right]_\ell := \frac{1}{N}\sum_{i=0}^{N} \psi_\ell^{\text{PER}}(\theta_i)\varphi(\theta_i).$$

Note that since the function is defined on $\mathbb{S}^1$ the formula is slightly different than the trapezoidal rule on the interval, Now, we can write the initial guess in matricial form as

$$\mathbf{D}_0 := \frac{1}{N}\Psi^\top\big(\varphi(\theta_0), \varphi(\theta_1), \dots, \varphi(\theta_{N-1})\big)^\top = \frac{1}{N}\Psi^\top\Phi^\top, \tag{4.14}$$

where we the N-dimensional vector $\Phi = \big(\varphi(\theta_0), \varphi(\theta_1), \dots, \varphi(\theta_{N-1})\big)$ is as in Equation 4.10. Thus, we need to compute Φ with *precision*. This can only be done in a $F_{\sigma,\varepsilon}$-basis, since small changes on σ or ε might lead to very different attractors. At the end of Chapter 2, Algorithm 2.8 allows us to compute the Lyapunov exponent on the attractor. Using this, we present a method to compute Φ with a prescribed precision.

Algorithm 4.5 (A sample of an SNA). Consider a skew product, given by Equation (2.1) and assume that has a hyperbolic attractor $\mathcal{A} \not\equiv 0$. To get a sample of
$mathcalA = \{(\theta, \varphi(\theta)) : \theta \in \mathbb{S}^1\}$ with precision at least δ perform the following steps:

Step 1. Fix $\nu > 0$ big enough and a tolerance δ small enough. Set $N = 2^{\nu+1}$ and take $\theta_i = {}^i\!/\!{}_N \in \mathbb{S}^1$ for $i = 0, \dots, N-1$.

Step 2. Compute λ_φ using Algorithm 2.8. Use λ_φ to compute the minimum number of iterates by

$$\eta \approx \frac{\log\left(\frac{\delta}{\delta_0}\right)}{\lambda_\varphi},$$

where δ_0 denotes the initial error, which may be unknown. To solve that issue and to ensure that so that possible errors on λ_ϕ will not interfere with the computation we will choose the number of iterates to be

$$n_0 := \mathsf{T} \cdot \frac{\log\left(\frac{\delta}{\delta_{\text{MAX}}}\right)}{\lambda_\varphi} \tag{4.15}$$

where $T \geq 1$ is a multiplicative factor to ensure that $n_0 \geq \eta$. In our case we have set $\mathsf{T} = 10$. Moreover, since we do not know the value of δ_0, we simply use an upper bound δ_{MAX} that will depend on each system.

Step 3. Using such n_0, compute the preimage of each $\theta_i \in \mathbb{S}^1$ by the rotation R_ω. That is, for $i = 0, \dots, N-1$ set

$$\widetilde{\theta}_i = R_\omega^{-n_0}(\theta_i).$$

Step 4. For each $i = 0, \ldots, N-1$ iterate forwards n_0 times the System (2.1) using such $\widetilde{\theta}_i$'s and $x_0 = c > \sup_{x\in\mathbb{R}} \max_{\theta\in[0,1]} F_{\sigma,\varepsilon}(\theta, x)$. ∎

Clearly, the output of the above algorithm is the following vector

$$\Phi = (\varphi(\theta_0), \varphi(\theta_1), \ldots, \varphi(\theta_{N-1}))^\top \tag{4.16}$$

which is at least a δ-close sample of φ evaluated at *our desired* points $\theta_i \in \mathbb{S}^1$, with $i = 0, 1, \ldots, N-1$.

Remark 4.6. For the cases we are going to study, we consider a uniformly distributed mesh in $\mathbb{S}^1$. However, one might as well choose a non-uniformly distributed mesh. In other words the choice of points in **Step 1** of the following algorithm can be done arbitrarily. The problem is that this choice has consequences, which must be taken into account further down the line.
Moreover, modifying choice will mean that the wavelet matrices Ψ and Ψ_R in Chapters 4 and 5 would have to be rewritten accordingly. ∎

Remark 4.7. In [**AMR16**] there is another strategy to compute such a Φ with precision. The idea is the use of Theorem 2.6(c) and an *explicit* diffeomorphism $h\colon \mathbb{S}^1 \longrightarrow \mathbb{S}^1$ (which is a $\mathcal{C}^2$ degree one rational quadratic monotone spline on the circle). In this thesis we have used another method because we explicitly need φ evaluated on a mesh of points. ∎

Finally we are able to give what is without any doubt the cornerstone of this chapter (and probably of this part of the thesis). The following algorithm allows us to compute the wavelet expansion of an attractor given by the graph of a function $\varphi : \mathbb{S}^1 \to \mathbb{R}$ in a quasi-periodically forced skew product.

Algorithm 4.8 (*Numerical Computation of an SNA with Wavelets*). Consider an skew product, given by Equation (2.1) with an invariant map $\varphi\colon \mathbb{S}^1 \longrightarrow \mathbb{R}$. That is, there exists a map φ such that for all $\theta \in \mathbb{S}^1$

$$F_{\sigma,\varepsilon}(\theta, \varphi(\theta)) = \varphi(R_\omega(\theta)).$$

Moreover, assume that we can evaluate $\varphi(\theta_i)$ with a desired precision. Fix ψ a Daubechies wavelet with p vanishing moments. To compute an approximation of an invariant curve of System (2.1) using a truncated Daubechies wavelet expression with N coefficients, perform the following steps:

Step 1. Fix an integer $\nu > 0$ as big as possible and two tolerances that will cause the Newton loop to break:

- `tol`, corresponding to finding an approximate solution to the Invariance Equation (4.2)
- `tol_zero`, corresponding to having two consecutive iterates very close, i.e. $\|\mathbf{D}_\mathsf{n} - \mathbf{D}_{\mathsf{n}-1}\| <$ `tol_zero` would trigger a stop in the loop. `tol_zero` should be significantly smaller that `tol`

Set $N = 2^{\nu+1}$ as the number of coefficients of the approximation. Fix a $\|\cdot\|$ on $\mathbb{R}^N$.

Step 2. Compute the $N \times N$ matrices Ψ_R and Ψ.

Step 3. To get a reasonable seed for Newton's Method, use Equation (4.14). That is, set

$$\mathbf{D}_0 = \frac{1}{N}\Psi^\top\Phi^\top,$$

where Φ is given by as the result of Algorithm 4.5. Hence, we can ensure we have control on the computation error of $\varphi(\theta_i)$, so this approximation is achievable.

Step 4. Find $\mathbf{D}_\mathsf{n} \in \mathbb{R}^N$ using a numerical solver to find the solution to the equation

$$\big(\Psi_R - \mathit{\Delta}^{(\mathsf{n})}_{\sigma,\varepsilon}\Psi\big)\mathrm{X}_\mathsf{n} = \wp_\mathsf{n} - \Psi_R,$$

where $\mathrm{X}_{\mathsf{n}} = \mathbf{D}_{\mathsf{n}} - \mathbf{D}_{\mathsf{n}-1}$ until either $\|\mathbf{D}_{\mathsf{n}} - \mathbf{D}_{\mathsf{n}-1}\| <$ `tol_zero` or $\|\wp_{\mathsf{n}} - \Psi_R\| <$ `tol`.

As a result, we get

$$\varphi(\theta) \approx d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell^{\mathrm{PER}}(\theta).$$

That is a numerical approximation, up to a precision `tol`, of an invariant object with wavelets.

Remark 4.9. Note that in this algorithm converge almost all the results introduced until now. From the understanding of the dynamical problem to the theory of wavelets. Moreover, all the constructions in this chapter are what allows the algorithm to be efficient and as simple as possible. Two main issues are left to be tackled though: solving Equation (4.13) in **Step 4**, and the computation of the matrices Ψ and Ψ_R. The first issue will be the subject of Section 4.2. As for the latter, the longest chapter of this thesis, Chapter 5, is solely dedicated to the computation of this matrices. ∎

Before starting with the numerical methods that will allow us to solve Equation (4.13) at each step of the algorithm, we would like to give some general properties of the wavelet matrices that show that if one understands their structure one can realize that they contain a lot of redundant information.

4.1.1. Some properties of the Wavelet Matrices. Recall that Ψ and Ψ_R have been defined in Definition 4.2 and contain all the $\psi_\ell^{\mathrm{PER}}(\theta_i)$ and $\psi_\ell^{\mathrm{PER}}(R_\omega(\theta_i))$ we require for the computations in Algorithm 4.8. Since they are square matrices of dimension $N \times N$, if all the coefficients were to be stored it would require a lot of memory. For example, if we were to use 2^{25} coefficients, one would need a matrix of size 2^{50}. Numerically, if we were to store every single matrix coefficient in `double` precision (8 bytes), we would require 8 petabytes of memory, which no single storage unit in the market can store, let alone finding any computer with the RAM memory capabilities in order to do anything remotely useful with it. However, there are two properties shared by both the wavelet matrix and the rotated wavelet matrix that allow us to save all the essential information while using less than $N \log_2(N)$ entries.

Remark 4.10. Let Ψ and Ψ_R be the wavelet matrix and the rotated wavelet matrix for a Daubechies wavelet of p vanishing moments. If we study the block of size $N \times 2^j$ as the one in Remark 4.3 we can apply Proposition 3.22 to see that the information contained in the first column is exactly the same that the information contained in any other column. The only difference is that they are shifted. For example, given two consecutive columns in the same j-block we can understand the situation as follows:

$$\begin{pmatrix}
\psi^{\mathrm{PER}}_{-j,n}(0) & \psi^{\mathrm{PER}}_{-j,n+1}(0) \\
\psi^{\mathrm{PER}}_{-j,n}\left(\frac{1}{N}\right) & \psi^{\mathrm{PER}}_{-j,n+1}\left(\frac{1}{N}\right) \\
\vdots & \vdots \\
\psi^{\mathrm{PER}}_{-j,n}\left(\frac{2^{\nu-j}-1}{N}\right) & \psi^{\mathrm{PER}}_{-j,n+1}\left(\frac{2^{\nu-j}-1}{N}\right) \\
\psi^{\mathrm{PER}}_{-j,n}\left(\frac{1}{2^j}\right) & \psi^{\mathrm{PER}}_{-j,n+1}\left(\frac{1}{2^j}\right) \\
\vdots & \vdots \\
\psi^{\mathrm{PER}}_{-j,n}\left(\frac{2\cdot 2^{\nu-j}-1}{N}\right) & \psi^{\mathrm{PER}}_{-j,n+1}\left(\frac{2\cdot 2^{\nu-j}-1}{N}\right) \\
\psi^{\mathrm{PER}}_{-j,n}\left(\frac{2}{2^j}\right) & \psi^{\mathrm{PER}}_{-j,n+1}\left(\frac{2}{2^j}\right) \\
\vdots & \vdots \\
\vdots & \vdots \\
\psi^{\mathrm{PER}}_{-j,n}\left(\frac{N-2^j}{N}\right) & \psi^{\mathrm{PER}}_{-j,n+1}\left(\frac{N-2^j}{N}\right) \\
\vdots & \vdots \\
\psi^{\mathrm{PER}}_{-j,n}\left(\frac{N-1}{N}\right) & \psi^{\mathrm{PER}}_{-j,n+1}\left(\frac{N-1}{N}\right)
\end{pmatrix}$$

In this case, we the colour coding indicates that the entries of the matrix are the same. ∎

In the following two pages, we have put examples of Ψ and Ψ_R with $N = 16$, marking their self-similarities with colours.

EXAMPLE 4.11. An example of a Wavelet Matrix in which we show the self-similarities of Ψ for a Daubechies wavelet with 10 vanishing moments and $\nu = 4$

$$\begin{pmatrix}
1 & -0.6125 & 0.4796 & -1.3458 & 1.1989 & -0.4557 & -0.5207 & -1.4475 & 1.5930 & -0.4988 & 0.2737 & -0.1833 & 0.1025 & -0.1457 & -1.0100 & -1.8638 \\
1 & -0.0910 & 1.3228 & -0.6709 & 1.7728 & -0.9019 & 0.8394 & 0.8712 & 0.3495 & -1.3670 & 0.8378 & -0.4019 & 0.1360 & -0.0141 & 0.4092 & 1.7831 \\
1 & 0.4609 & 1.8471 & -0.0217 & 0.3433 & -0.9766 & 0.8817 & 0.9766 & -1.8638 & 1.5930 & -0.4988 & 0.2737 & -0.1833 & 0.1025 & -0.1457 & -1.0100 \\
1 & 0.9572 & 1.6721 & 0.1968 & -1.8368 & -0.1393 & 0.0851 & -0.6906 & 1.7831 & 0.3495 & -1.3670 & 0.8378 & -0.4019 & 0.1360 & -0.0141 & 0.4092 \\
1 & 1.2907 & 0.8662 & 0.0000 & -1.4475 & 1.1989 & -0.4557 & -0.5207 & -1.0100 & -1.8638 & 1.5930 & -0.4988 & 0.2737 & -0.1833 & 0.1025 & -0.1457 \\
1 & 1.4098 & -0.2615 & -0.3903 & 0.8712 & 1.7728 & -0.9019 & 0.8394 & 0.4092 & 1.7831 & 0.3495 & -1.3670 & 0.8378 & -0.4019 & 0.1360 & -0.0141 \\
1 & 1.3215 & -1.2386 & -0.5868 & 0.9766 & 0.3433 & -0.9766 & 0.8817 & -0.1457 & -1.0100 & -1.8638 & 1.5930 & -0.4988 & 0.2737 & -0.1833 & 0.1025 \\
1 & 1.0432 & -1.6113 & -0.2576 & -0.6906 & -1.8368 & -0.1393 & 0.0851 & -0.0141 & 0.4092 & 1.7831 & 0.3495 & -1.3670 & 0.8378 & -0.4019 & 0.1360 \\
1 & 0.6125 & -1.3458 & 0.4796 & -0.5207 & -1.4475 & 1.1989 & -0.4557 & 0.1025 & -0.1457 & -1.0100 & -1.8638 & 1.5930 & -0.4988 & 0.2737 & -0.1833 \\
1 & 0.0910 & -0.6709 & 1.3228 & 0.8394 & 0.8712 & 1.7728 & -0.9019 & 0.1360 & -0.0141 & 0.4092 & 1.7831 & 0.3495 & -1.3670 & 0.8378 & -0.4019 \\
1 & -0.4609 & -0.0217 & 1.8471 & 0.8817 & 0.9766 & 0.3433 & -0.9766 & -0.1833 & 0.1025 & -0.1457 & -1.0100 & -1.8638 & 1.5930 & -0.4988 & 0.2737 \\
1 & -0.9572 & 0.1968 & 1.6721 & 0.0851 & -0.6906 & -1.8368 & -0.1393 & -0.4019 & 0.1360 & -0.0141 & 0.4092 & 1.7831 & 0.3495 & -1.3670 & 0.8378 \\
1 & -1.2907 & 0.0000 & 0.8662 & -0.4557 & -0.5207 & -1.4475 & 1.1989 & 0.2737 & -0.1833 & 0.1025 & -0.1457 & -1.0100 & -1.8638 & 1.5930 & -0.4988 \\
1 & -1.4098 & -0.3903 & -0.2615 & -0.9019 & 0.8394 & 0.8712 & 1.7728 & 0.8378 & -0.4019 & 0.1360 & -0.0141 & 0.4092 & 1.7831 & 0.3495 & -1.3670 \\
1 & -1.3215 & -0.5868 & -1.2386 & -0.9766 & 0.8817 & 0.9766 & 0.3433 & -0.4988 & 0.2737 & -0.1833 & 0.1025 & -0.1457 & -1.0100 & -1.8638 & 1.5930 \\
1 & -1.0432 & -0.2576 & -1.6113 & -0.1393 & 0.0851 & -0.6906 & -1.8368 & -1.3670 & 0.8378 & -0.4019 & 0.1360 & -0.0141 & 0.4092 & 1.7831 & 0.3495
\end{pmatrix}$$

■

EXAMPLE 4.12. An example of a Wavelet Matrix in which we show the self-similarities of Ψ for a Daubechies wavelet with 10 vanishing moments and $\nu = 4$

$$\left(\begin{array}{cccccccccccccccc}
1 & -0.4001 & -0.0756 & 1.8211 & 0.9386 & 1.1124 & 0.5991 & -1.0115 & -0.1350 & 0.0767 & -0.2089 & -1.2148 & -2.4446 & 1.2069 & -0.2802 & 0.1635 \\
1 & -0.9083 & 0.1950 & 1.7262 & 0.1707 & -0.5580 & -1.6766 & -0.2701 & -0.4169 & 0.1504 & 0.0078 & 0.5352 & 2.1352 & 1.0327 & -1.4666 & 0.8586 \\
1 & -1.2639 & 0.0370 & 0.9790 & -0.3994 & -0.6421 & -1.6542 & 1.0571 & 0.1635 & -0.1350 & 0.0767 & -0.2089 & -1.2148 & -2.4446 & 1.2069 & -0.2802 \\
1 & -1.4070 & -0.3470 & -0.1322 & -0.8619 & 0.7298 & 0.6705 & 1.7957 & 0.8586 & -0.4169 & 0.1504 & 0.0078 & 0.5352 & 2.1352 & 1.0327 & -1.4666 \\
1 & -1.3412 & -0.5898 & -1.1557 & -1.0115 & 0.9386 & 1.1124 & 0.5991 & -0.2802 & 0.1635 & -0.1350 & 0.0767 & -0.2089 & -1.2148 & -2.4446 & 1.2069 \\
1 & -1.0828 & -0.3183 & -1.6029 & -0.2701 & 0.1707 & -0.5580 & -1.6766 & -1.4666 & 0.8586 & -0.4169 & 0.1504 & 0.0078 & 0.5352 & 2.1352 & 1.0327 \\
1 & -0.6660 & 0.3857 & -1.4017 & 1.0571 & -0.3994 & -0.6421 & -1.6542 & 1.2069 & -0.2802 & 0.1635 & -0.1350 & 0.0767 & -0.2089 & -1.2148 & -2.4446 \\
1 & -0.1519 & 1.2345 & -0.7553 & 1.7957 & -0.8619 & 0.7298 & 0.6705 & 1.0327 & -1.4666 & 0.8586 & -0.4169 & 0.1504 & 0.0078 & 0.5352 & 2.1352 \\
1 & 0.4001 & 1.8211 & -0.0756 & 0.5991 & -1.0115 & 0.9386 & 1.1124 & -2.4446 & 1.2069 & -0.2802 & 0.1635 & -0.1350 & 0.0767 & -0.2089 & -1.2148 \\
1 & 0.9083 & 1.7262 & 0.1950 & -1.6766 & -0.2701 & 0.1707 & -0.5580 & 2.1352 & 1.0327 & -1.4666 & 0.8586 & -0.4169 & 0.1504 & 0.0078 & 0.5352 \\
1 & 1.2639 & 0.9790 & 0.0370 & -1.6542 & 1.0571 & -0.3994 & -0.6421 & -1.2148 & -2.4446 & 1.2069 & -0.2802 & 0.1635 & -0.1350 & 0.0767 & -0.2089 \\
1 & 1.4070 & -0.1322 & -0.3470 & 0.6705 & 1.7957 & -0.8619 & 0.7298 & 0.5352 & 2.1352 & 1.0327 & -1.4666 & 0.8586 & -0.4169 & 0.1504 & 0.0078 \\
1 & 1.3412 & -1.1557 & -0.5898 & 1.1124 & 0.5991 & -1.0115 & 0.9386 & -0.2089 & -1.2148 & -2.4446 & 1.2069 & -0.2802 & 0.1635 & -0.1350 & 0.0767 \\
1 & 1.0828 & -1.6029 & -0.3183 & -0.5580 & -1.6766 & -0.2701 & 0.1707 & 0.0078 & 0.5352 & 2.1352 & 1.0327 & -1.4666 & 0.8586 & -0.4169 & 0.1504 \\
1 & 0.6660 & -1.4017 & 0.3857 & -0.6421 & -1.6542 & 1.0571 & -0.3994 & 0.0767 & -0.2089 & -1.2148 & -2.4446 & 1.2069 & -0.2802 & 0.1635 & -0.1350 \\
1 & 0.1519 & -0.7553 & 1.2345 & 0.7298 & 0.6705 & 1.7957 & -0.8619 & 0.1504 & 0.0078 & 0.5352 & 2.1352 & 1.0327 & -1.4666 & 0.8586 & -0.4169
\end{array}\right)$$

■

Moreover, apart from the self-similarities explained above, it is easy to see that the j blocks of the matrix are sparse for all j such that $2^j > 2p-1$. This is due to the fact that by Remark 3.18 the support of $\psi^{\text{PER}}_{-j,n}$ is of length $\min\left(1, \frac{2p-1}{2^j}\right)$. Hence if one is to evaluate $\psi^{\text{PER}}_{-j,n}$ on dyadic points in $[0,1)$, there is going to be an increasing (as j increases) number of zeroes on each column.

Having surveyed some of the most important properties of the wavelet matrices, we are not ready to begin the discussion on solving Equation (4.13) in Algorithm 4.8

4.2. Solving Equation (4.13)

This section is devoted to the particulars of Step 4 of Algorithm 4.8, that is to solving the following equation:

$$\left(\Psi_R - \mathit{\Delta}^{(\mathsf{n})}_{\sigma,\varepsilon}\Psi\right)\mathrm{X}_{\mathsf{n}} = \wp_{\mathsf{n}} - \Psi_R.$$

Recall that Ψ and Ψ_R are the Wavelet Matrix and the Rotated Wavelet Matrix respectively (see Definition 4.2), and that

$$\left[\wp_{\mathsf{n}}\right]_i = F_{\sigma,\varepsilon}\left(\theta_i, [\Psi\mathbf{D}_{\mathsf{n}}]_i\right), \quad \text{and,}$$

$$\left[\mathit{\Delta}^{(\mathsf{n})}_{\sigma,\varepsilon}\right]_{i,\ell} = \begin{cases} \mathrm{D}_x F_{\sigma,\varepsilon}\left(\theta_i, [\Psi\mathbf{D}_{\mathsf{n}}]_i\right) & \text{if } i=\ell \\ 0 & \text{otherwise.}\end{cases}$$

This system poses some challenges that need to be addressed carefully. First of all, the dimension of the system is 2^ν, which grows very fast as we increase ν. Being our computational limit at $\nu = 30$ (see Chapter 5), we would like to solve the system for at least $\nu = 25$. Hence, the dimension $N = 33,554,432$. Moreover, as we have hinted the matrices will be stored taking advantage of their self-similarities and localized sparsity. Therefore, we have matrices that are very optimized for storage (which in any case is our main limiter) but not so much for computations. In Annex A one can find the data types used to deal with the matrices, as well as the implementation of their basic operations. Thus, any hope of using decomposition methods should be thrown out the window. Luckily for us (and as it is standard) we can use iterative methods to solve these kinds of equations with very large dimensions [**HdlL06b, HdlL06a, HdlL07**]. In Subsection 4.2.2 we shall study the mathematics behind these methods and we will intoduce the one that we have used for computations, the Transpose-Free Quasi Residual Method (or TFQMR).

Moreover, we need to study carefully the systems in case some sort of preconditioning is required. In Subsection 4.2.1 we will study the spectrum of

$$\Psi_R - \mathit{\Delta}^{(\mathsf{n})}_{\sigma,\varepsilon}\Psi$$

for various instances of the systems we study, and show how some preconditioning strategies may help with the computations.

4.2.1. Preconditioning strategies.

We shall see what preconditioners may help to solve the system. To this end we shall study the spectrum of

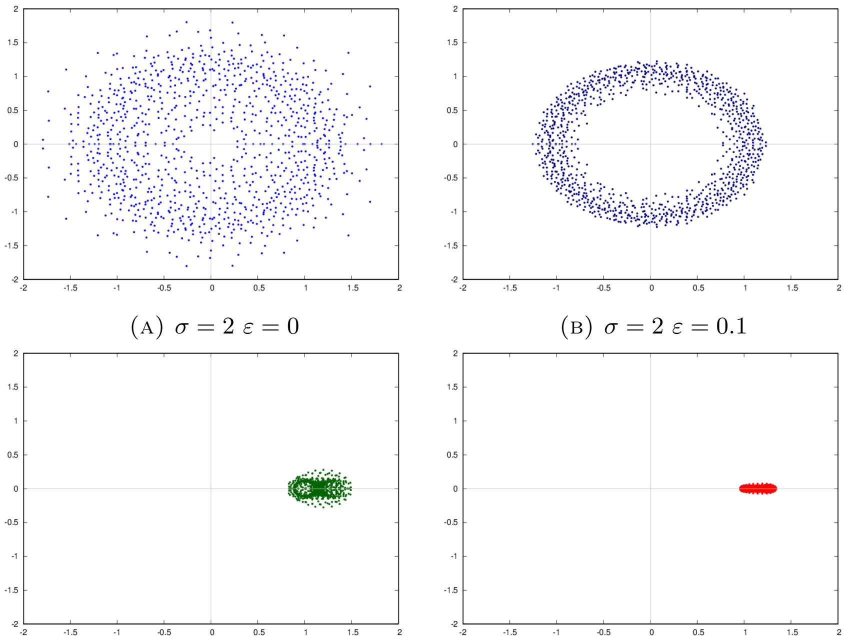
$$\Psi_R - \Delta_{\sigma,\varepsilon}\Psi \tag{4.17}$$

close to the solution to see how it may change with the preconditioners. In particular, we are going to use a left preconditioner, that is an invertible matrix P that allows us to re-write the system as

$$PAx = Pb, \tag{4.18}$$

instead of $Ax = b$. Note that the solutions of both systems are the same, since P is invertible. The goal of this matrix is usually to tame the condition number of the system. There are but few methods to find a condition matrix, all relegated to very particular cases. Many times, the ability to find a precondition matrix boils down to sheer luck.

4.2.1.1. *The Keller-GOPY system.* In Figure 4.1 (A) and (B) one we can see the spectrum of (4.17) for the Keller-GOPY model for a pinched and a non-pinched case for $\nu = 10$, that is dimension $N = 2^{10} = 1024$. Note that for the non-pinched case the eigenvalues are very close to the circle of radius one. However, for the pinched case, one can see that some eigenvalues may be very close to zero whilst some are clearly greater than one. However, we are very lucky that $\Psi_R^\intercal$ turns out to be a very good preconditioner. To see the effects of the preconditioner, we have computed the spectrum of (4.17) for the same two cases as in Figure 4.1 (A) and (B) but using $\Psi_R^\intercal$ as a preconditioner, that is the spectrum of $\Psi_R^\intercal\big(\Psi_R - \Delta_{\sigma,\varepsilon}\Psi\big)$. In Figure 4.1 (A) and (B) one can see the result. Note that now the eigenvalues are centred around 1 instead of being centred around 0, all of them to the right of $\Re(\lambda) = 0.8$. This ensures a much nicer condition number of the matrix, making its solving a much easier task.



(A) $\sigma = 2$ $\varepsilon = 0$

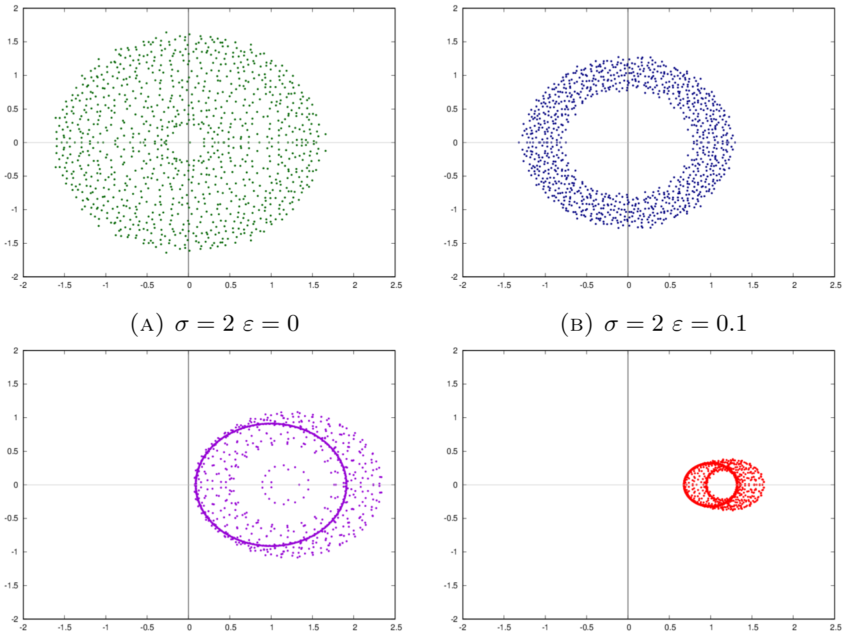
(B) $\sigma = 2$ $\varepsilon = 0.1$

(C) Preconditioned, $\sigma = 2$ $\varepsilon = 0$

(D) Preconditioned, $\sigma = 2$ $\varepsilon = 0.1$

Figure 4.1. **Keller-GOPY Attractor**:(A) and (B) correspond to the spectrum of the matrix $\Psi_R - \Delta_{\sigma,\varepsilon}\Psi$ for $\sigma = 2$. (A) corresponds to the pinched case while (B) corresponds to the non-pinched case. (C) and (D) correspond to the same situation for $\Psi_R^\intercal(\Psi_R - \Delta_{\sigma,\varepsilon}\Psi)$, i.e. using $\Psi_R^\intercal$ as a preconditioner.

4.2.1.2. *The Alsedà-Misiurewicz system.* This case is very similar to the one above, however the matrices do not behave as nicely. In Figure 4.2 one can see that the spectrum of the matrix for the pinched case is more scattered than in the Keller-GOPY model. Moreover, even though the preconditioning by $\Psi_R^\intercal$ still shows good results, notice that for the pinched case the eigenvalues get dangerously close to 0. In particular, we have definitely noticed a decrease in the speed of convergence when solving this model in comparison to the Keller-GOPY system.



(A) $\sigma = 2$ $\varepsilon = 0$

(B) $\sigma = 2$ $\varepsilon = 0.1$

(C) Preconditioned, $\sigma = 2$ $\varepsilon = 0$

(D) Preconditioned, $\sigma = 2$ $\varepsilon = 0.1$

FIGURE 4.2. **Alsedà-Misiurewicz Attractor**: (A) and (B) correspond to the spectrum of the matrix $\Psi_R - \Delta_{\sigma,\varepsilon}\Psi$ for $\sigma = 2$. (A) corresponds to the pinched case while (B) corresponds to the non-pinched case. (C) and (D) correspond to the same situation for $\Psi_R^{\intercal}(\Psi_R - \Delta_{\sigma,\varepsilon}\Psi)$, i.e. using $\Psi_R^{\intercal}$ as a preconditioner.

4.2.1.3. *The Nishikawa Kaneko system.* As it has been usual in the rest of this thesis, this case is essentially different from the previous two cases. For the Nishikawa-Kaneko model Matrix (4.17) has a very scattered spectrum and so far we have been unable to find good pre-conditioners for it. In Figure 4.3 one can see the spectrum of Matrix (4.17) for $\varepsilon = 0.15$.

4.2.2. Krylov methods. Krylov methods are a sub-class of iterative methods which include some of the most well known ones, such as GMRES or BiCGSTAB. Given a linear equation

$$Ax = b,$$

where A is an $N \times N$ non-singular matrix, $b \in \mathbb{R}^N$ and x is our unknown. Instead of trying to deal with the matrix itself and try to find some kind of decomposition, iterative methods try to find increasingly better approximations of the actual solution. All the cases we are going to study are particular instances of projection methods. That is, given x_0 an initial guess, we compute the initial residual $r_0 := b - Ax_0$ (that is, the difference between our guess and the actual solution), and two m dimensional subspaces $\mathcal{K}$ and $\mathcal{L}$ such that we can find a (hopefully better) approximation of the solution $\tilde{x}$ satisfying

$$\tilde{x} \in x_0 + \mathcal{K}, \text{ and } b - A\tilde{x} \perp \mathcal{L}.$$

It follows that one should choose the spaces $\mathcal{K}$ and $\mathcal{L}$ in a way that makes this approximation as efficient as possible. When $\mathcal{K} = \mathcal{L}$ we say that the method is an *orthogonal projection.* Whenever they are different, the projection is called *oblique.*

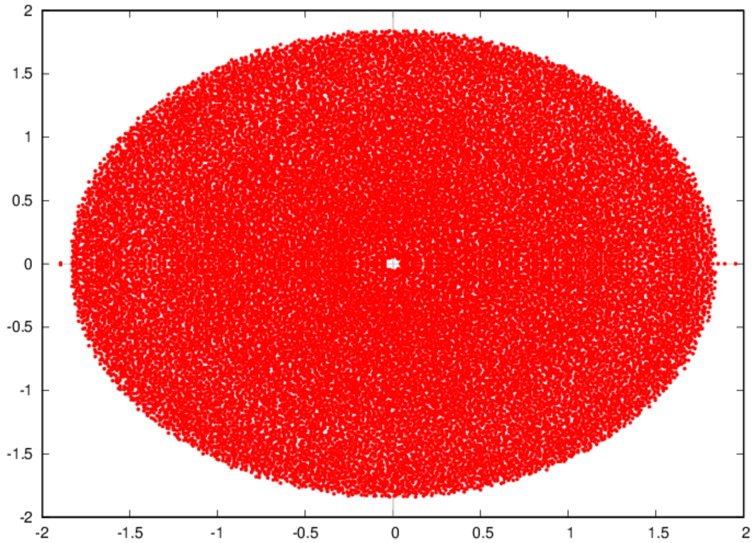


FIGURE 4.3. **Nishikawa-Kaneko Attractor**: eigenvalues corresponding to matrix $\Psi_R - \Delta_{\sigma,\varepsilon}\Psi$ with $\nu = 15$ (that is Dimension 2^{15}) and $\varepsilon = 0.15$. Note the incredible scattering of eigenvalues. We were unable to find any pre-conditioner that made things easier.

The main defining characteristic of Krylov methods is that the space $\mathcal{K}$ is the so-called *Krylov subspace*

$$\mathcal{K}_k(A, r_0) = \langle\!\langle r_0, Ar_0, A^2 r_0, \dots, A^k r_0 \rangle\!\rangle. \tag{4.19}$$

Where as before $r_0 = b - Ax_0$ is the initial residual of the system. The choice of $\mathcal{L}$ and how one deals with the bases of $\mathcal{K}_k(A, r_0)$ are what will differentiate one family of methods from another.

Note that all the $A^k r_0$ are linearly independent for $k = 0, 1, \dots, k_0$, where k_0 is smaller than the degree of the minimal polynomial. For example, if r_0 is an eigenvector then the dimension of the Krylov subspace is 1.

Since we want $\tilde{x} \in \mathcal{K}_k(A, r_0)$, it follows that our approximations will be of the form

$$A^{-1}b \approx \tilde{x} = x_0 + p_{m-1}(A) r_0,$$

where p_{m-1} is a polynomial of degree $m-1$. However, the Krylov basis is far from ideal. Mostly because as k increases we might have that $A^k b \approx A^{k+1} b$ due to the dominance of the greatest eigenvector. Hence, it is desirable to rewrite the basis $r_0, Ar_0, A^2 r_0, \dots, A^k r_0$ in a way that allows us to do easier computations. This is mainly done through the Arnoldi process, which consists of a simple modification of the Gram-Schmid procedure.

ALGORITHM 4.13 (Arnoldi's Method, Algorithm 6.1 in [**Saa03**]).

```
procedure ARNOLDI
    Choose $v_1$ such that $\|v_1\| = 1$
    for $j \leftarrow 1, \dots, m$ do
        for $i \leftarrow 1, \dots, j$ do
            $h_{i,j} = \langle Av_j, v_i \rangle$
        end for
        $w_j \leftarrow Av_j - \sum_{i=1}^{j} h_{i,j} v_i$
        $h_{j+1,j} \leftarrow \|w_j\|$
        **if** $h_{j+1,j} = 0$ **then**
            STOP
        **end if**
        $v_{j+1} \leftarrow w_j / h_{j+1,j}$
    **end for**
**end procedure**
```

■

The Arnoldi process lies on the centre of some of the most well known iterative methods, such as General Minimal Residual Method (or GMRES). However, the general implementations demand the storage of large amounts of memory. For example in GMRES, to guarantee convergence, one needs to store most $h_{i,j}$ (using methods such as the Givens rotation some of them may be discarded), which requires $\frac{(N+1)N}{2}$ items, hence requiring the storage of $\mathcal{O}(N^2)$ elements. We have been concerned with minimizing our memory usage of floating point numbers thus such requirements negates all the efforts. If one were to use restarted versions of the algorithms, they are specially suited for very particular cases, such as when the eigenvalues are clustered, which, in general, is not the case for us. Hence, we have discarded the Arnoldi route in favour of the methods that use *Lanczos biorthogonalization* techniques. In this circumstances the space $\mathcal{L}_k$ is defined to be

$$\mathcal{L}_k = \mathcal{K}_k(A^\intercal, r_0) = \langle\!\langle r_0, A^\intercal b, (A^\intercal)^2 r_0, \ldots, (A^\intercal)^k r_0 \rangle\!\rangle.$$

The Lanczos biorthogonalization allows us then to obtain bases of $\mathcal{K}_k$ and $\mathcal{L}_k$ that are biorthogonal, that is

$$\langle v_i, w_j \rangle = \delta_{i,j}, \quad 1 \le i, j \le m.$$

Algorithm 4.14 (Lanczos Biorthogonalization, Algorithm 7.1 in [**Saa03**]).

```
**procedure** Lanczos
    Choose $v_1$ and $w_1$ such that $\langle v_1, w_1 \ \rangle = 1$
    $\beta_1 \leftarrow \delta_1 \leftarrow 0$
    $w_0 \leftarrow v_0 \leftarrow 0$
    **for** $j \leftarrow 1, \ldots, m$ **do**
        $\alpha_j \leftarrow \langle Av_j, w_j \rangle$
        $v_{j+1} \leftarrow Av_j - \alpha_j v_j - \beta_j v_{j-1}$
        $w_{j+1} \leftarrow A^\intercal w_j - \alpha_j w_j - \delta_j w_{j-1}$
        $\delta_{j+1} \leftarrow \sqrt{|\langle v_{j+1}, w_{j+1} \rangle|}$
        **if** $\delta_{j+1} = 0$ **then**
            STOP
        **end if**
        $\beta_{j+1} \leftarrow \langle v_{j+1}, w_{j+1} \rangle / \delta_{j+1}$
        $v_{j+1} \leftarrow v_{j+1} / \delta_{j+1}$
        $w_{j+1} \leftarrow w_{j+1} / \beta_{j+1}$
    **end for**
**end procedure**
```

■

Remark 4.15. Note that for computing w_{j+1} and v_{j+1} we only require α_j, β_j, $\beta_j + 1$, δ_j and $\delta_j + 1$. Hence, contrary to what happened for the Arnoldi Procedure, not all the coefficients α, β and δ need to be stored for the whole algorithm. ■

The following proposition shows us that the vectors obtained are indeed bases for the $\mathcal{K}_k$ and $\mathcal{L}_k$ spaces and that these bases are biorthogonal.

PROPOSITION 4.16 (Proposition 7.1 in [**Saa03**]). *If Algorithm 4.14 does not break down before step m, then the vectors $\{v_i\}_{i=1}^m$ and $\{w_j\}_{j=1}^m$ form a biorthogonal system. Moreover, $\{v_i\}_i$ is a basis of $\mathcal{K}_k(A, v_1)$, and $\{v_i\}_i$ is a basis of $\mathcal{K}_k(A^\intercal, w_1)$.*

There might be two reasons why $\delta_{j+1} = 0$. Either one of the vectors v_{j+1} or w_{j+1} is zero or they are orthogonal. The former case is not very worrying, since it indicates convergence of the method and the finding of an exact solution. The latter is quite inconvenient. However, some techniques exist to try to avoid this scenario. Depending on the matrix (for example symmetric matrices) different methods may be applied, giving rise to different systems. Many iterative methods that use Lanczos biorthogonalization techniques actually solve $Ax = b$ and $A^\intercal x = b$, while requiring to multiply by both A and $A^\intercal$ at each iteration. Since we are only focused in solving $Ax = b$, we have decided to use a method called Transpose-Free Quasi-Residual Method, or TFQMR. It is quite robust and only requires one matrix multiplication per iteration, hence since one of our bottlenecks is the product of a matrix by a vector it is very convenient.

ALGORITHM 4.17 (TFQMR, an adaptation of Algorithm 7.8 in [**Saa03**]). This algorithm is an adaptation suited for our computations.

```
procedure TFQMR($A$, $b$, $x_0$, TOL, $M$)        ▷ $M$ is the max number of iterates
    $w_0 \leftarrow u_0 \leftarrow r_0 \leftarrow b - Ax_0$
    $\tau_0 \leftarrow \|r_0\|$
    $\theta_0 \leftarrow \eta_0 \leftarrow 0$
    $\rho_0 \leftarrow \langle r_0, r_0 \rangle$                    ▷ Technically we can use another $\tilde{r}_0$
    for $j \leftarrow 1, \ldots, M$ do
        if $j \mod 2 = 0$ then
            $\alpha_{j+1} \leftarrow \alpha_j \leftarrow \frac{\rho_j}{\langle v_j, r_0 \rangle}$
            $u_{j+1} \leftarrow u_j - \alpha_j v_j$
        end if
        $w_{j+1} \leftarrow w_j - \alpha_j A u_j$
        $d_{j+1} \leftarrow u_j + (\theta_j^2 / \alpha_j) \eta_j$
        $\theta_{j+1} \leftarrow \|w_{j+1}\| / \tau_j$
        $c_{j+1} \leftarrow (1 + \theta_{j+1}^2)^{-1/2}$
        $\tau_{j+1} \leftarrow \tau_j \theta_{j+1} x_{j+1}$
        $\eta_{j+1} \leftarrow c_{j+1}^2 \alpha_j$
        $x_{j+1} \leftarrow x_j + \eta_{j+1} d_{m+1}$
        if $\tau_{j+1} \cdot \sqrt{j+1}$ <TOL then          ▷ By Equation 7.83 in [Saa03]
            return x                                   ▷ $\|b - Ax_j\| \leq \tau_{j+1}\sqrt{j+1}$
        end if
        if $j \mod 2 = 1$ then
            $\rho_{j+1} \leftarrow \langle w_{j+1}, r_0 \rangle$
            $\beta_{j-1} \leftarrow \rho_{j+1} / \rho_{j-1}$
            $u_{j+1} \leftarrow w_{j+1} + \beta_{j-1} u_m$
            $v_{j+1} \leftarrow A u_{j+1} + \beta_{j-1}(A u_j + \beta_{j-1} v_{j-1})$    ▷ $Au_j$ is already computed
        end if
    end for
end procedure
```

■

REMARK 4.18. We have also tried using other methods like restarted GMRES or BiCGSTAB, but the results have been quite lacklustre both in terms of speed and in terms of convergence. ■

4.3. Testing the Algorithm I: The Keller-GOPY and Alsedà-Misiurewicz models

In this section we will compute the wavelet expansion and regularities of the attractors given by the graph of $\varphi : \mathbb{S}^1 \to \mathbb{R}$ for the systems we call the Keller-GOPY and the Alsedà-Misiurewicz models (see Subsection 2.1.3). Recall that these systems correspond to the systems for which we have existing analytical results regarding the strangeness. For both systems we will compute the truncated wavelet expansion (or the vector $\mathbf{D}$ as in Equation (4.5)) for various values of the parameters σ and ε using Algorithm 4.8. To test the fitness of the expansion, we have recovered the graph of the attractor using Equation 4.10 and comparing it to the values obtained from Algorithm 4.5.

In all cases, we have used Daubechies wavelets with $p = 10$ vanishing moments. This is done as a compromise between ease of computation (see Chapter 5) and the ability to compute regularities as in Theorem 3.35. Regarding the regularity, results for an interesting cohort of examples have been obtained by defining a dependence $\varepsilon(\sigma)$ so that whilst sweeping the parameter σ we can obtain results for different values of ε. It should be remembered that for this systems what determines the strangeness is the value $\varepsilon = 0$.

With all the obtained results we will proceed to compute the regularities of φ for each system using Algorithm 3.38, that is by computing

$$s_{-j} := \log_2 \left(\sup_{0 \le n \le 2^j - 1} \left| \left\langle f, \psi^{\text{PER}}_{-j,n} \right\rangle \right| \right),$$

doing a linear fitting on the pairs $(-j, s_{-j})$, and applying Theorem 3.35.

4.3.1. The Keller-GOPY model.

In Subsection 2.1.2 we defined the Keller-GOPY model as

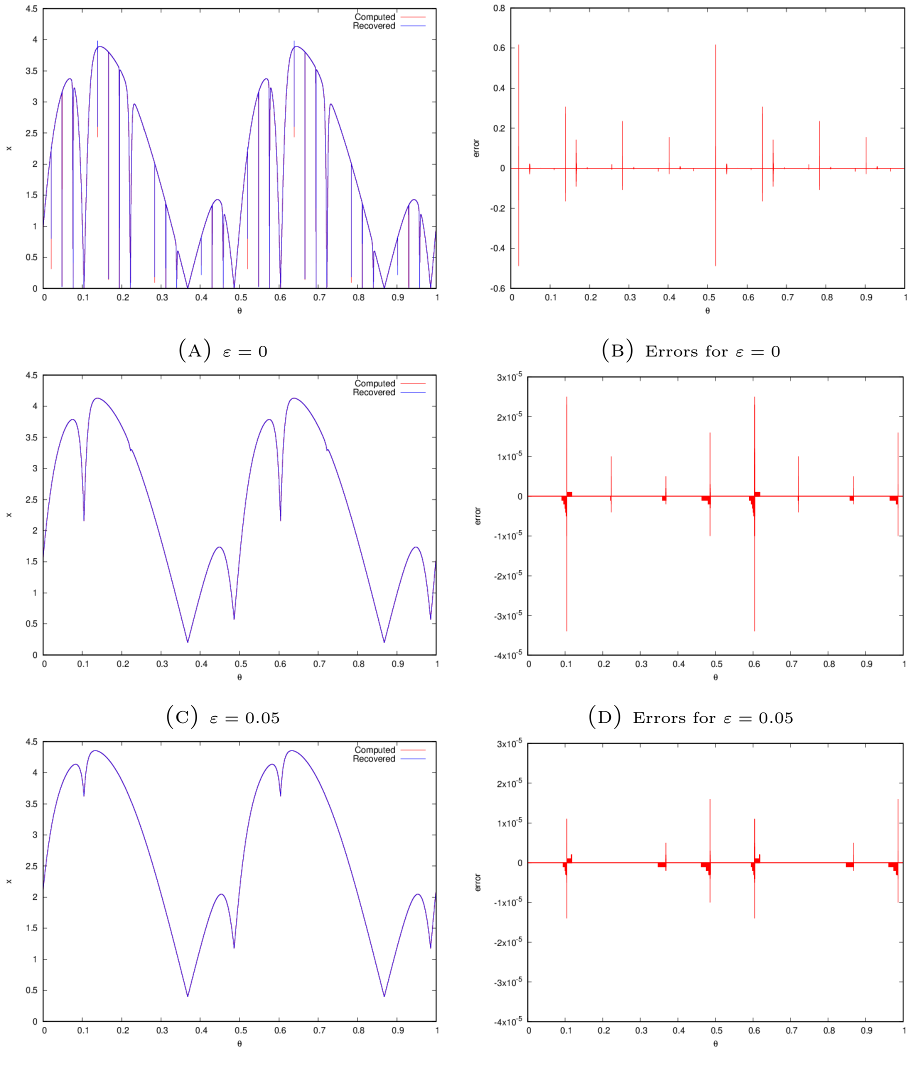
$$\begin{pmatrix} \theta_{n+1} \\ x_{n+1} \end{pmatrix} = \mathfrak{F}_{\sigma,\varepsilon}(\theta_n, x_n) = \begin{pmatrix} R_\omega(\theta_n) \\ 2\sigma \tanh(x_n) \cdot (\varepsilon + |\cos(2\pi\theta_n)|) \end{pmatrix}, \tag{4.20}$$

where $\omega = \frac{\sqrt{5}-1}{2}$. Recall that the name indicates that it is a particular instance of the family of systems in which we can apply the results by Keller [**Kel96**], and that the system chosen is entirely based on the GOPY model [**GOPY84**]. From [**Kel96**] if $\varepsilon = 0$ we have that the set

$$\left\{ \left(\tfrac{i}{4} + n\omega \pmod 1, 0\right) : \; n \in \mathbb{N}, \; i \in \{1, 3\} \right\}$$

is both a subset of the attractor and is dense (and invariant) in $x \equiv 0$, which for $\sigma > 1$ is a repellor.

On the other hand, if $\varepsilon > 0$ the system is not pinched and the attractor is regular. Hence, the regularity is zero if $\varepsilon = 0$ and is positive if $\varepsilon > 0$ [**AMR16**]. Now, we would like to study how good are the approximations we have done. To this end we have obtained an approximation of the attractor on dyadic points using Algorithm 4.5 fixing the tolerance to be at least 10^{-32}. Then, we have compared this approximation with the one obtained from Equation (4.10). The results can be found in Figure 4.4. One can see that the differences between the computed attractor and the recovered from the wavelet expansion become much bigger for $\varepsilon = 0$, i.e the system is pinched, hence the regularity is zero. For these cases, however, when we zoom in the points where the errors are the greatest, we see that those spots correspond to the pinched points (see Figure 4.10). Moreover, the shape of the reconstructed attractor is still very similar to the one of the computed attractor, but since the jump is so sudden, small differences lead to greater errors. This is clearly visible in Figure 4.5.



(A) $\varepsilon = 0$

(B) Errors for $\varepsilon = 0$

(C) $\varepsilon = 0.05$

(D) Errors for $\varepsilon = 0.05$

(E) $\varepsilon = 0.1$

(F) Errors for $\varepsilon = 0.1$

FIGURE 4.4. Comparisons for the Keller-GOPY system between the computed attractor using Algorithm 4.5 and the one obtained from the reconstruction in Equation (4.10). All plots correspond to instances of System (2.5) with $\sigma = 2$. Figures (A), (C) and (E) contain two plots. In red the one corresponding to the direct approximation of the attractor using Algorithm 4.5. In blue the one obtained by the reconstruction using Equation 4.10. For both cases we have fixed $\nu = 20$, that is 2^{20} points and wavelet coefficients. Figures (B), (D) and (F) correspond to the difference *Computed-Recovered*. Note the change of scale between Figure (B) and Figures (D) and (F). A closer look to this phenomenon can be found in Figure 4.5.

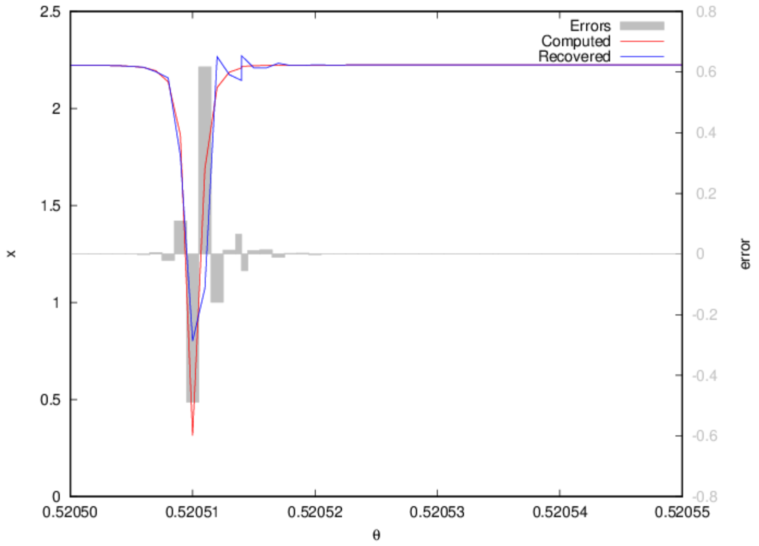
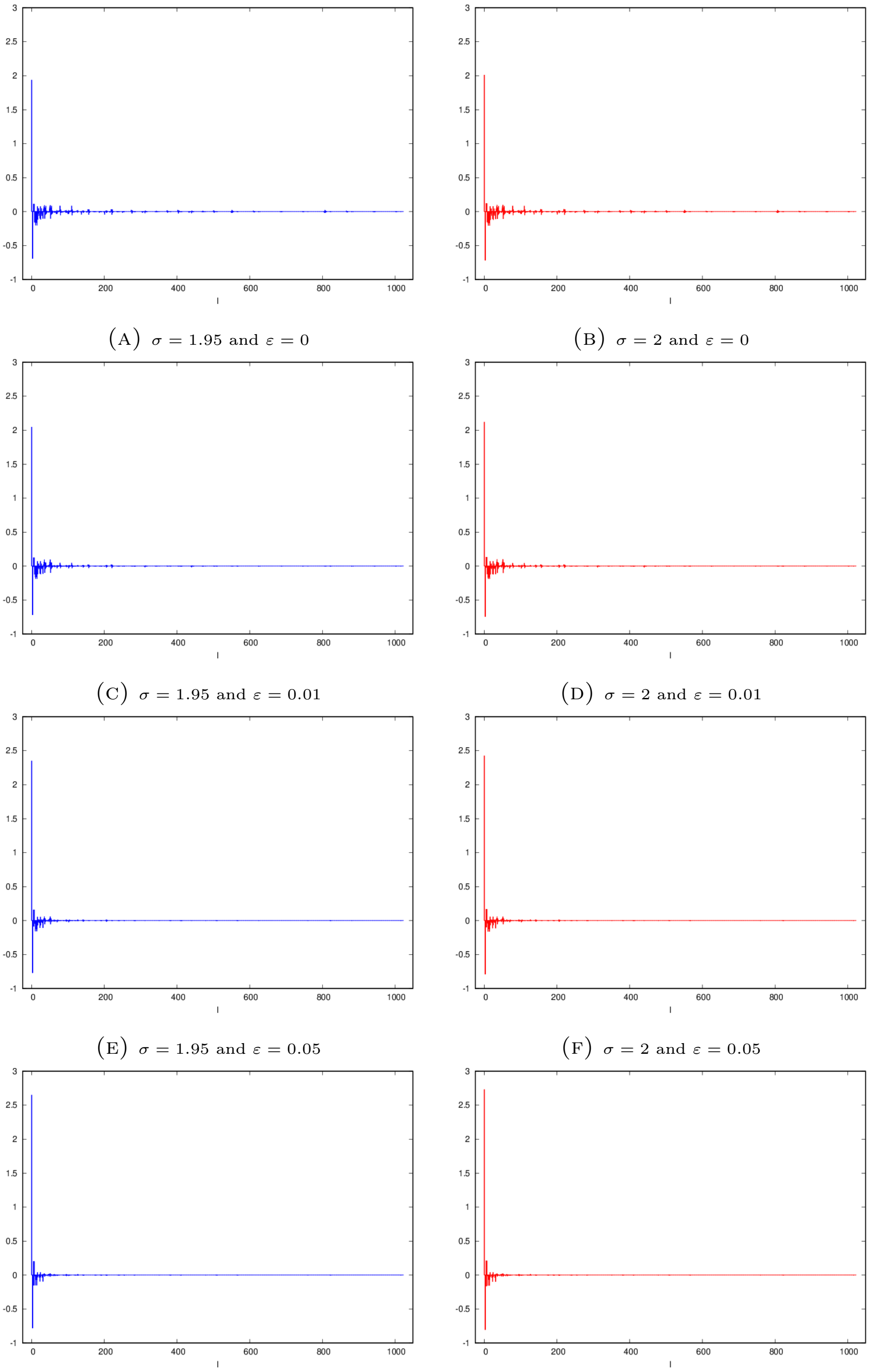


FIGURE 4.5. For the Keller-GOPY system for $\nu = 20$, $\sigma = 2$ and $\varepsilon = 0$, zoom into the area of greatest difference between the computed attractor and the recovered attractor in Figure 4.4 A and D.

Furthermore, we want to study in a more or less qualitative manner how slight changes in the values of σ and ε modify the wavelet expansion of the object. This is crucial when it comes to considering using continuation techniques in the Newton steps. In Figure 4.6 one can see that there is a big change between systems between the case $\varepsilon = 0$ and $\varepsilon > 0$. On the other hand, if we fix ε a change on σ produces mellower changes in the expansion. Hence, it seems that continuation should be applied as long as one takes into account that the approximation to $\varepsilon = 0$ should be done carefully. In Table 4.1 one can see a more quantitative presentation. Note that as the system is less regular, the maximum value of the coefficients for the larger computed values of j becomes bigger, while the values for smaller j increase with ε. This is completely expected in the light of Theorem 3.35 and Remark 3.36. Finally, we would like to compute the regularities of the attractors using Algorithm 3.38. We have studied the regularities for the Keller-GOPY model using a parametric sweep for $\sigma \in [1.9, 2]$ and using the following parametrization for ε:

$$\varepsilon(\sigma) = \begin{cases} 0 & \text{if } \sigma \leq 2, \\ (\sigma - 2)^2 & \text{if } \sigma > 2. \end{cases} \tag{4.21}$$

Note that $\varepsilon(\sigma) = 0$ if $\sigma \leq 2$, hence by Theorem 2.6 and Proposition 3.34 we would expect the regularity to be zero for $\sigma \in [1.9, 2]$, whilst it should be greater than 0 whenever $\sigma > 2$. In Figure 4.7 one can see the computed regularity using Algorithm 3.38 for various values of ν. Note that as the value of ν increases the more pronounced the transition from non-zero regularity to zero regularity is, that is, the behaviour is closer to what one would expect at the light of Example 3.33, since Theorem 2.6 tells us that the attractor is the graph of a continuous function if $\varepsilon > 0$ whilst it is upper semi-continuous for $\varepsilon = 0$. Finally, in Figure 4.8 one can see the intermediate steps of Algorithm 3.38. For this system, the linear correlation is very clear, as it can be seen by the high values of the Pearson correlation coefficient r^2.



(A) $\sigma = 1.95$ and $\varepsilon = 0$

(B) $\sigma = 2$ and $\varepsilon = 0$

(C) $\sigma = 1.95$ and $\varepsilon = 0.01$

(D) $\sigma = 2$ and $\varepsilon = 0.01$

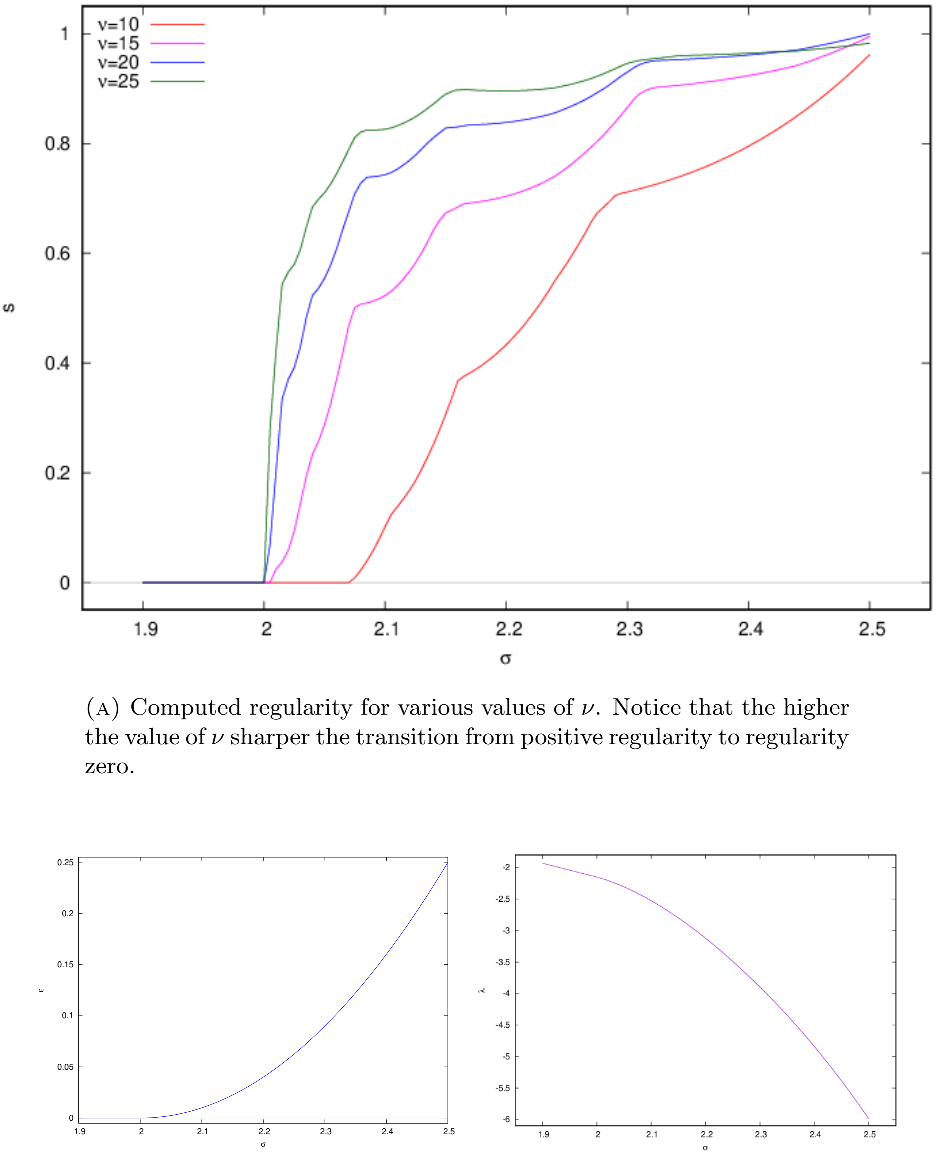
(E) $\sigma = 1.95$ and $\varepsilon = 0.05$

(F) $\sigma = 2$ and $\varepsilon = 0.05$

(G) $\sigma = 1.95$ and $\varepsilon = 0.1$

(H) $\sigma = 2$ and $\varepsilon = 0.1$

FIGURE 4.6. Representation of the computed value of $\mathbf{D}$ for $\nu = 10$ and various values of σ and ε. Note the flattering of $[\mathbf{D}]_\ell$ for larger values of ℓ as ε increases.

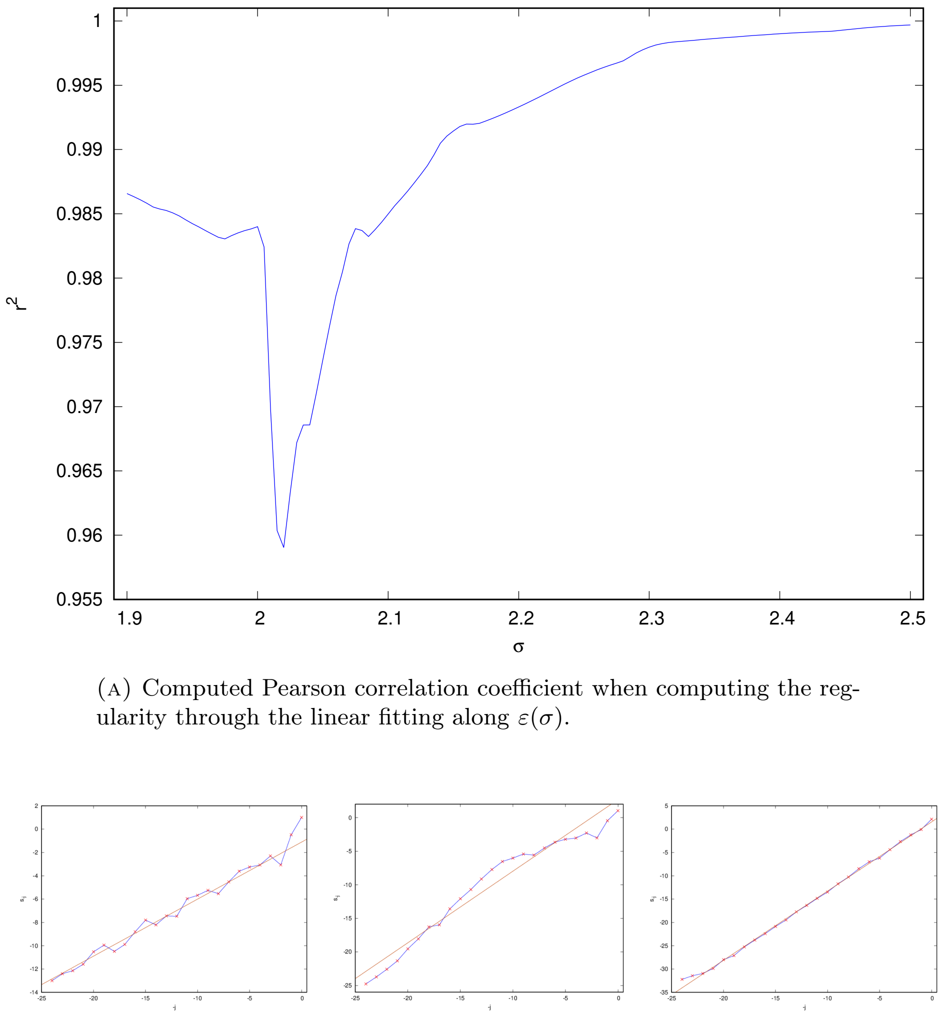


(A) Computed regularity for various values of ν. Notice that the higher the value of ν sharper the transition from positive regularity to regularity zero.

(B) Plot of the parametrization given by Equation (4.21)

(C) Computed Lyapunov exponent on the attractor using Algorithm 2.8 for the parametrization given by Equation (4.21)

Figure 4.7. Regularity computed from the wavelet coefficients for different values of ν. Moreover, i n Figure (B) we have plotted the parametrization (4.21) for the sake of clarity. In Figure (C) one can find the Lyapunov exponent on the attractor along this parametrization.



(A) Computed Pearson correlation coefficient when computing the regularity through the linear fitting along $\varepsilon(\sigma)$.

(B) Linear plot for $\sigma = 1.9$, the equation of the regression line is $y = 0.489x - 1.110$.

(C) Linear plot for $\sigma = 2.02$, the equation of the regression line is $y = 1.066x + 2.682$. This is the plot with lowest correlation coefficient

(D) Linear plot for $\sigma = 2.5$, the equation of the regression line is $y = 1.483x + 1.552$.

FIGURE 4.8. For the Keller-GOPY system, Pearson correlation coefficient of the linear fit required in Algorithm 3.38 to compute the regularity corresponding to the parametrization $\varepsilon(\sigma)$ given by Equation 4.21. Moreover, three examples of the correlation corresponding to local extrema of r^2. In red the computed points, in orange the computed linear regression.

TABLE 4.1. Computed values of s_{-j} for $\nu = 15$, $\sigma = 2$, and various values of ε. It is interesting to focus on the difference between the values of s_{-j} between $\varepsilon = 0$ and $\varepsilon = 0.01$. While for $0 \leq j \leq 8$ the values are very similar, the situation changes for larger values of j. Note, for example, that for $j = 14$ the value of s_{-14} for $\varepsilon = 0$ is two orders of magnitude bigger than the one for $\varepsilon = 0.01$. Moreover, this disparity is exclusive to the transition from $\varepsilon = 0$ to $\varepsilon > 0$, since the difference between other values of ε are not as pronounced.

j	$\varepsilon = 0$	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.25$
0	1.863285	1.972168	2.270392	2.567642	3.282877
1	0.664647	0.691259	0.742475	0.759416	0.733901
2	0.103665	0.115256	0.149286	0.188689	0.281358
3	0.195352	0.180068	0.148248	0.142602	0.125911
4	0.112117	0.114462	0.101045	0.089480	0.050348
5	0.096182	0.093136	0.055595	0.022944	0.010252
6	0.082908	0.047025	0.018986	0.012364	0.007496
7	0.039285	0.025321	0.012168	0.005657	0.002634
8	0.021483	0.013034	0.002136	0.001218	0.000761
9	0.023645	0.003753	0.001533	0.000757	0.000274
10	0.019941	0.001475	0.000275	0.000137	0.000081
11	0.012349	0.000518	0.000215	0.000107	0.000032
12	0.007805	0.000203	0.000065	0.000032	0.000011
13	0.006357	0.000069	0.000025	0.000012	0.000005
14	0.002564	0.000023	0.000005	0.000002	0.000002

4.3.2. The Alsedà-Misiurewicz System. What we call the Alsedà-Misiurewicz [**AM08**] system is the following instance of System (2.1)

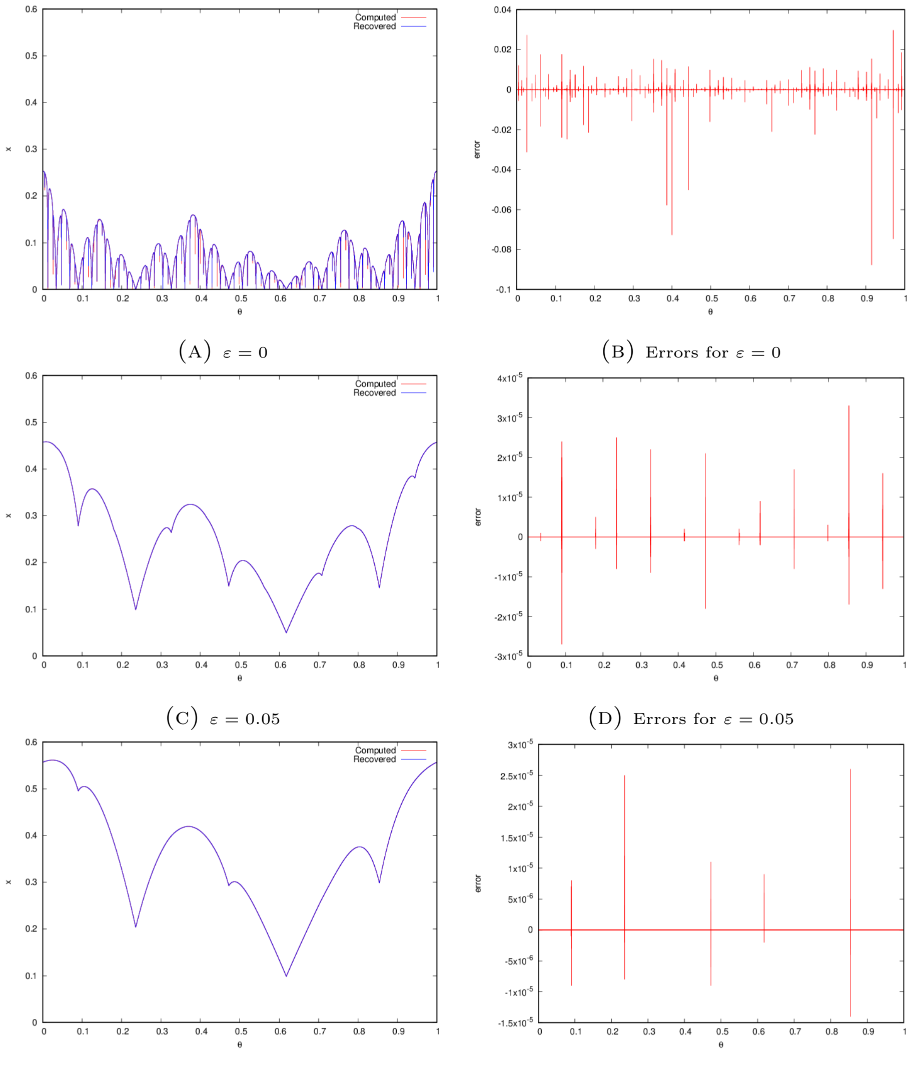
$$\begin{pmatrix} \theta_{n+1} \\ x_{n+1} \end{pmatrix} = \mathfrak{F}_{\sigma,\varepsilon}(\theta_n, x_n) = \begin{pmatrix} R_\omega(\theta_n) \\ 4x_n(1-x_n)(\varepsilon + \sigma_g \theta_n(1-\theta_n)) \end{pmatrix}. \tag{4.22}$$

In Chapter 2 we have already introduced this system and have shown that there exist analytical results that show the existence of an attractor that is strange provided $\varepsilon = 0$. As in the case of the Keller-GOPY model, we will start by comparing the reconstructed attractor form the computed wavelet expansion and the approximation of the attractor obtained with Algorithm 4.5. The results can be found in Figure 4.9. We find similar phenomena as in the Keller-GOPY system regarding the difference between the computed and reconstructed attractors when $\varepsilon = 0$. We have added a zoom around the biggest difference in Figure 4.10, which parallels Figure 4.5. The changes on the wavelet expansion corresponding to small changes on ε and σ completely mimic the Keller-GOPY case. Hence, for the sake of brevity we have not included their discussion here.

When it comes to computing regularities, we would like to validate the result that D. Romero obtained in [**RiS15**] using the Haar basis for the following parametrization of $\varepsilon(\sigma)$:

(4.23)

$$\varepsilon(\sigma) := \begin{cases} -\frac{3323299}{500000}\sigma^2 + \frac{60871083}{2500000}\sigma - \frac{34284691}{1562500} & \text{if } \sigma \in [1.6, 2.0), \\ (\sigma - 2.8)^8 + 0.0005 & \text{if } \sigma \in [2.0, 3.49), \\ \frac{366298374428641}{861000000000000}\sigma^2 - \frac{7013481614144179}{2152500000000000}\sigma + \frac{179131268100176321}{28700000000000000} & \text{if } \sigma \in [3.49, 3.9], \\ 0 & \text{otherwise.} \end{cases}$$



(A) $\varepsilon = 0$

(B) Errors for $\varepsilon = 0$

(C) $\varepsilon = 0.05$

(D) Errors for $\varepsilon = 0.05$

(E) $\varepsilon = 0.1$

(F) Errors for $\varepsilon = 0.1$

FIGURE 4.9. Comparisons for the Alsedà-Misiurewicz system between the computed attractor using Algorithm 4.5 and the one obtained from the reconstruction given by Equation (4.10). All plots correspond to instances of System (2.6) with $\sigma = 2$. Figures (A), (E) and (E) contain two graphs. In red the one corresponding to the direct approximation of the attractor using Algorithm 4.5. In blue the one obtained from the reconstruction given by (4.10). For both cases we have fixed $\nu = 15$, that is 2^{15} points and wavelet coefficients. Figures (B), (D) and (F) correspond to the difference *Computed-Recovered*. Note the change of scale between Figure (B) and Figures (D) and (F). A closer look around the points with greater error can be found in Figure 4.10.

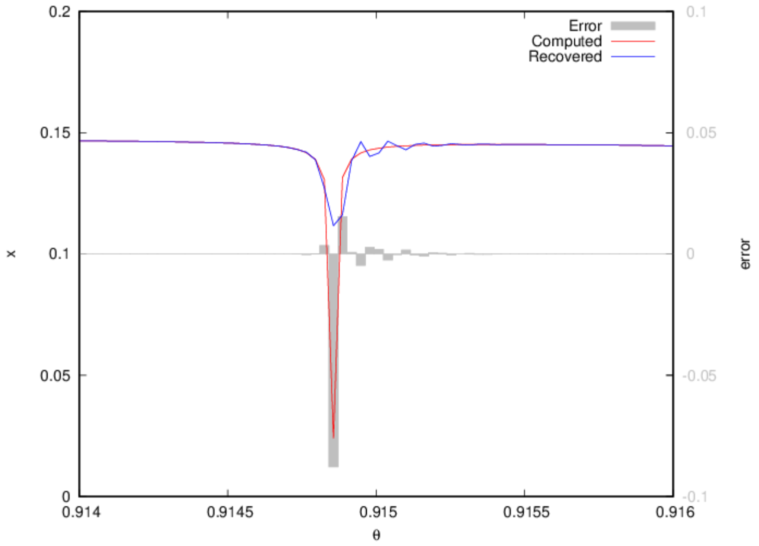
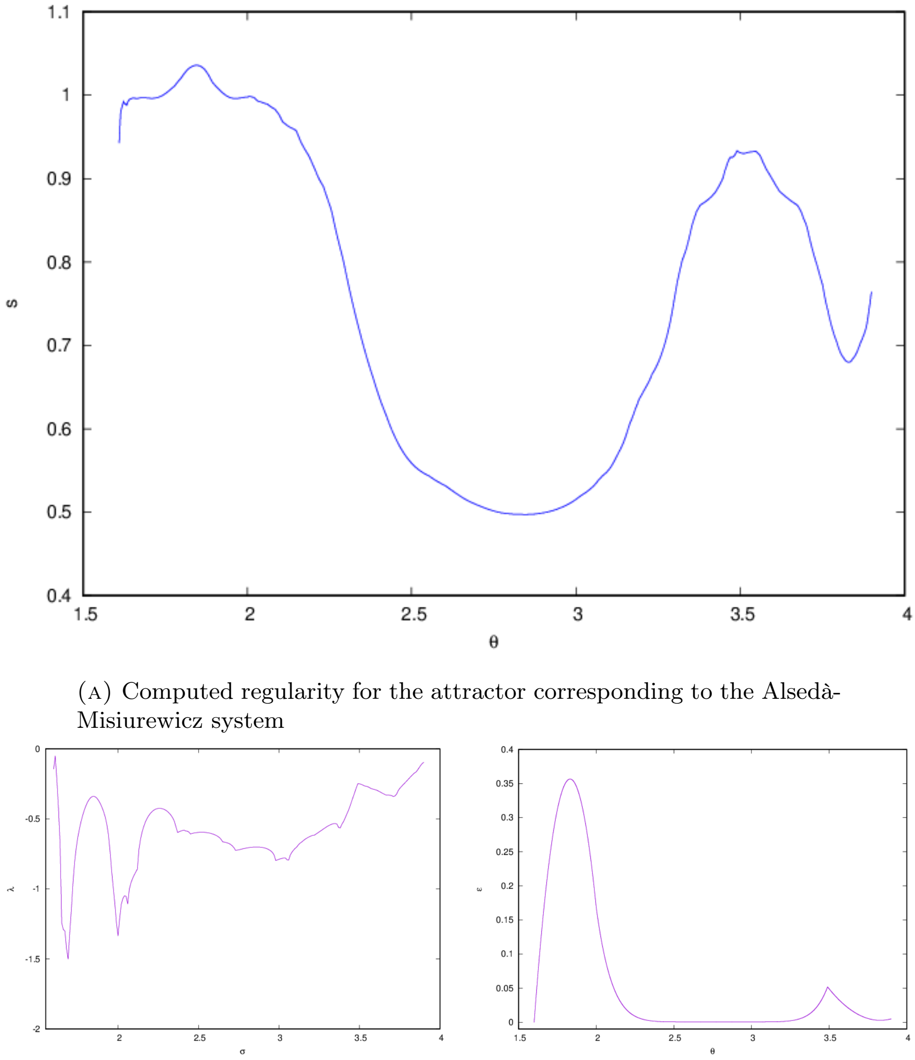


FIGURE 4.10. For the Alsedà-Misiurewic system for $\nu = 15$, $\sigma = 2$ and $\varepsilon = 0$, zoom into the area of greatest difference between the computed attractor and the recovered attractor in Figure 4.4 A and D.

The plot of $\varepsilon(\sigma)$ can be found ind Figure 4.11(C). We have computed the wavelet expansion doing a parametric sweep on σ and computed ε accordingly. Then, using Algorithm 3.38 we have computed the regularity of the attractor. In Figure 4.11(A) one can find the computed regularity using $\nu = 20$ as a function of σ. Note that when ε becomes small the regularity decreases as well. Figure 4.12(A) corresponds to the Pearson correlation coefficient obtained when computing the regularities corresponding to Figure 4.11. Moreover, one can find some examples of the linear correlation formed by $-j$ and s_{-j} corresponding to the intermediate steps in Algorithm 3.38 in this case.

After the obtained results we conjecture that the attractor for these systems is most likely regular. However, it is very complicated anyway, which limits the information that a numerical approximation may provide.

4.3.3. Assessment of the Obtained Results. The study of these two systems can be understood as a proof of concept for the Algorithm. First of all, because the recovery of the attractor from the wavelet expansion given by Equation 4.10 has been quite successful. Moreover, the fact that we have analytical results regarding the existence of SNAs has allowed us to study the behaviour of the wavelet expansion in and around the cases where we know that the attractor of the system is strange. This has shown that the strategy of using the regularity and Algorithm 3.38 to compute seems a good strategy to discern the cases in which the attractor is the graph on an upper semi-continuous function, and hence strange.



(A) Computed regularity for the attractor corresponding to the Alsedà-Misiurewicz system

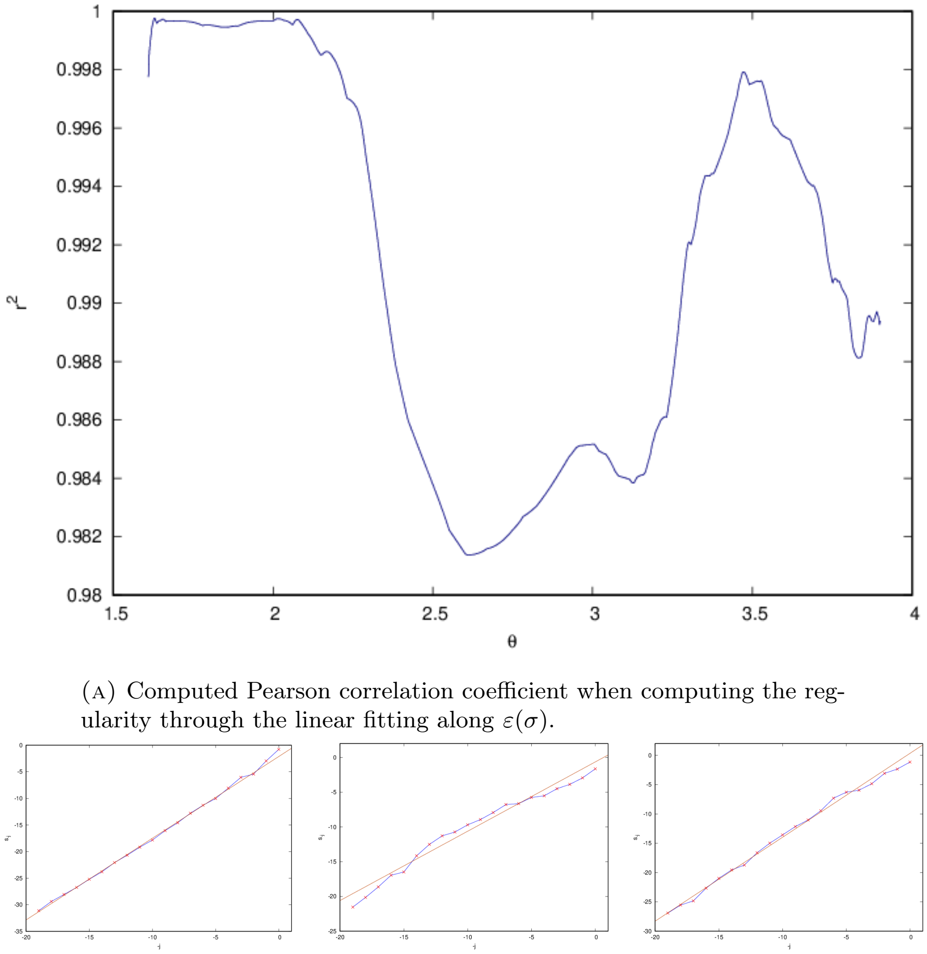
(B) The Lyapunov exponent of the attractor as a function of σ and $\varepsilon(\sigma)$

(C) Parametrization $\sigma(\varepsilon)$

Figure 4.11. Computed regularity of the attractor for the Alsedà-Misiurewicz system. Moreover, information regarding the parametrization used has also been included for the sake of clarity.

4.4. Testing the Algorithm II: The Nishikawa-Kaneko Model

In this section we finally venture into the unknown, since for this model we have no analytical results that prove or disprove the strangeness. In this case the perceived strangeness does not stem from the fact that part of the attractor lies within an invariant repellor of the system (what we call pinching with respect to $x \equiv 0$). In this case, the perceived strangeness comes from the fact that the attractor seems to be becoming increasingly complicated, what the authors call a *fractalization*. This is paired with the system being on the course of transitioning from non-chaotic to chaotic.



(A) Computed Pearson correlation coefficient when computing the regularity through the linear fitting along $\varepsilon(\sigma)$.

(B) Linear plot for $\sigma = 1.8460$, the equation of the regression line is $y = 1.535884x - 2.122176$.

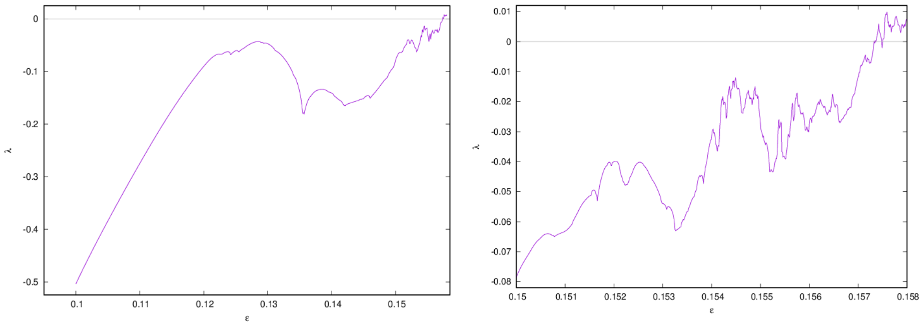
(C) Linear plot for $\sigma = 2.8460$, the equation of the regression line is $y = 0.997386x - 0.639696$.

(D) Linear plot for $\sigma = 3.5000$, the equation of the regression line is $y = 1.430911x + 0.334801$.

FIGURE 4.12. For the Alsedà-Misiurewicz system, Pearson correlation coefficient of the linear fit required in Algorithm 3.38 to compute the regularity corresponding to the parametrization $\varepsilon(\sigma)$ given by Equation (4.23). Moreover, three examples of the correlation corresponding to local extrema of r^2. In red the computed points, in orange, the computed linear regression.

In this sense, the computations become extremely tricky. The vertical Lyapunov exponent goes to zero (see Figure 4.13), meaning that the convergence to the attractor becomes increasingly slower. This poses some issues for us. For example, whenever we want to compute the seeds for the Newton method using Algorithm 4.5 the amount of iterates required for convergence becomes quite large. In this framework, rounding errors start playing a crucial role, sometimes completely invalidating the results given by iteration. Since the matrix

$$\Psi_R - \Delta_{\sigma,\varepsilon}\Psi$$



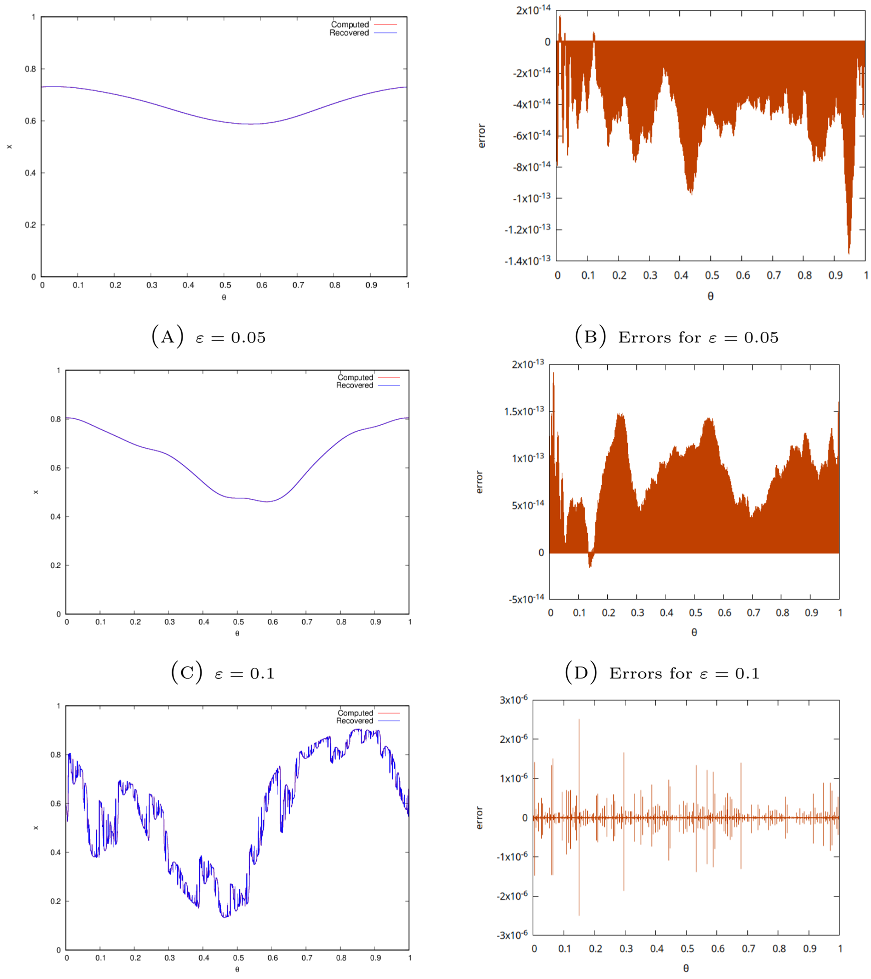
(A) Computed Lyapunov exponent for the Nishikawa-Kaneko attractor for a parametric sweep on ε around the area we studied.

(B) Zoom of the Lyapunov exponent to the area around the transition from non-chaotic to chaotic attractor. Note that the transition happens at least twice in this image. Mainly due to the sharp peak right after $\varepsilon = 0.157$ and the subsequent sharp dip that brings the exponent below zero again. The tolerance corresponding to these computations is of around 10^{-16}.

FIGURE 4.13. Computed value of the Lyapunov exponent for the Nishikawa-Kaneko attractor using Algorithm 2.8.

is especially bad conditioned in this case (see Section 4.2.1 in this Chapter), ensure that we have very good seeds for the Newton's method. In particular, continuation techniques have been unable to give consistent convergence of the method. What is even worst, for most values, using iterative methods actually worsens results compared to the ones given directly seed computed by Algorithm 4.5. Partially this stems from the fact that this seed gives excellent approximations (see the second column in Figure 4.14). Therefore, all the results here presented stem directly from the seed given by Algorithm 4.5. Partially due to this fact, we have not yet been able to obtain any wavelet expansions for what Nishikawa and Kaneko call a *fractal torus*, that is for values $0.1553 < \varepsilon < 0.1573$, because after around 0.1552 the seed obtained by Algorithm 4.5 does not give good enough approximations. This might be due to the fact that we have observed a slow but steady increase of the value of the coefficients corresponding to higher frequencies for the values of ε leading up to this point (see for example Figures 4.15 and 4.16). However, in any case this means that we have not obtained interesting results while studying this system. First of all, as in the previous two sections, we have recovered the attractor from the wavelet expansion and compared the results approximation of the attractor using Algorithm 4.5. The results can be found in Figure 4.14.

When it comes to computing regularities using Algorithm 3.38 we do see a huge change with respect to the previous two examples. While we had very clear linear correlations between $-j$ and $s_{-j} = \log_2\left(\max_{n\in\{0,\ldots,2^j-1\}}\{d_{-j,n}\}\right)$ in the Keller-GOPY system (Figure 4.8) and in the Alsedà-Misiurewicz sytem (Figure 4.12) this is not the case for this system. In Figure 4.15 one can see the strange behavior that we observe. For smaller values of ε one can see that the shape of the pairs $(-j, s_{-j})$ resembles a hockey stick. However, it seems that the flatter part corresponding to larger values of j is only numerical noise, because we have tried to manually set all the coefficients in this area to zero and the recovery of the attractor mostly remain



(A) $\varepsilon = 0.05$

(B) Errors for $\varepsilon = 0.05$

(C) $\varepsilon = 0.1$

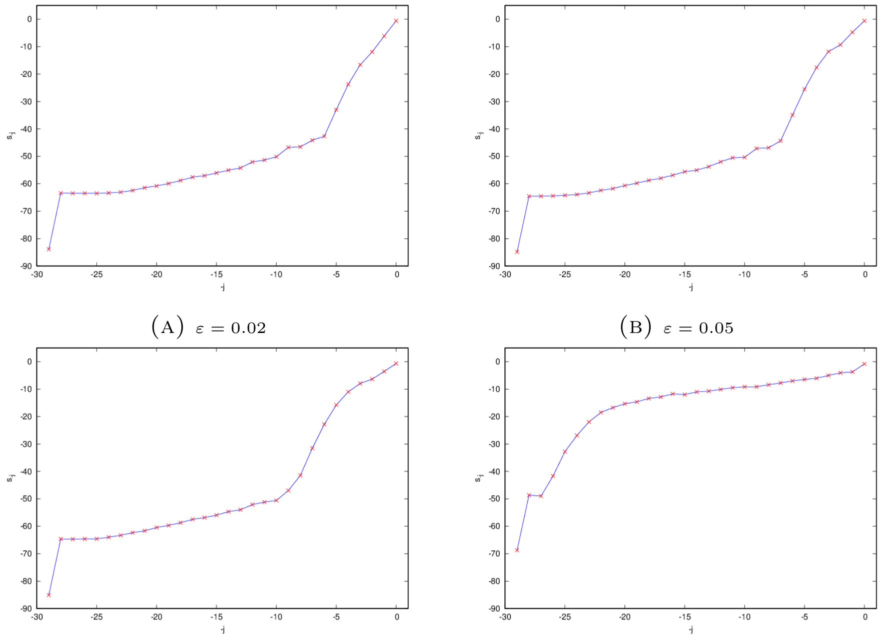
(D) Errors for $\varepsilon = 0.1$

(E) $\varepsilon = 0.155$

(F) Errors for $\varepsilon = 0.155$

FIGURE 4.14. Comparisons for the Nishikawa-Kaneko system between the computed attractor using Algorithm 4.5 and the one obtained from the reconstruction given by Equation (4.10). All plots correspond to instances of System 2.7 with $\sigma = 3$. Figures (A), (C) and (E) contain two graphs. In red the one corresponding to the direct approximation of the attractor using Algorithm 4.5. In blue the one obtained from the reconstruction given by (4.10). For both cases we have fixed $\nu = 25$, that is 2^{25} points and wavelet coefficients. Figures (B) (D) and (F) correspond to the difference *Computed-Recovered.* Note the change of scale between Figure (F) and Figures (B) and (D), since (F) is closer to what is referred as a *fractal torus*

unchanged. Hence, it would seem that the modulus of the coefficients becomes very small very fast and that we are left with a sort of numerical noise baseline. Hence to obtain the regularity for this points we would need to consider the slope of the steeper part of the graph. However, as it can be seen in Figure 4.15 at around $\varepsilon \approx 0.5$ one sees that the behaviour starts to change. What what was previously a hockey stick now becomes progressively more humpy, with a flatter part in the beginning and a more sudden drop. The interesting part about this is that the flatter part becomes wider and wider as ε approaches 0.1551, the maximum value for which we have been able to compute the wavelet expansion (see Figure 4.16). In this case, the flatter part spans the first 21 coefficients. However, we still have a change in slope after that point. Considering this behaviour, it would seem that the move from non-strangeness to alleged strangeness is a process that starts with the increase in value of the coefficients of low frequency, and that as ε increases more and more of the coefficients corresponding to higher frequencies become larger. However, contrary to what happened in Section 4.3 this is always a movement from lower to higher frequencies, not a sudden jump in the higher frequencies.



(A) $\varepsilon = 0.02$

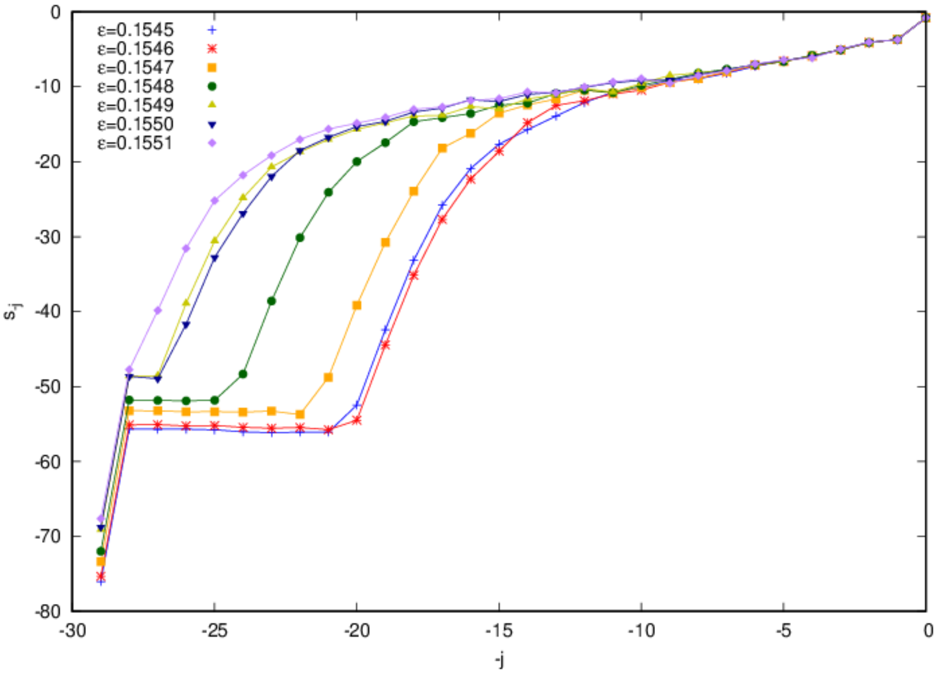
(B) $\varepsilon = 0.05$

(C) $\varepsilon = 0.1$

(D) $\varepsilon = 0.155$

FIGURE 4.15. Graphs of $(-j, s_{-j})$ for the intermediate steps of Algorithm 3.38 for the Nishikawa-Kaneko system. Note the start of a Hump for $\varepsilon \approx 0.5$ that becomes clearer as ε increases. The speed in which the hump becomes more pronounced can be better seen in Figure 4.16, where we have plotted this graphs for some coefficients leading up to $\varepsilon = 0.155$ at once. The flatter tails correspond to numerical noise, since we have set all the coefficients for large values of j to zero and the recovery of the attractor barely changed.

All of this also begs the question, which slope one should take when applying Algorithm 3.38, since either the slope of the flatter part or the slope of the steeper part at each side of the hump would fall within the possibilities of Theorem 3.35. However, provided that the behaviour for the higher frequency coefficient follows the one observed in the steeper part, it would seem that the results obtained from doing the linear fitting on this side are more optimal, since s would be larger. Considering this, the results obtained can be found in Figure 4.16.



ε	regularity (flat)	regularity (steep)
0.1545	0.234988	7.819649
0.1546	0.246864	6.162333
0.1547	0.135883	7.331269
0.1548	0.150465	5.815172
0.1549	0.095662	7.830169
0.1550	0.107555	8.414967
0.1551	0.107555	7.797493

FIGURE 4.16. A more careful look at the possible computations of regularity regarding the Nishikawa-Kaneko model. Note the disparity between the regularities computed using the steeper part and the one computed with the flatter one. If one considered the former as valid, then the attractor cannot be considered strange. Otherwise, the regularity falls within the gray zone described in Section 3.4 and we cannot decide anything.

4.5. Some comments on the results

When it comes to the computations, we would like to summarize some of the problems and particularities that we have encountered and that are common throughout all computations in this thesis.

First of all, regarding the bottlenecks in the computation we have mainly identified three of them:

(i) The computation of the initial seed of the system. The main reason is that Algorithm 4.5 requires to iterate the function n_0 times on each $\theta_i = i/N$, which is dependent on the Lyapunov exponent. Hence, the reason of this bottleneck is twofold. On one hand, if the Lyapunov exponent is small the number of iterates n_0 to obtain a good approximation might become very large. On the other hand, As N increases the number of points for which this sampling must be done increases. Hence, at the end of the day one needs to do $n_0 \cdot N$ iterations to obtain a good approximation. For a Lyapunov exponent of $\lambda_\varphi \approx -0.01$ (which is not out of the ordinary for the Nishikawa-Kaneko system) $n_0 \approx 5 \cdot 10^5$. On the other hand, if $\nu = 25$, $N = 2^\nu = 33,554,432$. Hence, we might be talking of having to perform around three trillion iterations for a single seed. Parallelization techniques might help, but the computational time required is still demanding.

(ii) The product of a matrix and a vector. Despite our efforts to implement all the matrix operations in an efficient manner, if we are dealing with very large N the operations still take a very long time. Despite all the simplifications the algorithm still scales close to $\mathcal{O}(N^2)$, hence the times become very big very fast. Partially this stems from the fact that the way we have decided to store the matrices is optimized for reducing the storage required to save them, as it can be seen in Annex A. This, however, imposes quite a few restrictions on the matrix operations, since we need to recover a whole matrix from a fraction of the information stored in it.

(iii) Lack of regularity complicates the convergence of the iterative solvers. As we have commented in Subsection 4.2.1 as the system looses regularity the matrices to be solved have worse condition numbers. Hence, the computation tend to work flawlessly for more regular systems struggle quite a bit for the non-regular case. Moreover, the seed given by Equation (4.14) does not perform as well for cases with lower regularity. So one might find oneself in a position where a lot of time is dedicated to compute a mediocre seed for the Newton method.

To try to deal with problems (i) and (iii), for the Keller-GOPY and Alsedà-Misiurewicz systems we have resolved to use continuation techniques to obtain the wavelet expansion, especially around $\varepsilon = 0$, but also for positive regularities. The results are somewhat better, since we need to compute fewer new seeds for newton, as the previous result works as a seed for the newer one. However, in light of what has been shown in Figure 4.6 and Table 4.1, the approach must be done with very fine steps, since around $\varepsilon = 0$ slight variations in the value of ε mean noticeable changes in the wavelet expansion. On the other hand, as stated on Section 4.4 the continuation approach has not given any meaningful results when dealing with the Nishikawa-Kaneko model.

The system that requires furthest study, though, is the Nishikawa-Kaneko one. We have been unable to find a good enough preconditioner to help solve the system in a reliable manner. In fact, as stated in Subsection 4.4 all results have been obtained directly from the seed. As already explained in Section 4.4, the seed has been so good that it already gave an acceptable solution without any further iterations (see for example Figure 4.14). It remains to be seen if a good preconditioner may unlock the possibility of using continuation or further refinement the *fractal torus* case.

CHAPTER 5

Computing the Wavelets Matrices

In this chapter we give a thorough explanation on how to compute the wavelet matrices Ψ and Ψ_R from Chapter 3 Section 4.1. To this end we will present various implementations of the Daubechies-Lagarias algorithm [**DL91, DL92**] suited to our needs. This chapter has two main parts, one centred on evaluating a Daubechies wavelet with p vanishing moments on dyadic points and another revolving on evaluating the same wavelet on a dyadic points plus an irrational rotation. To fully understand this section one needs to have in mind the introduction to Daubechies wavelets from Section 3.2.

An important note is that the results presented in this chapter, irrespective of their theoretical interested, are implementation-focused. That is, they are presented in a manner that facilitates their translation to any programming language. In particular, since we are dealing with a large volume of computations and memory usage the underlying assumption is that the programming language will be low-level and compiled. In particular, we have used the C programming language for the implementations. To this end, the main result of every section will have two versions, a general one and, for lack of a better term, a *dyadic* one. This stems from the fact that to evaluate the wavelet at a point x the Daubechies-Lagarias algorithm requires for us to compute the dyadic expansion of $\{\{2x\}\}$. Since we are mostly interested in dyadic rationals $\{\{2x\}\}= \frac{m}{2^n}$, $0 \leq m < 2^n$ (or rotated ones, but that is better left for Section 5.5), the dyadic expansion of $\{\{2x\}\}$ can be easily inferred from the binary expansion of m. Moreover, since our goal is an implementation in C, the binary expansion of m is the default way the computer stores m. Hence, using only very efficient bitwise operators one can do computations that otherwise would be neither as easy nor as fast. In the case of rotated wavelet values in Section 5.5, we have adapted the algorithm to compute $\psi_{i,j}^{\mathrm{PER}}(R_\omega(\theta))$ while still being able to operate in the dyadic-as-integer frame. This however, turns out to be more involved, and it is better left for a comprehensive discussion in that section.

5.1. Notation and Definitions

For the sake of completeness within this chapter, let us fix some definitions anew. By default we will consider that all vectors are written as column vectors. Moreover $\overrightarrow{\mathbf{1}_k}$ will denote the vector of dimension k with all its components equal to 1. We will use $\{\{\cdot\}\}$, $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ to for the fractional part, the ceiling function and the floor function respectively. Furthermore, let us recall some basic notions.

Definition 5.1 (Binary expansion). Let $q \in \mathbb{N}$. Then we can write

$$q = a_\ell 2^\ell + a_{\ell-1}2^{\ell-1} + \cdots + a_1 2 + a_0.$$

with $a_i \in \{0,1\}$, $0 \leq i < \ell$, and $a_\ell = 1$. Note that q is odd if and only if $a_0 = 1$. The string

$$\mathsf{binary}(q) := a_\ell a_{\ell-1} \dots a_1 a_0$$

is called the *binary expansion of q*. *The length of the binary expansion* is by definition exactly $\ell + 1$ and will be denoted by $\mathsf{length}(\mathsf{binary}(q))$. Observe that $\mathsf{length}(\mathsf{binary}(q)) \leq \ell$ if and only if $q < 2^{\ell+1}$. ∎

Now, very closely related to the previous definition we have the definition of the Dyadic expansion of a real number $x \in [0, 1)$.

Definition 5.2 (Dyadic expansion). Given a real number $x = \sum_{i=1}^{\infty} \frac{a_i}{2^i} \in [0, 1)$ with $a_i \in \{0, 1\}$ we will call the sequence $a_1 a_2 \ldots a_n \ldots$ the *the dyadic expansion of x* and denote it by $\mathsf{dyad}(x)$. Also, $\mathsf{dyad}(x, i) = a_i$ denotes the i-th digit of the dyadic expansion of x. We say that $0 \leq x < 1$ is a dyadic rational if the dyadic expansion is finite, i.e. $x = \frac{m}{2^\nu}$ for some $0 \leq m < 2^\nu$. ∎

Note that if $x = \frac{m}{2^\nu}$, for some $\nu \in \mathbb{N}$ and $0 \leq m < 2^\nu$, then

$$\mathsf{dyad}(x) = \mathbf{0} \ldots \mathbf{0}\, \mathsf{binary}(m).$$

5.2. The Daubechies-Lagarias Algorithm

In this section we will introduce the Daubechies-Lagarias algorithm [**DL91, DL92**]. This algorithm allows the computation of the value of a Daubechies wavelet at an arbitrary point. As has been shown in Section 3.2 this is far from trivial. Since the construction of the Daubechies wavelet family seek to minimize the support for a given number of vanishing moments, these wavelets are not optimized for computational ease.

Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p > 1$ vanishing moments with scaling filter h. We define two $(2p-1) \times (2p-1)$ matrices $\mathbf{M}_0$ and $\mathbf{M}_1$, where

$$(\mathbf{M}_0)_{i,j} = \sqrt{2} h[2i - j - 1] \quad \text{and} \quad (\mathbf{M}_1)_{i,j} = \sqrt{2} h[2i - j]$$

for $i, j \in \{1, 2, \ldots, 2p - 1\}$. Recall that we consider $\mathrm{supp}(h) = [0, 2p - 1]$.

Example 5.3. Consider the $\mathbb{R}$-Daubechies wavelet with $p = 2$ vanishing moments. Then we have

$$h = \left(\frac{1+\sqrt{3}}{4\sqrt{2}}, \frac{3+\sqrt{3}}{4\sqrt{2}}, \frac{3-\sqrt{3}}{4\sqrt{2}}, \frac{1-\sqrt{3}}{4\sqrt{2}} \right)$$

Hence we get the matrices

$$\mathbf{M}_0 = \sqrt{2} \begin{pmatrix} h[0] & 0 & 0 \\ h[2] & h[1] & h[0] \\ 0 & h[3] & h[2] \end{pmatrix} = \begin{pmatrix} \frac{1+\sqrt{3}}{4} & 0 & 0 \\ \frac{3-\sqrt{3}}{4} & \frac{3+\sqrt{3}}{4} & \frac{1+\sqrt{3}}{4} \\ 0 & \frac{1-\sqrt{3}}{4} & \frac{3-\sqrt{3}}{4} \end{pmatrix}, \text{ and}$$

$$\mathbf{M}_1 = \sqrt{2} \begin{pmatrix} h[1] & h[0] & 0 \\ h[3] & h[2] & h[1] \\ 0 & 0 & h[3] \end{pmatrix} = \begin{pmatrix} \frac{3+\sqrt{3}}{4} & \frac{1+\sqrt{3}}{4} & 0 \\ \frac{1-\sqrt{3}}{4} & \frac{3-\sqrt{3}}{4} & \frac{3+\sqrt{3}}{4} \\ 0 & 0 & \frac{1-\sqrt{3}}{4} \end{pmatrix}.$$

∎

Now we have all the tools required to prove the Daubechies-Lagarias main result, which leads to the various algorithms we will use to evaluate wavelets on a single point.

Theorem 5.4 (Theorem 2.2 and subsequent remarks in [**DL92**]). *Let ϕ be the scaling function of an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments. Then*

$$\lim_{n\to\infty} \prod_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}(x,i)} \tag{5.1}$$

exists for every $x \in [0,1)$, and it is equal to

$$\begin{pmatrix} \phi(x) & \phi(x) & \dots & \phi(x) \\ \phi(x+1) & \phi(x+1) & \dots & \phi(x+1) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(x+2p-2) & \phi(x+2p-2) & \dots & \phi(x+2p-2) \end{pmatrix}$$

Example 5.5. Following with Example 5.3, we can compute the products of the $\mathbf{M}_0$ and $\mathbf{M}_1$ matrices for $p = 2$ from before. In particular we will compute the limits of the products for $x = 0$ and $x = \frac{1}{2}$.

- $x = 0$, this means that $\mathsf{dyad}(x) = \mathbf{0}^\infty$, by numerically approximating the product of infinitely many matrices we obtain,

$$\lim_{n\to\infty} \mathbf{M}_0^n \approx \begin{pmatrix} 0 & 0 & 0 \\ 1.3660254 & 1.3660254 & 1.3660254 \\ -0.3660254 & -0.3660254 & -0.3660254 \end{pmatrix}.$$

- $x = \frac{1}{2}$, this means that $\mathsf{dyad}(x) = \mathbf{10}^\infty$, by using the previous result we obtain,

$$\lim_{n\to\infty} \mathbf{M}_1\mathbf{M}_0^n \approx \begin{pmatrix} 0.9330127 & 0.9330127 & 0.9330127 \\ 0 & 0 & 0 \\ 0.0669872 & 0.0669872 & 0.0669872 \end{pmatrix}.$$

■

Using Theorem 5.4 and Remark 3.8 (a) we can compute the value of ψ at an arbitrary point.

Corollary 5.6. *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments and let h be its scaling filter. Then, for every $x \in \mathbb{R}$ we have*

$$\psi(x) = -(-1)^{d(x)} \tfrac{\sqrt{2}}{2p-1}\, u(x)^{\mathsf{T}} \left(\lim_{n\to\infty} \prod_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}(\{\!\{2x\}\!\},i)} \right) \vec{\mathbf{1}}, \tag{5.2}$$

where $d(x) = 1 - \lfloor 2x \rfloor \in \mathbb{Z}$, and

$u(x)^{\mathsf{T}} = \big(h[d(x)], -h[1+d(x)], h[2+d(x)], \dots, -h[2p-3+d(x)], h[2p-2+d(x)]\big)$.

Proof. Denote by $\mathbf{\Phi}(x)$ the vector

$$\Big(\phi\big(\{\!\{2x\}\!\}\big), \phi\big(\{\!\{2x\}\!\}+1\big), \phi\big(\{\!\{2x\}\!\}+2\big), \dots, \phi\big(\{\!\{2x\}\!\}+2p-2\big)\Big)^{\mathsf{T}}.$$

We have

$$\begin{aligned}
\psi(x) &= \sqrt{2} \sum_{n\in\mathbb{Z}} (-1)^{1-n} h[1-n]\, \phi(2x-n) \quad \text{Equation (3.10)} \\
&= \sqrt{2} \sum_{n\in\mathbb{Z}} (-1)^{1-n}\, h[1-n]\, \phi\big(\{\!\{2x\}\!\}+\lfloor 2x \rfloor - n\big) \\
&= \sqrt{2} \sum_{m\in\mathbb{Z}} -(-1)^{m+d(x)}\, h\big[m+d(x)\big]\, \phi\big(\{\!\{2x\}\!\}+m\big) \quad \text{Re-indexing by } m = \lfloor 2x \rfloor - n \\
&= -(-1)^{d(x)}\sqrt{2} \sum_{m=0}^{2p-2} (-1)^m\, h\big[m+d(x)\big]\, \phi\big(\{\!\{2x\}\!\}+m\big) \quad \operatorname{supp}(\phi) = [0, 2p-1] \\
&= -(-1)^{d(x)}\sqrt{2}\, u(x)^{\mathsf{T}} \mathbf{\Phi}(x) \quad \text{Matricial formula of scalar product} \\
&= -(-1)^{d(x)}\sqrt{2}\, u(x)^{\mathsf{T}} \left(\tfrac{1}{2p-1} \left(\lim_{n\to\infty} \prod_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}(\{\!\{2x\}\!\},i)} \right) \vec{\mathbf{1}} \right) \quad \text{Theorem 5.4} \\
&= -(-1)^{d(x)} \tfrac{\sqrt{2}}{2p-1}\, u(x)^{\mathsf{T}} \left(\lim_{n\to\infty} \prod_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}(\{\!\{2x\}\!\},i)} \right) \vec{\mathbf{1}},
\end{aligned}$$

□

Definition 5.7. Note that in the evaluation of a Daubechies wavelet in Corollary 5.6 there are two sides on the formula, hence it makes sense to give a proper name to each half. On one hand we have *the vector* $u(x)$. On the other hand, the limit of products of matrices,

$$\mathbf{DL}\big(\{\!\{2x\}\!\}\big) := \lim_{n\to\infty} \prod \mathbf{M}_{\mathsf{dyad}(\{\!\{2x\}\!\},i)},$$

which we will refer to as the *matrix product at* x. ■

Remark 5.8. Note that since we are dealing with Daubechies wavelets the filter h has support equal to $[0, 2p-1]$. Hence,

$$h[i+d(x)] = 0 \quad \text{whenever} \quad i+d(x) \notin [0, 2p-1].$$

■

5.3. How to Compute the Value of a Daubechies Mother Wavelet for all Dyadic rationals

In this section we will explore a very fast implementation of the Daubechies-Lagarias algorithm stemming from Corollary 5.6 to evaluate a Daubechies mother wavelet on dyadic points. For us, this works mostly as a motivation for the next sections, in which we will compute the values of periodized wavelets on the same set of dyadic points. However, we feel that this implementation is useful on its own right: many implementations in which wavelet values need to be computed at an arbitrary point simply use the *cascade algorithm* [**Dau92**] to obtain a the wavelet value on a dyadic mesh and then use interpolation [**Boo**]. We believe that the implementation of this algorithm might lead to much lower computation times while having a higher precision. For our implementation on obtaining the wavelet expansions of SNAs, obtaining the wavelet value at a point with maximum precision was a top priority, therefore we chose a Daubechies-Lagarias implementation over the cascade algorithm. However, it turned out that the computations became extremely fast in the process.

5.3.1. How to efficiently approximate the matrix products for dyadic x's. In this subsection we shall see how to efficiently compute the matrix products for x, i.e.

$$\mathbf{DL}\big(\{\!\{2x\}\!\}\big) = \lim_{n\to\infty} \prod \mathbf{M}_{\mathsf{dyad}(\{\!\{2x\}\!\},i)}$$

when x is a dyadic rational. Let us first formalize this notion of dyadic rationals of order ν.

Definition 5.9. Let $\nu \in \mathbb{N}$. We define the set of *dyadic rationals of order* ν as the set

$$\left\{\frac{r}{2^\nu} : 0 \le r < 2^\nu\right\} = \{0\} \cup \left\{\frac{2^m q}{2^\nu} : 0 \le m < \nu, q \in \mathbb{N} \text{ odd, and } 2^m q < 2^\nu\right\}$$

■

As we already have stated in the introduction to this chapter, in the case of dyadic rationals of order ν computing $\mathsf{dyad}(\{\!\{2x\}\!\}, i)$ is equivalent to computing the ith element of $\mathsf{binary}(2^{m+1}q)$.

Moreover, note that we only need to concern ourselves with $x \in \left[0, \frac{1}{2}\right)$, since any number x can be written as $x = \lfloor x \rfloor + \{\!\{x\}\!\}$, hence $\{\!\{2x\}\!\} = \{\!\{2\{\!\{x\}\!\}\}\!\}$. Moreover, since $\{\!\{2\left(x+\frac{1}{2}\right)\}\!\} = \{\!\{2x\}\!\} = 2x$, it follows that for $x \in \left[0, \frac{1}{2}\right)$ the matrix product at x equals the matrix product at $x+\ell+\frac{1}{2}$, $\ell \in \mathbb{Z}$. Therefore, if we want to compute the matrix product at x for all dyadic rationals we only need to compute the matrix product for $x = 0$ and for

$$x = \frac{2^m q}{2^\nu} < \frac{1}{2} = \frac{2^{\nu-1}}{2^\nu} \tag{5.3}$$

Remark 5.10. Note that $2^m q < 2^{\nu-1}$ implies that $m < \nu - 1$ and $q < 2^{\nu-m-1}$. Therefore

$$\mathsf{length}(\mathsf{binary}(q)) = \ell + 1 \leq \nu - m - 1.$$

■

Under the assumption of Equation (5.3), we have that

$$\left\{\!\!\left\{ 2\left(\frac{2^m q}{2^\nu}\right) \right\}\!\!\right\} = \frac{2^{m+1} q}{2^\nu} = \frac{q}{2^{\nu-m-1}}.$$

Hence we can write the dyadic expansion of $\{\{2x\}\}$ as

$$\mathbf{0}^{\lambda(q,m,\nu)} 1 a_{l-2} \cdots a_1 a_0 \mathbf{0}^\infty, \tag{5.4}$$

where $\mathbf{0}^{\lambda(q,m,\nu)}$ is a sequence of $\lambda(q,m,\nu) = (\nu - 1) - (m + \mathsf{length}(\mathsf{binary}(q))) \geq 0$ zeros, $l = \mathsf{length}\left(\mathsf{binary}(q)\right)$ and $\mathsf{binary}(q) = 1a_{l-2}a_{l-3}\cdots a_1 a_0$, and $\mathbf{0}^\infty$ denotes an infinite sequence of zeros. Now, we have enough information to compute the matrix products for any $x \in [0, 1/2)$.

$$\begin{aligned} \mathbf{DL}\big(\{\{2x\}\}\big)\vec{\mathbf{1}} &= \lim_{n\to\infty} \mathbf{M}_{\mathsf{dyad}(2x,0)} \cdot \mathbf{M}_{\mathsf{dyad}(2x,1)} \cdots \mathbf{M}_{\mathsf{dyad}(2x,n)} \cdot \vec{\mathbf{1}} = \\ &\mathbf{M}_0^{\lambda(q,m,\nu)} \cdot \mathbf{M}_1 \cdot \mathbf{M}_{a_{l-2}} \cdot \mathbf{M}_{a_{l-3}} \cdots \mathbf{M}_{a_1} \cdot \mathbf{M}_{a_0} \cdot \Big(\lim_{n\to\infty} \mathbf{M}_0^n \cdot \vec{\mathbf{1}}\Big). \end{aligned} \tag{5.5}$$

Since the matrix $\mathbf{M}_0$ has 1 as a simple dominant eigenvalue (Theorem 2.2 in [**DL92**]) by the power method we have that

$$\lim_{n\to\infty} \mathbf{M}_0^n \vec{\mathbf{1}} = \vec{\mathbf{V}}^{(0)} = \vec{\mathbf{V}}^{(0)}.$$

Where $\vec{\mathbf{V}}^{(0)}$ is an eigenvector of eigenvalue 1 such that

$$\vec{\mathbf{1}} = \vec{\mathbf{V}}^{(0)} + \sum_{i=1}^{2p-2} c_i \vec{\mathbf{V}}_i(0),$$

where $c_i \in \mathbb{R}$ and $\vec{\mathbf{V}}_i(0)$ are the other eigenvectors of M_0. Now we can rewrite Equation (5.5) as

$$\mathbf{DL}\big(\{\{2x\}\}\big)\vec{\mathbf{1}} = \mathbf{M}_0^{\lambda(q,m,\nu)} \cdot \mathbf{M}_1 \cdot \mathbf{M}_{a_{l-2}} \cdot \mathbf{M}_{a_{l-3}} \cdots \mathbf{M}_{a_1} \cdot \mathbf{M}_{a_0} \cdot \vec{\mathbf{V}}^{(0)}. \tag{5.6}$$

In particular when $x = 0$ we get

$$\mathbf{DL}(0)\vec{\mathbf{1}} == \lim_{n\to\infty} \mathbf{M}_0^n \cdot \vec{\mathbf{1}} = \vec{\mathbf{V}}^{(0)}. \tag{5.7}$$

Note that a possible conclusion of this subsection is that if one divides the real line by semi-integer intervals (i.e. $\ldots, [-n, -n+1/2), \ldots, [0, 1/2), [1/2, 1), \ldots, [n, n+1/2), \ldots$) the information given by $\mathbf{DL}\big(\{\{2x\}\}\big)$ is the same for each interval, since given $y \in \operatorname{supp}(\psi)$ it can be written as either $y = x + \frac{1}{2} + \ell$ or $y = x + \ell$ with $x \in [0, 1/2)$ and $\ell \in \{1-p, 2-p, \ldots, -1, 0, 1 \ldots p-1, p\}$ we have,

$$\begin{aligned} \mathbf{DL}\big(\{\{2y\}\}\big) &= \lim_{n\to\infty} \prod\nolimits_{i=1}^n \mathbf{M}_{\mathsf{dyad}(\{\{2y\}\},i)} = \\ &\lim_{n\to\infty} \prod\nolimits_{i=1}^n \mathbf{M}_{\mathsf{dyad}(\{\{2x\}\},i)} = \mathbf{DL}\big(\{\{2x\}\}\big). \end{aligned} \tag{5.8}$$

In a way, the matrix products only care about the (semi-)fractional part of x. As we shall see in the following subsection, it will be the vector $u(x)$ that will contain the information regarding the (semi-)integral part of x. Hence, splitting Corollary 5.6 into two parts, one for the matrix products and another one for the vectors u, is not only done for convenience, but it points to a deeper duality within the computation of wavelets.

5.3.2. How to compute the vector $u(x)$. Only the vector $u(x)$ remains to be computed in order to be able to evaluate the mother Daubechies wavelet at a point x. As we will see, $u(x)$ complements the information missing in the matrix products, because it is where de dependence on the (semi-)integral part of x is found.

Recall that a Daubechies wavelet ψ with p vanishing moments has support exactly $[1-p,p]$ (Theorem 3.10). As we saw in the previous subsection, it is useful to divide the whole support of the wavelet in intervals of length $\frac{1}{2}$:

$$\begin{aligned} \operatorname{supp}(\psi) =&[1-p,p] = \bigcup_{\ell=1-p}^{p-1} [\ell,\ell+1) \cup \{p\} \\ &= \bigcup_{\ell=1-p}^{p-1} \Big(\{\ell\} \cup \big(\ell,\ell+\tfrac{1}{2}\big) \cup \big[\ell+\tfrac{1}{2},\ell+1\big)\Big) \cup \{p\}. \end{aligned}$$

In the previous subsection we have already seen that $\mathbf{DL}\big(\{\!\{2y\}\!\}\big)$, $y \in \operatorname{supp}(\psi)$ only depends on the (semi-)fractional part of y. Hence, we shall focus on the computation of $u(y)$. First we need to compute the values of $d(y)$ in terms of $x \in \big[0,\frac{1}{2}\big)$ and ℓ, where either $y = x+\ell$ or $y = x+\ell+\frac{1}{2}$. Recall that $d(y) = 1 - \lfloor 2y \rfloor$ is as in Corollary 5.6.

$$\begin{aligned} d(x+\ell) &= 1 - \lfloor 2x+2\ell \rfloor = 1 - \lfloor 2x \rfloor - 2\ell = 1-2\ell = d(\ell), \text{ and} \\ d\big(x+\tfrac{1}{2}+\ell\big) &= 1 - \lfloor 2x+1+2\ell \rfloor = -\lfloor 2x \rfloor - 2\ell = -2\ell = d\big(\tfrac{1}{2}+\ell\big). \end{aligned}$$

It is to be observed that the value of $d(y)$ depends only on the semi-integral subinterval in which it is found. In particular, the term $-(-1)^{d(y)}$ will be positive if $y = x+\ell$ and negative if $y = x+\ell+\frac{1}{2}$. Finally, by the definition of $u(x)$ stated in Corollary 5.6 we get

$$\begin{aligned} u(x+\ell)^{\mathsf{T}} &= \big(h[d(x+\ell)], -h[1+d(x+\ell)], \dots, h[2p-2+d(x+\ell)]\big) \\ &= \big(h[1-2\ell], -h[2-2\ell], \dots, h[2p-1-2\ell]\big) \\ &= u(\ell)^{\mathsf{T}}, \text{ and} \\ u\big(x+\tfrac{1}{2}+\ell\big)^{\mathsf{T}} &= \big(h[-2\ell], -h[1-2\ell], \dots, h[2p-2-2\ell]\big) = u\big(\tfrac{1}{2}+\ell\big)^{\mathsf{T}}. \end{aligned} \tag{5.9}$$

Then, by Corollary 5.6 and its proof, and the results from the previous subsection we get

COROLLARY 5.11 (To Corollary 5.6). *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments and let h be its scaling filter. Then, for every $\ell \in \mathbb{Z}$ and $x \in \big(0,\frac{1}{2}\big)$ we have*

$$\begin{aligned} \psi(\ell) &= \tfrac{\sqrt{2}}{2p-1}\ u(\ell)^{\mathsf{T}} \overrightarrow{\mathbf{V}}^{(0)}, \\ \psi(x+\ell) &= \tfrac{\sqrt{2}}{2p-1}\ u(\ell)^{\mathsf{T}} \left(\lim_{n\to\infty} \prod\nolimits_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}(2x,i)}\right) \vec{\mathbf{1}}, \\ \psi\big(\tfrac{1}{2}+\ell\big) &= -\tfrac{\sqrt{2}}{2p-1}\ u\big(\tfrac{1}{2}+\ell\big)^{\mathsf{T}} \overrightarrow{\mathbf{V}}^{(0)}, \ \textit{and} \\ \psi\big(x+\tfrac{1}{2}+\ell\big) &= -\tfrac{\sqrt{2}}{2p-1}\ u\big(\tfrac{1}{2}+\ell\big)^{\mathsf{T}} \left(\lim_{n\to\infty} \prod\nolimits_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}(2x,i)}\right) \vec{\mathbf{1}}. \end{aligned}$$

Note that this Corollary clearly shows the dichotomy with which we ended the last subsection. On one hand, the matrix products give us the information about the fractional (or the *halftional*, in a sense) part of the value of the wavelet, while the vector $u(x)$ gives the information on the postion of x within the vector $[1-p,p]$.

Example 5.12. Now that we have all to tools to compute the value of a Daubechies wavelet at a point, let us finally compute $\psi\left(\frac{1}{4}\right)$ for ψ with two vanishing moments (we simply follow through with Examples 5.3 and 5.5). First of all, we have already computed $\lim_{n\to\infty}\prod_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}(2x,i)}$ in Example 5.5, since $2x=\frac{1}{2}$ in our case.

As for the for the vector $u(0)$ since $d\left(\frac{1}{4}\right)=1-\left\lfloor\frac{1}{2}\right\rfloor=1$

$$
\begin{aligned}
u\left(\frac{1}{4}\right)^{\mathsf{T}} &= u(0)^{\mathsf{T}} = (h[1],-h[2],h[3])\\
&= \left(\frac{3+\sqrt{3}}{4\sqrt{2}},\frac{3-\sqrt{3}}{4\sqrt{2}},\frac{1-\sqrt{3}}{4\sqrt{2}}\right).
\end{aligned}
$$

Hence

$$
\begin{aligned}
\psi\left(\frac{1}{4}\right) =& \frac{\sqrt{2}}{3}\left(\frac{3+\sqrt{3}}{4\sqrt{2}},\frac{3-\sqrt{3}}{4\sqrt{2}},\frac{1-\sqrt{3}}{4\sqrt{2}}\right)\begin{pmatrix}0.9330127 & 0.9330127 & 0.9330127\\ 0 & 0 & 0\\ 0.0669872 & 0.0669872 & 0.0669872\end{pmatrix}\vec{\mathbf{1}}\\
\approx& 1.0915063.
\end{aligned}
$$

As a comparison, the BoostC++ [**Boo**] library implementation gives a value of $\psi\left(\frac{1}{4}\right)=1.0915063$ ■

5.3.3. Algorithm to compute ψ by using Corollary 5.11 and Equation (5.6). Finally, we are able to give the outline of our algorithm to compute the wavelet values on all the dyadic points at once. This will be done by carefully re-using as many computations as possible. This allows us to plot any Daubechies mother wavelet or store these values to do further computations or interpolations. In particular, since we have that

$$
\psi_{j,n}(x)=2^{-j/2}\psi\left(\frac{x-2^{j}n}{2^{j}}\right)
$$

it follows that if $x=\frac{m}{2^{\nu}}$, then $\frac{x-2^{j}n}{2^{j}}=\frac{m-2^{j+\nu}n}{2^{j+\nu}}$, which is still a dyadic rational. Hence the values of $\psi_{j,n}(x)$ on a dyadic rational, can be computed more or less directly by having a mesh of values of $\psi(x)$ on dyadic rationals of order $2^{j+\nu}$ (recall that j might be negative, hence $j+\nu<\nu$). Even though we will not be using this strategy it may be useful to keep it in mind.

When it comes to the actual implementation of Corollary 5.6 as an algorithm, we need to take into account that by Equation (5.8) $\mathbf{DL}\big(\{\{2x\}\}\big)=\mathbf{DL}\big(\{\{2y\}\}\big)$ if $x-y=k\frac{1}{2}$, for some $k\in\mathbb{Z}$. Hence, our goal is to compute $\mathbf{DL}\big(\{\{2x\}\}\big)$ for all dyadic $x\in[0,1/2)$ and then multiply them for the various $u(\ell)$ and $u\left(\ell+\frac{1}{2}\right)$ as needed to obtain all the dyadic points within the support. Moreover, we will compute the matrix products $\mathbf{DL}\big(\{\{2x\}\}\big)$ as efficiently as possible. Recall that from (5.3) we have to compute the dyadic expansion of all numbers of the form

$$
\frac{2^{m}q}{2^{\nu}}=\frac{q}{2^{\nu-m}},\ \text{where } m<\nu \text{ and } q<2^{\nu-m-1}.
$$

For this numbers one can compute This leads to one of the most important simplifications. Note that if we have two numbers $x_1=\frac{2^{m}q}{2^{\nu}}$ and $x_2=\frac{2^{m'}q}{2^{\nu}}$, with $m<m'$ then by equation (5.4) we have

$$
\begin{aligned}
\mathsf{dyad}(x_1) &= \overbrace{\mathbf{0}\ldots\mathbf{0}}^{m-m'}\ \overbrace{\mathbf{0}\ldots\mathbf{0}}^{\lambda(q,m',\nu)}\ 1a_{l-2}\cdots a_1a_0\mathbf{0}^{\infty},\ \text{and}\\
\mathsf{dyad}(x_2) &= \underbrace{\mathbf{0}\ldots\mathbf{0}}_{\lambda(q,m',\nu)}\ 1a_{l-2}\cdots a_1a_0\mathbf{0}^{\infty},
\end{aligned}
$$

recall that $\lambda(q,m,\nu) = (\nu - 1 - (m + \mathsf{length}(\mathsf{binary}(q))))$ is as in (5.4), i.e., the number of initial zeroes of $\mathsf{dyad}\left(\frac{2^m q}{2^\nu}\right)$. Therefore, a good strategy is to compute the matrix product for $m_{\max}$, the m such that $\lambda(q, m_{\max}, \nu) = 0$. This is done by simply setting

$$m_{\max} = (\nu - 1) - \mathsf{length}(\mathsf{binary}(q)) \tag{5.10}$$

This stems from the fact that for any $n \in \mathbb{Z}^+$ we have

$$2^{\mathsf{length}(\mathsf{binary}(n))-1} \leq n < 2^{\mathsf{length}(\mathsf{binary}(n))}, \text{ and also}$$
$$\mathsf{length}\big(\mathsf{binary}\big(2^m n\big)\big) = m + \mathsf{length}(\mathsf{binary}(n)).$$

Hence, the equality $m_{\max} + \mathsf{length}(\mathsf{binary}(q)) = \nu - 1$ is equivalent to $m_{\max}$ being the largest m for which $2^m q < 2^{\nu-1}$ (thus $\frac{2^m q}{2^\nu} < \frac{1}{2}$). Therefore, in the case when $m = m_{\max}$ the matrix product is simply

$$\mathbf{M}_1 \cdot \mathbf{M}_{a_{l-2}} \cdot \mathbf{M}_{a_{l-3}} \cdots \mathbf{M}_{a_1} \cdot \mathbf{M}_{a_0} \overrightarrow{\mathbf{V}}^{(0)}.$$

with $\mathsf{binary}(q) = 1a_{l-2}a_{l-3}\cdots a_1 a_0$. Recall that the contribution of $\mathbf{M}_0^\infty$ has been simplified by the use of the vector $\overrightarrow{\mathbf{V}}^{(0)}$ in Equation (5.6).

Once the matrix product for $m_{\max}$ has been computed for $0 \leq m < m_{\max}$ we have that our matrix product will be

$$M_0^{m_{\max}-m} \underbrace{\mathbf{M}_1 \cdot \mathbf{M}_{a_{l-2}} \cdot \mathbf{M}_{a_{l-3}} \cdots \mathbf{M}_{a_1} \cdot \mathbf{M}_{a_0} \overrightarrow{\mathbf{V}}^{(0)}}_{\text{Matrix products for } m_{\max}}. \tag{5.11}$$

Thus the strategy is to compute the matrix products for $m_{\max}$ and then obtain all the others by iteration on m by simply multiplying the previous product by $\mathbf{M}_0$ on the left.

Then, by the dichotomy between matrix products and the vector u we simply need to multiply the resulting matrix product by the suitable $u(\ell)$ or $u\big(\ell + \frac{1}{2}\big)$. Finally, to obtain the wavelet value on any $\frac{2^m q}{2^\nu} + \ell$ or $\frac{2^m q}{2^\nu} + \ell + \frac{1}{2}$ we simply multiply the result by $\pm\frac{\sqrt{2}}{2p-1}$.

Algorithm 5.13 (Computation of $\psi(y)$ for all $y \in \operatorname{supp}(\psi)$, y dyadic of order ν.). This algorithm corresponds to the efficient implementation of Corollary 5.6.

Initialize: pre-compute the vectors $u(\ell)$ and $u\big(\frac{1}{2} + \ell\big)$ for $\ell = 1-p, \ldots, p-1$ using (5.9).

Initialize: pre-compute 2^ν and the coefficient $\frac{\sqrt{2}}{2p-1}$

Step 0: Compute $\psi(\ell)$ and $\psi\big(\frac{1}{2}+\ell\big)$ for $\ell = 1-p, \ldots, p-1$ by using Corollary 5.11.

Initialize: $q = 1$

Step 1: Compute the Daubechies–Lagarias Products Vector

$$\overrightarrow{\mathsf{DL}}(q) := \mathbf{M}_1 \cdot \mathbf{M}_{a_{l-2}} \cdot \mathbf{M}_{a_{l-3}} \cdots \mathbf{M}_{a_1} \cdot \mathbf{M}_{a_0} \cdot \overrightarrow{\mathbf{V}}^{(0)} \tag{5.12}$$

with $\mathsf{binary}(q) = 1a_{l-2}a_{l-3}\cdots a_1 a_0$. Moreover, by Equation (5.10) set

$$m_{\max} = (\nu - 1) - \mathsf{length}(\mathsf{binary}(q))$$

Step 2: Compute $\psi(x+\ell)$ and $\psi\bigl(x+\frac{1}{2}+\ell\bigr)$ with $x = \frac{2^{m_{\max}}q}{N}$ for the integers $\ell = 1-p,\dots,p-1$, using Corollary 5.11 by

$$\begin{aligned}\psi(x+\ell) &= \tfrac{\sqrt{2}}{2p-1}u(l)\cdot\left(\lim_{n\to\infty}\prod_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}(2x,i)}\right)\vec{\mathbf{1}}\\ &= \tfrac{\sqrt{2}}{2p-1}u(\ell)\cdot\vec{\mathsf{DL}}(q) \qquad \text{and,}\\ \psi\left(x+\ell+\frac{1}{2}\right) &= -\tfrac{\sqrt{2}}{2p-1}u\left(\ell+\frac{1}{2}\right)\cdot\left(\lim_{n\to\infty}\prod_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}(2x,i)}\right)\vec{\mathbf{1}}\\ &= -\tfrac{\sqrt{2}}{2p-1}u\left(\ell+\frac{1}{2}\right)\cdot\vec{\mathsf{DL}}(q).\end{aligned}$$

The last equality holds due to Equations (5.6) and (5.10).

Step 3: For every $m = m_{\max}-1, m_{\max}-2,\dots,0$ (if any) set

$$\vec{\mathsf{DL}}(q) = \mathbf{M}_0\vec{\mathsf{DL}}(q).$$

Then compute $\psi(x+\ell)$ and $\psi\bigl(x+\frac{1}{2}+\ell\bigr)$ with $x = \frac{2^{m}q}{N}$ for the integers $\ell = 1-p,\dots,p-1$, using Corollary 5.11 with

$$\begin{aligned}\psi(x+\ell) &= \tfrac{\sqrt{2}}{2p-1}u(l)\cdot\left(\lim_{n\to\infty}\prod_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}(2x,i)}\right)\vec{\mathbf{V}}^{(0)}\\ &= \tfrac{\sqrt{2}}{2p-1}u(\ell)\cdot\vec{\mathsf{DL}}(q) \qquad \text{and,}\\ \psi\left(x+\ell+\frac{1}{2}\right) &= -\tfrac{\sqrt{2}}{2p-1}u\left(\ell+\frac{1}{2}\right)\cdot\left(\lim_{n\to\infty}\prod_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}(2x,i)}\right)\vec{\mathbf{V}}^{(0)}\\ &= -\tfrac{\sqrt{2}}{2p-1}u\left(\ell+\frac{1}{2}\right)\vec{\mathsf{DL}}(q).\end{aligned}$$

This follows because

$$\lambda(q,m-1,\nu) = \lambda(q,m,\nu)+1.$$

Iterate: $q \leftarrow q+2$; `if` $q < 2^{\nu-1}$ `goto` **Step 1**

∎

An efficiency comment to Step 1: The *Compact Binary Expansion* of an odd positive integer q is a triplet $\bigl(\mathsf{lb},\kappa,(b_1,b_2,\dots,b_\kappa)\bigr)$ where $\kappa \geq 1$ is odd, $\mathsf{lb} = \sum_{i=1}^{\kappa} b_i$ and the numbers b_i are positive integers such that

$$\mathsf{binary}(q) = \mathbf{1}^{b_\kappa}\mathbf{0}^{b_{\kappa-1}}\mathbf{1}^{b_{\kappa-2}}\cdots\mathbf{0}^{b_2}\mathbf{1}^{b_1},$$

where $\mathbf{0}^{b_i}$ (respectively $\mathbf{1}^{b_i}$) is a block of b_i consecutive 0's (respectively 1's). Observe that the Compact Binary Expansion of an odd number is always well defined and $\mathsf{lb} = \mathsf{length}\bigl(\mathsf{binary}(q)\bigr)$.

By using Compact Binary Expansions we can write:

$$\begin{aligned}\vec{\mathsf{DL}}(q) := \mathbf{M}_1\cdot\mathbf{M}_{a_{\kappa-2}}\cdot\mathbf{M}_{a_{\kappa-3}}\cdots\mathbf{M}_{a_1}\cdot\mathbf{M}_{a_0}\cdot\vec{\mathbf{V}}^{(0)} &=\\ \mathbf{M}_1^{b_\kappa}\cdot\mathbf{M}_0^{b_{\kappa-1}}\cdot\mathbf{M}_1^{b_{\kappa-2}}\cdots\mathbf{M}_0^{b_2}\cdot\mathbf{M}_1^{b_1}\cdot\vec{\mathbf{V}}^{(0)}\end{aligned}$$

Moreover, the computation of $\lambda(q,m,\nu)$ becomes almost trivial, since

$$\lambda(q,m,\nu) = (\nu-1)-(m+\mathsf{lb}).$$

Now, $m_{\max} = (\nu-1)-\mathsf{lb}$ is trivial to compute. Moreover, we can greatly improve computation times by using pre-computed powers of the matrices $\mathbf{M}_0$ and $\mathbf{M}_1$. We believe that it is worth to store $\mathbf{M}_0^n$ and $\mathbf{M}_1^n$ for $n = 1,2,\dots,\nu-2$. We are well aware that this leads to an increase in the usage of memory. However, considering that $\mathbf{M}_0$ and $\mathbf{M}_1$ have dimension $(2p-1)\times(2p-1)$ and that our final goal is to compute the two $2^\nu \times 2^\nu$ wavelet matrices, the a remarkable increase in speed that follows from the storage of the powers $\mathbf{M}_0^n$ and $\mathbf{M}_1^n$ for $n = 1, 2, \dots, \nu - 2$ trumps any slight (in comparison) increase in memory requirements.

5.4. Evaluating Periodized Wavelets on Dyadic Points

Until now we have been dealing with computing the Daubechies mother wavelet ψ on dyadic points. However, in Chapter 3 we mostly work with periodized wavelets, $\psi_{j,n}^{\text{PER}}$, see Section 3.3. Nevertheless, the results from the previous section will be fundamental for our purposes. We will study two cases, in Subsection 5.4.1 we focus on the periodized wavelet $\psi^{\text{PER}} = \psi_{0,0}^{\text{PER}}$, in which the dichotomy between the vector u and the matrix products is not as obvious. In fact the vector u will be trivial. In Subsection 5.4.2 we are interested in computing $\psi_{-j,n}$, with $j \geq 1$ and $n \in \{0, \dots 2^j - 1\}$, where the dichotomy will come back in full force.

5.4.1. The Periodized Mother Wavelet ψ^{PER}. In this subsection we will give an algorithm to compute

$$\psi^{\text{PER}}(\theta) = \psi_{0,0}^{\text{PER}}(\theta) = \psi_0^{\text{PER}}(\theta) \quad \text{for every} \quad \theta \in \mathbb{S}^1 = [0, 1).$$

To this end, we will apply the Daubechies-Lagarias algorithm and give an adaptation of Algorithm 5.13 for the particular case of the periodized wavelet.

Corollary 5.14. *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments. Then, for every $\theta \in \mathbb{S}^1$,*

$$\begin{aligned}\psi^{\text{PER}}(\theta) &= \sum_{\ell \in \mathbb{Z}} \psi(\theta + \ell) = \sum_{\ell = 1-p}^{p-1} \psi(\theta + \ell) = \\ &= -(-1)^{d(\theta)} \tfrac{\sqrt{2}}{2p-1} \left(\sum_{\ell=1-p}^{p} u(\theta + \ell) \right)^{\mathsf{T}} \mathbf{DL}(\{\!\{2\theta\}\!\}) \vec{\mathbf{1}}.\end{aligned}$$

Proof. The first equality is the definition of periodized wavelets as in Equation (3.18). The second equality follows from Lemma 3.24. The last equality is a direct consequence of Corollary 5.11 and the following two facts. On one hand, we have that $\{\!\{2(\theta+\ell)\}\!\} = \{\!\{2\theta\}\!\}$. On the other, by the definition of $d(x)$ we have that

$$d(\theta + \ell) = 1 - \lfloor 2\theta + 2\ell \rfloor = d(\theta) - 2\ell, \tag{5.13}$$

hence $(-1)^{d(\theta+\ell)} = (-1)^{d(\theta)}$. □

Now, let us introduce a technical proposition that will allow us to simplify the computations down the line.

Proposition 5.15. *Let ψ be an $\mathbb{R}$-Daubechies wavelets with more than $p \geq 1$ vanishing moments. Then the following hold:*

(a) For every $\theta \in \left[0, \frac{1}{2}\right)$,

$$\sum_{\ell \in \mathbb{Z}} u(\theta + \ell) = \frac{\sqrt{2}}{2} \, \vec{\boldsymbol{ua}}_{2p-1} = \sum_{\ell \in \mathbb{Z}} u\big(\theta + \tfrac{1}{2} + \ell\big),$$

where $\vec{\boldsymbol{ua}}_{2p-1}^{\mathsf{T}}$ denotes the row vector $(1, -1, 1, -1, \dots, -1, 1)$ of dimension $2p{-}1$.

(b) $\psi^{\text{PER}}(\theta) = -\psi^{\text{PER}}\left(\theta + \frac{1}{2}\right)$ for every $\theta \in \left[0, \frac{1}{2}\right)$

Proof. To prove (a), let us denote the i-th component of $u(\theta)$ as $u_i(\theta) = (-1)^i h[i + d(x)]$, $i \in \{0, 1, \ldots, 2p-2\}$. Then, by Remark 3.8 (b)

$$\begin{aligned}
\sum_{\ell\in\mathbb{Z}} u_i\big(\theta+\ell\big) &= \sum_{\ell\in\mathbb{Z}} (-1)^i\, h\big[i + d(\theta+\ell)\big] \quad \text{Equation(5.13)} \\
&= (-1)^i \sum_{\ell\in\mathbb{Z}} h\big[i + d(\theta) - 2\ell\big] \quad \text{Equation(3.11)} \\
&= (-1)^i \frac{\sqrt{2}}{2} \\
&= (-1)^i \sum_{\ell\in\mathbb{Z}} h\big[i + d(\theta) - 1 - 2\ell\big] \quad \text{Equation(3.11)} \\
&= \sum_{\ell\in\mathbb{Z}} (-1)^i\, h\big[i + d\big(\theta + \tfrac{1}{2} + \ell\big)\big] \quad \text{since } d\left(\theta+\tfrac{1}{2}\right) = d(\theta) - 1 \\
&= \sum_{\ell\in\mathbb{Z}} u_i\big(\theta + \tfrac{1}{2} + \ell\big).
\end{aligned}$$

Finally, (b) follows directly from (a), Corollary 5.14, and the fact that $d\left(\theta+\frac{1}{2}\right) = d(\theta) - 1$. □

Note that form Proposition 5.15(b) it follows that to compute $\Psi^{\text{PER}}(\theta)$ for every $\theta \in \mathbb{S}^1$ we only need to compute the values of $\psi^{\text{PER}}(\theta)$ for $\theta \in \left[0, \frac{1}{2}\right)$.

Notice that $d(\theta) = 1 - \lfloor 2\theta \rfloor = 1$ whenever $\theta \in \left[0, \frac{1}{2}\right)$. Hence, from Corollary 5.14 and Proposition 5.15 we get

Corollary 5.16. *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments. Then, for every $\theta \in \left[0, \frac{1}{2}\right)$,*

$$\psi^{\text{PER}}(\theta) = \tfrac{1}{2p-1}\ \overrightarrow{\boldsymbol{ua}}^{\mathsf{T}}_{2p-1}\mathbf{DL}\big(\{\!\{2\theta\}\!\}\big)\,\vec{\mathbf{1}}.$$

Moreover, we can re-write the corollary by considering θ to be a dyadic rational of the form $\frac{2^m q}{2^\nu}$, that is, we only need to compute $\mathsf{binary}(q)$ when it comes to $\mathsf{dyad}(\theta)$. Hence, from Equations (5.6) and (5.7), we get

Corollary 5.16 (Dyadic version). *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments. Then,*

$$\psi^{\text{PER}}(0) = \tfrac{1}{2p-1}\ \overrightarrow{\boldsymbol{ua}}^{\mathsf{T}}_{2p-1}\vec{\mathbf{V}}^{(0)},$$

and for every $\nu \in \mathbb{N} \setminus \{1\}$, $m \in \{0, 1, \ldots, \nu-2\}$ and $q < 2^{(\nu-1)-m}$ odd,

$$\psi^{\text{PER}}\left(\frac{2^m q}{2^\nu}\right) = \tfrac{1}{2p-1}\ \overrightarrow{\boldsymbol{ua}}^{\mathsf{T}}_{2p-1}\Big(\mathbf{M}_0^{\lambda(q,m,\nu)} \cdot \mathbf{M}_1 \cdot \mathbf{M}_{a_{\ell-2}} \cdot \mathbf{M}_{a_{\ell-3}} \cdots \mathbf{M}_{a_1} \cdot \mathbf{M}_{a_0} \cdot \vec{\mathbf{V}}^{(0)}\Big),$$

where $\mathsf{binary}(q) = 1a_{\ell-2}a_{\ell-3}\cdots a_1a_0$ *and* $\lambda(q,m,\nu) = (\nu-1) - \big(m + \mathsf{length}(\mathsf{binary}(q))\big)$.

Now we have all the information to be able to provide an adaptation of Algorithm 5.13 to the periodized wavelet. That is, on the computation of the dyadic expansion of odd integers q and then computing $m_{\max}$ for each q. The computation of all the other $0 \leq m < m_{\max}$ then follows. Note that most of the notation and ideas behind the algorithm (such as $m_{\max}$ and using odd qs) correspond to the ones in Algorithm 5.13, therefore they will not be repeated here.

Algorithm 5.17 (Computation of $\psi^{\text{PER}}(\theta)$ for dyadic $\theta \in \mathbb{S}^1$). This algorithm is the implementation of Corollary 5.14 following the structure and techniques of Algorithm 5.13.

```
Step 0: Compute $\psi^{\text{PER}}(0) = \tfrac{1}{2p-1}\ \overrightarrow{\mathbf{ua}}^{\mathsf{T}}_{2p-1}\vec{\mathbf{V}}^{(0)}$.
Initialize: set $q = 1$
```

Step 1: Compute the Daubechies–Lagarias Products Vector

$$\overrightarrow{\mathsf{DL}}(q) := \mathbf{M}_1 \cdot \mathbf{M}_{a_{\ell-2}} \cdot \mathbf{M}_{a_{\ell-3}} \cdots \mathbf{M}_{a_1} \cdot \mathbf{M}_{a_0} \cdot \vec{\mathbf{V}}^{(0)}$$

with $\mathsf{binary}(q) = 1a_{\ell-2}a_{\ell-3}\cdots a_1a_0$. Compute also

$$m_{\max} := (\nu - 1) - \mathsf{length}(\mathsf{binary}(q)).$$

Step 2: Compute

$$\psi^{\text{PER}}\left(\frac{2^{m_{\max}}q}{2^\nu}\right) = \tfrac{1}{2p-1}\ \overrightarrow{\mathsf{ua}}^{\mathsf{T}}_{2p-1}\overrightarrow{\mathsf{DL}}(q).$$

Step 3: For every integer $m = m_{\max} - 1, m_{\max} - 2, \ldots, 0$ (if there exists any) re-define $\overrightarrow{\mathsf{DL}}(q) = \mathbf{M}_0\overrightarrow{\mathsf{DL}}(q)$ and compute

$$\psi^{\text{PER}}\left(\frac{2^{m}q}{2^\nu}\right) = \tfrac{1}{2p-1}\ \overrightarrow{\mathsf{ua}}^{\mathsf{T}}_{2p-1}\overrightarrow{\mathsf{DL}}(q).$$

Iterate: $q \leftarrow q + 2$; `if` $q < 2^{\nu-1}$ `goto` **Step 1**

■

5.4.2. The Periodized Wavelets $\psi^{\text{PER}}_{-j,n}$ for $j \geq 1$. When it comes to evaluating $\psi^{\text{PER}}_{-j,n}$ on dyadic points for $j \geq 1$ the dichotomy between $u(\theta)$ and the matrix products $\mathbf{DL}\big(\{\!\{2\theta\}\!\}\big)$ comes back in full force. However, in this case instead of dividing the support in sub-intervals of size $\frac{1}{2}$, $\mathbb{S}^1 \equiv [0,1)$ is divided into subintervals of length $\frac{1}{2^j}$. Even though the vector u now will depend on j and n, we will be able to give general formulae for their computation. First, let us start with a corollary of Proposition 3.22 to remind us of the self-similarities of the periodized wavelets.

Corollary 5.18 (of Proposition 3.22). *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments. Then, for every $\theta \in \big[0, \frac{1}{2^j}\big)$, $j \in \{1, \ldots, J\}$, and $n \in \{0, 1, \ldots, 2^j - 1\}$,*

$$\psi^{\text{PER}}_{-j,n}(\theta) = \psi^{\text{PER}}_{-j,\mathsf{mod}\left(n+k,\,2^j\right)}\left(\theta + \tfrac{k}{2^j}\right)$$

for every $k = 1, 2, \ldots, 2^j - 1$.

Remark 5.19. Implementation-wise this is a crucial lemma. As shown in Remark 4.10, it allows us to slash the computation times and storage required for the matrices, since for any j and ν fixed, the columns of the block of size $2^\nu \times 2^j$ starting at the column 2^j are merely permutations of the same column. Hence, both when computing these blocks and when it comes to storing them, we only need to handle any one of the 2^j columns involved. Therefore, the efficiency per block both in terms of memory usage and computation-wise is increased by a factor of 2^j, which for large values of j is remarkable. ■

Now we need the analogue of Corollary 5.14 to deal with the effective computation of $\psi^{\text{PER}}_{-j,n}$ at the dyadic points.

In what follows we set

$$\overrightarrow{\mathsf{uj}}(a) := \sum_{\ell\in\mathbb{Z}} u\big(2^j\ell - a\big) \tag{5.14}$$

Corollary 5.20. *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments and let $\nu \in \mathbb{N} \setminus \{1\}$, $j \in \{1, \ldots, \nu - 1\}$ and $n \in \{0, 1, \ldots, 2^j - 1\}$. Then* *for every* $\theta \in \big(0, \frac{1}{2^{j+1}}\big)$,

$$\psi^{\text{PER}}_{-j,n}(0) = \frac{\sqrt{2}^{j+1}}{2p-1}\ \vec{\boldsymbol{uj}}(n)^{\mathsf{T}}\vec{\mathbf{V}}^{(0)},$$

$$\psi^{\text{PER}}_{-j,n}(\theta) = \frac{\sqrt{2}^{j+1}}{2p-1}\ \vec{\boldsymbol{uj}}(n)^{\mathsf{T}}\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}(2^{j+1}\theta,i)}\right)\vec{\mathbf{1}},$$

$$\psi^{\text{PER}}_{-j,n}\big(\tfrac{1}{2^{j+1}}\big) = -\frac{\sqrt{2}^{j+1}}{2p-1}\ \vec{\boldsymbol{uj}}\big(n-\tfrac{1}{2}\big)^{\mathsf{T}}\vec{\mathbf{V}}^{(0)},\ \textit{and}$$

$$\psi^{\text{PER}}_{-j,n}\big(\theta+\tfrac{1}{2^{j+1}}\big) = -\frac{\sqrt{2}^{j+1}}{2p-1}\ \vec{\boldsymbol{uj}}\big(n-\tfrac{1}{2}\big)^{\mathsf{T}}\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}(2^{j+1}\theta,i)}\right)\vec{\mathbf{1}},$$

Now, we can write what we call the dyadic version of the previous corollary. Recall that in this case we are interested in the case when θ is a dyadic rational and hence we can explicitly compute its dyadic expansion in terms of the binary expansion of an integer.

Corollary 5.20 (Dyadic version). *Let* ψ *be an* $\mathbb{R}$*-Daubechies wavelet with* $p \geq 1$ *vanishing moments, let* $\nu \in \mathbb{N}\setminus\{1\}$, $j \in \{1,\dots,\nu-1\}$ *and* $n \in \{0,1,\dots,2^j-1\}$, *and let* $\frac{2^m q}{2^\nu}$ *be a dyadic rational in the interval* $\big(0,\frac{1}{2^{j+1}}\big)$, *where* $m \in \mathbb{Z}^+$ *and* $q \in \mathbb{N}$ *is odd. Then,*

$$\psi^{\text{PER}}_{-j,n}\left(\frac{2^m q}{2^\nu}\right) = \frac{\sqrt{2}^{j+1}}{2p-1}\ \vec{\boldsymbol{uj}}(n)^{\mathsf{T}} \left(\mathbf{M}_0^{\lambda(q,m+j,\nu)}\cdot\mathbf{M}_1^{b_\kappa}\cdot\mathbf{M}_0^{b_{\kappa-1}}\cdots\mathbf{M}_0^{b_2}\cdot\mathbf{M}_1^{b_1}\cdot\vec{\mathbf{V}}^{(0)}\right),\ \textit{and}$$

$$\psi^{\text{PER}}_{-j,n}\left(\frac{2^m q}{2^\nu}+\tfrac{1}{2^{j+1}}\right) = -\frac{\sqrt{2}^{j+1}}{2p-1}\ \vec{\boldsymbol{uj}}\big(n-\tfrac{1}{2}\big)^{\mathsf{T}} \left(\mathbf{M}_0^{\lambda(q,m+j,\nu)}\cdot\mathbf{M}_1^{b_\kappa}\cdot\mathbf{M}_0^{b_{\kappa-1}}\cdots\mathbf{M}_0^{b_2}\cdot\mathbf{M}_1^{b_1}\cdot\vec{\mathbf{V}}^{(0)}\right),$$

where $\big(\mathsf{lb},\kappa,(b_1,b_2,\dots,b_\kappa)\big)$ *is the Compact Binary Expansion of* q *(more concretely* $\mathsf{binary}(q) = \mathbf{1}^{b_\kappa}\mathbf{0}^{b_{\kappa-1}}\cdots\mathbf{0}^{b_2}\mathbf{1}^{b_1}$*), and* $\lambda(q,m+j,\nu) = (\nu-1)-(m+j+\mathsf{lb})$.

To obtain useful (implementable) and efficient formulae for $\psi^{\text{PER}}_{-j,n}(\theta)$, we additionally need explicit expressions for the vectors $\vec{\mathbf{uj}}(n)$ and $\vec{\mathbf{uj}}\big(n-\frac{1}{2}\big)$. This will be done in the next subsection, after the proof of Corollary 5.20.

Proof of Corollary 5.20. Recall that whenever $k \in \mathbb{Z}$

$$\{\!\{2(\theta+k)\}\!\} = \{\!\{2\theta\}\!\} \quad\text{and}\quad (-1)^{d(\theta+k)} = (-1)^{d(\theta)-2k} = (-1)^{d(\theta)}.$$

So, by Equation (3.18) and Corollary 5.6,

$$\begin{aligned}\psi^{\text{PER}}_{-j,n}(\theta) &= \sum_{\ell\in\mathbb{Z}}\psi_{-j,n}(\theta+\ell) = 2^{j/2}\sum_{\ell\in\mathbb{Z}}\psi\left(\frac{(\theta+\ell)-2^{-j}n}{2^{-j}}\right)\\ &= 2^{j/2}\sum_{\ell\in\mathbb{Z}}\psi\big(2^j\theta+2^j\ell-n\big)\\ &= -(-1)^{d(2^j\theta)}\frac{\sqrt{2}^{j+1}}{2p-1}\left(\sum_{\ell\in\mathbb{Z}}u\big(2^j\theta+2^j\ell-n\big)\right)^{\mathsf{T}}\\ &\qquad\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}(\{\!\{2^{j+1}\theta\}\!\},i)}\right)\vec{\mathbf{1}}.\end{aligned}$$

for every $\theta \in [0, 1)$, $j \in \{1, \dots, \nu - 1\}$, and $n \in \{0, 1, \dots, 2^j - 1\}$. As before, to minimize computations, we have to improve the expressions of

$$d(2^j\theta),\ \sum_{\ell \in \mathbb{Z}} u\big(2^j\theta + 2^j\ell - n\big) \text{ and } \{\!\!\{2^{j+1}\theta\}\!\!\}.$$

For every $\theta \in \Big[0, \frac{1}{2^{j+1}}\Big)$ we have

$$\begin{aligned}
\bigg\{\!\!\!\bigg\{ 2^{j+1}\Big(\theta + \tfrac{1}{2^{j+1}}\Big)\bigg\}\!\!\!\bigg\} &= \{\!\!\{2^{j+1}\theta\}\!\!\} = 2^{j+1}\theta,\\
d\big(2^j\theta + 2^j\ell - n\big) &= 1 - \big\lfloor 2\big(2^j\theta + 2^j\ell - n\big)\big\rfloor = 1 - \big\lfloor 2^{j+1}\theta\big\rfloor - 2\big(2^j\ell - n\big)\\
&= 1 - 2\big(2^j\ell - n\big) = d\big(2^j\ell - n\big),\\
d\big(2^j\theta\big) &= d(0) = 1,\\
d\Big(2^j\big(\theta + \tfrac{1}{2^{j+1}}\big) + 2^j\ell - n\Big) &= 1 - \Big\lfloor 2\Big(2^j\big(\theta + \tfrac{1}{2^{j+1}}\big) + 2^j\ell - n\Big)\Big\rfloor\\
&= 1 - \big\lfloor 2^{j+1}\theta + 1 + 2\big(2^j\ell - n\big)\big\rfloor\\
&= -\big\lfloor 2^{j+1}\theta\big\rfloor - 2\big(2^j\ell - n\big) = -2\big(2^j\ell - n\big)\\
&= 1 - \big\lfloor 2\big(\tfrac{1}{2} + 2^j\ell - n\big)\big\rfloor = d\big(\tfrac{1}{2} + 2^j\ell - n\big), \text{ and}\\
d\Big(2^j\big(\theta + \tfrac{1}{2^{j+1}}\big)\Big) &= d\big(\tfrac{1}{2}\big) = 0.
\end{aligned}$$

Consequently, if $u_i(\theta) = (-1)^i h[i + d(\theta)]$ denotes the i-th component of the vector $u(\theta)$ with $i \in \{0, 1, \dots, 2p - 2\}$, we get

$$\begin{aligned}
u_i\big(2^j\theta + 2^j\ell - n\big) &= (-1)^i\ h\big[i + d\big(2^j\theta + 2^j\ell - n\big)\big]\\
&= u_i\big(2^j\ell - n\big), \text{ and}\\
u_i\Big(2^j\big(\theta + \tfrac{1}{2^{j+1}}\big) + 2^j\ell - n\Big) &= (-1)^i\ h\Big[i + d\Big(2^j\big(\theta + \tfrac{1}{2^{j+1}}\big) + 2^j\ell - n\Big)\Big]\\
&= u_i\big(\tfrac{1}{2} + 2^j\ell - n\big).
\end{aligned}$$

In summary,

$$\begin{aligned}
u\big(2^j\theta + 2^j\ell - n\big) &= u\big(2^j\ell - n\big), \text{ and}\\
u\Big(2^j\big(\theta + \tfrac{1}{2^{j+1}}\big) + 2^j\ell - n\Big) &= u\big(\tfrac{1}{2} + 2^j\ell - n\big).
\end{aligned}$$

Concerning the dyadic statements, note that the condition

$$\frac{2^m q}{2^\nu} < \frac{1}{2^{j+1}} = \frac{2^{\nu-j-1}}{2^\nu}$$

is equivalent to $2^m q < 2^{\nu-j-1}$, which implies $q < 2^{\nu-j-m-1}$ and hence, $m < \nu - j - 1$ because $q \geq 1$. Consequently, $\mathsf{lb} = \mathsf{length}(\mathsf{binary}(q)) \leq \nu - j - m - 1$.

On the other hand,

$$\mathsf{dyad}\left(2^{j+1}\frac{2^m q}{2^\nu}\right) = \mathsf{dyad}\left(\frac{q}{2^{\nu-m-j-1}}\right) = \mathbf{0}^{\lambda(q,m+j,\nu)}\mathbf{1}^{b_\kappa}\mathbf{0}^{b_{\kappa-1}}\cdots\mathbf{0}^{b_2}\mathbf{1}^{b_1}\mathbf{0}^{\infty}.$$

Now the corollary follows putting all the above together and by using Equations (5.6) and (5.7). □

5.4.2.1. *On the efficient computation of the vectors* $\overrightarrow{\boldsymbol{uj}}(n)$ *and* $\overrightarrow{\boldsymbol{uj}}\big(n - \frac{1}{2}\big)$. The only remaining issue regarding an implementation of Corollary 5.20 is to give explicit formulae for the vectors $\overrightarrow{\mathbf{uj}}(n)$ and $\overrightarrow{\mathbf{uj}}\big(n - \frac{1}{2}\big)$, desirably depending only on j and the number of vanishing moments p. Recall that $\overrightarrow{\mathbf{uj}}(a) = \sum_{\ell \in \mathbb{Z}} u(2^j\ell - a)$. To this end, let us study some properties of $u(2^j\ell - a)$ whenever $a \in \mathbb{Z} + \frac{1}{2}\mathbb{Z}$. In particular, since $[u(k)]_i = (-1)^i h[i + 1 - 2k]$ for $k \in \mathbb{Z} + \frac{1}{2}\mathbb{Z}$ it follows that $[u(k)]_i = 0$

whenever $i+1-2k \notin \operatorname{supp}(h) = [0, 2p-1]$. Hence, it follows that $u(k) = \vec{\mathbf{0}}$ if $\{i+1-2k\}_{i=0}^{2p-2} \notin \operatorname{supp}(h)$. Hence by defining

$$\operatorname{base}\big(u(k)\big) := \{1-2k, 2-2k, \dots, 2p-1-2k\} \tag{5.15}$$

we have that $u(k) = \vec{\mathbf{0}}$ if and only if $\operatorname{base}(u(k)) \cap \operatorname{supp}(h) = \emptyset$.

As we want to compute $\vec{\mathbf{uj}}(n)$ and $\vec{\mathbf{uj}}(n-\frac{1}{2})$, we are interested in the conditions on ℓ as a function of n and δ such that

$$\operatorname{base}\left(u\left(2^j\ell - n + \frac{\delta}{2}\right)\right) \cap \operatorname{supp}(h) \neq \emptyset,$$

where $\delta \in \{0, 1\}$.

Lemma 5.21. *Let $\ell \in \mathbb{Z}$ then*

$$\operatorname{base}\left(u\left(2^j\ell - n + \frac{\delta}{2}\right)\right) \cap \operatorname{supp}(h) \neq \emptyset,$$

if and only if ℓ is such that

$$\ell_{\min} := \left\lceil \frac{1-p+n-\frac{\delta}{2}}{2^j} \right\rceil \leq \ell \leq \left\lfloor \frac{p+n-\frac{\delta+1}{2}}{2^j} \right\rfloor =: \ell_{\max} \tag{5.16}$$

Proof. Not that if we explicitaly compute the base we get

$$\operatorname{base}\left(u\left(2^j\ell - n + \frac{\delta}{2}\right)\right) = \Big\{1 - 2^{j+1}\ell + 2n - \delta, \dots, 2p-1-2^{j+1}\ell + 2n - \delta\Big\}.$$

Since $\operatorname{supp}(h) = \{0, \dots, 2p-1\}$ it is clear that $\operatorname{base}\big(u\left(2^j\ell - n + \frac{\delta}{2}\right)\big) \cap \operatorname{supp}(h) \neq \emptyset$ if and only if $1 - 2^{j+1}\ell + 2n - \delta \leq 2p-1$ and $2p-1-2^{j+1}\ell + 2n - \delta \geq 0$. Since $\ell \in \mathbb{Z}$, it follows that the valid values of ℓ are

$$\left\lceil \frac{1-p+n-\frac{\delta}{2}}{2^j} \right\rceil \leq \ell \leq \left\lfloor \frac{p+n-\frac{\delta+1}{2}}{2^j} \right\rfloor.$$

Hence the result follows. □

Remark 5.22. These $\ell_{\max}$ and $\ell_{\min}$ allow us to compute the vectors $\vec{\mathbf{uj}}(n)$ and $\vec{\mathbf{uj}}\left(n-\frac{1}{2}\right)$ in a realistic way, since they limit the ℓ required to compute them. That is,

$$\vec{\mathbf{uj}}(n) = \sum_{\ell\in\mathbb{Z}} u(2^j\ell - n) = \sum_{\ell_{\min}}^{\ell_{\max}} u(2^j\ell - n) \quad \text{and}$$

$$\vec{\mathbf{uj}}\left(n-\frac{1}{2}\right) = \sum_{\ell\in\mathbb{Z}} u\left(2^j\ell - n + \frac{1}{2}\right) = \sum_{\ell_{\min}}^{\ell_{\max}} u\left(2^j\ell - n + \frac{1}{2}\right)$$

■

Finally, we are able to compute the vectors $\vec{\mathbf{uj}}(n-\frac{\delta}{2})$ as a function of only h, j and n.

Proposition 5.23. *Let*

$$\ell_{\min} = \left\lceil \frac{1-p+n-\frac{\delta}{2}}{2^j} \right\rceil \qquad \ell_{\max} = \left\lfloor \frac{p+n-\frac{\delta+1}{2}}{2^j} \right\rfloor$$

be as in Lemma 5.21. Then, following statements hold:

(a) For every $j \in \mathbb{N}$, $n \in \{0, 1, \dots, 2^j - 1\}$ *and* $i \in \{0, 1, 2, \dots, 2p - 2\}$,

$$\big[\overrightarrow{\boldsymbol{uj}}(n)\big]_i = (-1)^i \sum_{\substack{\ell=\ell_{\min} \\ i+1+2n-2^{j+1}\ell \in \operatorname{supp}(h)}}^{\ell_{\max}} h\big[i + 1 + 2n - 2^{j+1}\ell\big].$$

(b) For every $j \in \mathbb{N}$ *and* $n \in \{0, 1, \dots, 2^j - 1\}$,

$$\big[\overrightarrow{\boldsymbol{uj}}(n - \tfrac{1}{2})\big]_0 = \sum_{\ell=\left\lceil \frac{2n-2p+1}{2^{j+1}} \right\rceil}^{\left\lfloor \frac{n}{2^j} \right\rfloor} h\big[2n - 2^{j+1}\ell\big].$$

(c) For every $j \in \mathbb{N}$, $n \in \{0, 1, \dots, 2^j - 1\}$ *and* $i \in \{1, 2, \dots, 2p - 2\}$,

$$\big[\overrightarrow{\boldsymbol{uj}}(n - \tfrac{1}{2})\big]_i = -\big[\overrightarrow{\boldsymbol{uj}}(n)\big]_{i-1}.$$

Proof. (a) follows directly from Lemma 5.21. (b) follows directly from the definition of $\overrightarrow{\mathbf{uj}}$ applying it to the $i = 0$ position. As for Statement (c), since $d(x) = 1 - \lfloor 2x \rfloor$, for every $m \in \mathbb{Z}$ we have

$$d\big(m + \tfrac{1}{2}\big) = 1 - \big\lfloor 2\big(m + \tfrac{1}{2}\big)\big\rfloor = 1 - \lfloor 2m \rfloor - 1 = d(m) - 1.$$

Hence,

$$\begin{aligned}\big[\overrightarrow{\mathbf{uj}}(n - \tfrac{1}{2})\big]_i = \Bigg[\sum_{\ell \in \mathbb{Z}} u\big(2^j\ell - n + \tfrac{1}{2}\big)\Bigg]_i &= \sum_{\ell \in \mathbb{Z}} \big[u\big(2^j\ell - n + \tfrac{1}{2}\big)\big]_i = \\ &(-1)^i \sum_{\ell \in \mathbb{Z}} h\big[i + d\big(2^j\ell - n + \tfrac{1}{2}\big)\big] = \\ &\quad - (-1)^{i-1} \sum_{\ell \in \mathbb{Z}} h\big[(i-1) + d\big(2^j\ell - n\big)\big] = -\big[\overrightarrow{\mathbf{uj}}(n)\big]_{i-1}.\end{aligned}$$

□

Remark 5.24. Proposition 5.23 tells us that when it comes to the computation of the vectors $\overrightarrow{\mathbf{uj}}\big(n - \frac{\delta}{2}\big)$ we only need to compute the values when $\delta = 0$ and only compute the first component when $\delta = 1$. ■

Remark 5.25. Recall that $n \in \{0, \dots, 2^j - 1\}$, hence if $2^j > p$, we have that

$$\ell_{\min} = \begin{cases} 0 & \text{if } n \le p - 1 + \frac{\delta}{2} \\ 1 & \text{if } n > p - 1 + \frac{\delta}{2} \end{cases} \qquad \ell_{\max} = \begin{cases} 0 & \text{if } n \le 2^j - p + \frac{\delta+1}{2} \\ 1 & \text{if } n > 2^j - p + \frac{\delta+1}{2} \end{cases}$$

Consequently, if $n \in \big[p - 1, 2^j - p + 1\big] \cap \mathbb{Z}$ it follows that $\ell_{\min} > \ell_{\max}$, therefore we get $\overrightarrow{\mathbf{uj}}\big(n - \frac{\delta}{2}\big) = \overrightarrow{\mathbf{0}}$. In Figure 5.1 one can see a sketch of this fact. Moreover, by Corollaries 5.20 and 3.22 we can determine the values of θ_i, j and n such that $\psi^{\mathrm{PER}}_{-j,n}(\theta_i) = 0$ beforehand, since $\overrightarrow{\mathbf{uj}}\big(n\big) = \overrightarrow{\mathbf{0}}$ implies that $\psi^{\mathrm{PER}}_{-j,n}(\theta_i) = 0$ for $\theta_i = 0, \dots, \frac{1}{2^{j+1}}$. This is coherent with Remark 3.18 that tells us that support of the periodized wavelets keeps getting smaller as j increases. ■

5.4.2.2. *Algorithm to compute* $\psi^{\mathrm{PER}}_{-j,n}$ *on dyadic rationals of order* ν. Now we are finally ready to give an explicit dyadic algorithm for the non rotated $\psi^{\mathrm{PER}}_{-j,n}$ case. It follows the same notation and structure as the previous Algorithms 5.13 and 5.17.

Algorithm 5.26 (Computation of $\psi^{\mathrm{PER}}_{-j,n}(\theta)$ for dyadic $\theta \in \mathbb{S}^1$). This algorithm is an implementation of Corollary 5.20. Here the definitions of $m_{\max}$ are as in Subsection 5.3.3.

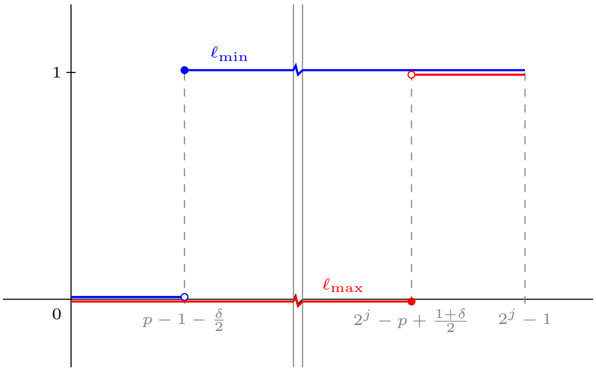


FIGURE 5.1. Graphs of $\ell_{\min}$ and $\ell_{\max}$ as functions of n when $2^j > p$. Note the area in the centre where $\ell_{min} > \ell_{\max}$.

Initialize: pre-compute the vectors $\overrightarrow{\mathbf{uj}}(n)$ and $\left[\overrightarrow{\mathbf{uj}}(n-\frac{1}{2})\right]_0$ for $n = 0, \dots, 2^{j-1}$ using Proposition 5.23.

Initialize: pre-compute 2^ν and the coefficient $\frac{\sqrt{2}^{j+1}}{2p-1}$

Step 0: Compute $\psi^{\text{PER}}_{j,n}(0)$ and $\psi^{\text{PER}}_{i,j}\left(\frac{1}{2^{j+1}}\right)$ for $n = 0, \dots, 2^j - 1$ using Corollary 5.20.

Initialize: $q = 1$

Step 1: Compute the Daubechies–Lagarias Products Vector

$$\overrightarrow{\mathsf{DL}}(q) := \mathbf{M}_1^{b_\kappa} \cdot \mathbf{M}_0^{b_{\kappa-1}} \cdots \mathbf{M}_0^{b_2} \cdot \mathbf{M}_1^{b_1} \cdot \vec{\mathbf{V}}^{(0)} \tag{5.17}$$

with $\mathsf{binary}(q) = \mathbf{1}^{b_\kappa}\mathbf{0}^{b_{\kappa-1}} \cdots \mathbf{0}^{b_2}\mathbf{1}^{b_1}$. Moreover, by Equation (5.10) set

$$m_{\max} = (\nu - 1) - j - \mathsf{lb}$$

Step 2: Compute $\psi^{\text{PER}}_{j,n}(\theta)$ and $\psi^{\text{PER}}_{j,n}\left(\theta + \frac{1}{2^{j+1}}\right)$ with $\theta = \frac{2^{m_{\max}}q}{2^\nu}$ for the integers $\ell = 1-p, \dots, p-1$, using Corollary 5.20 by

$$\psi^{\text{PER}}_{-j,n}(\theta) = \frac{\sqrt{2}^{j+1}}{2p-1}\overrightarrow{\mathbf{uj}}(n) \cdot \overrightarrow{\mathsf{DL}}(q) \qquad \text{and,}$$
$$\psi^{\text{PER}}_{-j,n}\left(\theta + \frac{1}{2^{j+1}}\right) = -\frac{\sqrt{2}^{j+1}}{2p-1}\overrightarrow{\mathbf{uj}}\left(n - \frac{1}{2}\right) \cdot \overrightarrow{\mathsf{DL}}(q).$$

Where the product by $\overrightarrow{\mathbf{uj}}\left(n + \frac{1}{2}\right)$ is done using Proposition 5.23 (b) and (c).

Step 3: For every $m = m_{\max -1}, m_{\max -2}, \dots, 0$ (if any) set

$$\overrightarrow{\mathsf{DL}}(q) = \mathbf{M}_0\overrightarrow{\mathsf{DL}}(q).$$

and repeat Step 2 to for $\theta = \frac{2^m q}{2^\nu}$. This follows because

$$\lambda(q, m+j-1, \nu) = \lambda(q, m+j, \nu) + 1.$$

Iterate: $q \leftarrow q + 2$; `if` $q < 2^{\nu-j-1}$ `goto` **Step 1**

■

5.5. The Rotated Wavelet Value

This section is completely analogous to the previous one, but dealing with $\psi^{\text{PER}}_{j,n}\left(R_\omega(\theta_i)\right)$, where R_ω is as in Equation (2.1) the irrational rotation by ω. That is, as in Equation (2.1), given by

$$\begin{aligned} R_\omega(\theta) &= \theta + \omega \mod 1 \\ &= \{\!\{\theta + \omega\}\!\} \\ &= \theta + \omega - \lfloor \theta + \omega \rfloor. \end{aligned}$$

Since all the systems studied usually have $\omega = \frac{1}{2}(\sqrt{5} - 1) \approx 0.61803$ [**AMR16, Kan84, NK96, AM08, AM15, Kel96**] we will assume the following condition on ω:

$$2\omega = \{\!\{2\omega\}\!\} + 1 \tag{5.18}$$

Note that this condition is equivalent to asking that $\frac{1}{2} < \omega < 1$. Still related to ω, let us define

$$\widetilde{\omega} := \{\!\{2\omega\}\!\}, \tag{5.19}$$

which for all of our cases $\widetilde{\omega} = \sqrt{5} - 2 \approx 0.236067978$.

Moreover, since we are interested in using the Daubechies Lagarias algorithm on irrational points, we have to move forward with some precaution. First and foremost, we relied heavily on the fact that we were using dyadic numbers, and hence their dyadic expansion was finite. Thus, we used the eigenvector $\overrightarrow{\mathbf{V}}^{(0)}$ as a placeholder for whatever zeroes may lay on the right hand size of the limit. In this case, we have to tread more carefully, considering we are using an irrational number, which has an infinite dyadic expansion. To this end, we will introduce a series of vectors $\overrightarrow{\mathbf{V}}_k^{(\omega)}$ that will play a similar role to $\overrightarrow{\mathbf{V}}^{(0)}$. However, as we shall see these have to be precomputed for various values of k.

As we have been doing in the previous sections, we will treat the dyadic numbers as integers. The underlying reason for this is mostly for computational efficiency. On one hand, with this setup we know exactly where the influence of the dyadic points begins, so we can better use the vectors $\overrightarrow{\mathbf{V}}_k^{(\omega)}$. On the other hand, we can optimize the memory usage of our computer, by using only the amount of memory required at each computation. If we were to use double precision floating point arithmetic (64 bits), we would always need to save the whole dyadic number as an entire floating point, even if we are limited to matrices of size 2^{30}, hence actually needing dyadics up to order $\nu = 30$, which need at most 30 bits. Therefore, even though the use of integer arithmetic for storing rotated dyadic rationals might involve higher theoretical preparation and may seem more involved than the ones used for plain dyadic numbers, we believe it still poses advantages in terms of speed and memory efficiency.

Additionally, the way we have designed the computations, we are positive that the precision in which we can compute the rotated wavelet values almost corresponds to the maximum precision allowed by the data type used. This stems from two facts. One one hand, the techniques used minimize all other possible errors, since the dyadic approach allows us to deal with the irrationality almost as if we were working with rational numbers. On the other hand, the errors corresponding to the floating point arithmetic can be vanquished by using a floating point with higher precision for certain operations and then truncating to the actually desired precision for the final result (such as using `__float128` for the matrix products regarding $\mathbf{M}_0$ and $\mathbf{M}_1$ and then storing the obtained result as a `double`).

Furthermore, when considering how one computes and stores the dyadic expansions of the points, it can be useful to take into account the maximum precision to be used, that is the number of digits we will work with. To this end, we work in a framework in which all the values used have a maximum precision $\nu^\star$, i.e. the number of digits of the dyadic expansion of ω we are willing to store.

For the following subsections will proceed in a similar manner as before, first computing the periodic mother wavelet ψ^{PER} on dyadic rationals rotated by an *irrational* rotation, followed by the computation of $\psi_{i,j}^{\mathrm{PER}}$ on the same rotated dyadic points.

5.5.1. Computation of the Rotated ψ^{PER}. Now we are finally ready to start computing some wavelet values. The main goal of this subsection is to give an algorithm to evaluate ψ^{PER} on rotated dyadic rationals. As we will see, we will obtain a very similar results to Corollary 5.14 but with a caveat: the effects of the rotation distort the simplicity of Corollary 5.14 on the matrix product part **DL**. This mostly stems from the fact that $\frac{1}{2} < \omega < 1$. Therefore, when computing the matrix products we have to be careful, since if $1 - \omega < \theta < \frac{1}{2}$ it follows that $R_\omega(\theta) = \theta + \omega - 1$. This does not complicate that much the general Daubechies-Lagarias formulae. However, it requires a lot of care when considering their dyadic version. In particular, one needs to find a dyadic number $\frac{\tau(\nu)}{2^\nu}$ such that $\omega + \frac{\tau(\nu)-1}{2^\nu} < 1 < \omega + \frac{\tau(\nu)}{2^\nu}$. It is self evident that this number τ depends on both ν and ω. The computation of $\tau(\nu)$ becomes one of the crucial factors in being able to write the dyadic version of the Daubechies-Lagarias formulae. In fact the whole Subsubsection 5.5.1.2 is almost exclusively devoted to efficiently computing $\tau(\nu)$.

Let us begin by giving the analogous of Corollary 5.14 for ψ^{PER}.

Corollary 5.27. *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments. Then for every $\theta \in \mathcal{S}^1$,*

$$\begin{aligned}\psi^{\mathrm{PER}}(R_\omega(\theta)) &= \sum_{\ell\in\mathbb{Z}} \psi(R_\omega(\theta) + \ell)\\ &= -(-1)^{d\left(R_\omega(\theta)\right)} \tfrac{\sqrt{2}}{2p-1} \left(\sum_{\ell\in\mathbb{Z}} u(R_\omega(\theta)+\ell)\right)^{\mathsf{T}} \mathbf{DL}\big(\{\!\{2\theta + \widetilde{\omega}\}\!\}\big)\,\vec{\mathbf{1}}.\end{aligned}$$

Proof. The statement follows from the straightforward application of Corollary 5.14 and the fact that

$$\begin{aligned}\{\!\{2R_\omega(\theta)\}\!\} &= \big\{\!\!\big\{2(\theta + \omega - \lfloor \theta+\omega \rfloor)\big\}\!\!\big\} = \big\{\!\!\big\{2\theta + \widetilde{\omega} + 1 - 2\lfloor \theta + \omega \rfloor\big\}\!\!\big\}\\ &= \{\!\{2\theta + \widetilde{\omega}\}\!\}\end{aligned}$$

□

Let us prove a technical proposition that will pave the way to obtaining a Daubechies-Lagarias computation for $\Psi^{\mathrm{PER}}\big(R_\omega(\theta)\big)$ similar to Corollary 5.14.

Proposition 5.28. *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments. Then, $\psi^{\mathrm{PER}}\big(R_\omega(\theta)\big) = \sum_{\ell=1-p}^{p-1} \psi\big(R_\omega(\theta) + \ell\big)$ for every $x \in \mathbb{S}^1$. Moreover, the following statements hold for every $\theta \in \big[0, \frac{1}{2}\big)$:*

(a) $d\big(R_\omega(x)\big) = \begin{cases} 0 & \text{if } \theta \in [0, 1-\omega), \\ 1 & \text{if } \theta \in \big[1-\omega, \frac{1}{2}\big). \end{cases}$

(b)
$$\sum_{\ell\in\mathbb{Z}} u\big(R_\omega(\theta) + \ell\big) = \frac{\sqrt{2}}{2}\, \vec{\boldsymbol{ua}}_{2p-1} = \sum_{\ell\in\mathbb{Z}} u\big(R_\omega\big(\theta + \tfrac{1}{2}\big) + \ell\big).$$

(c) $\psi^{\mathrm{PER}}\big(R_\omega(\theta)\big) = -\psi^{\mathrm{PER}}\big(R_\omega\big(\theta + \frac{1}{2}\big)\big)$.

Proof. The first statement follows from Corollary 5.14.

To prove (a) we need to compute $d\big(R_\omega(\theta)\big)$ in terms of θ. We have

$$\big\lfloor 2R_\omega(\theta)\big\rfloor = \lfloor 2(\theta + \omega - \lfloor \theta + \omega \rfloor)\rfloor = \lfloor 2(\theta+\omega)\rfloor - 2\lfloor \theta + \omega \rfloor.$$

On the other hand, $\frac{1}{2} < \omega < 1$ implies that $1 - \omega < \frac{1}{2}$. Thus, if $\theta \in [0, 1 - \omega)$ (equivalently $\omega \leq \theta + \omega < 1$), $1 < 2\omega \leq 2(\theta + \omega) < 2$, and hence, $\lfloor 2(\theta + \omega) \rfloor = 1$ and $\lfloor \theta + \omega \rfloor = 0$. If $\theta \in \big[1 - \omega, \frac{1}{2}\big)$ we obtain

$$1 \leq \theta + \omega < \tfrac{1}{2} + \omega < \tfrac{3}{2} \qquad \text{and} \qquad 2 \leq 2(\theta + \omega) < 3,$$

and hence, $\lfloor 2(\theta + \omega) \rfloor = 2$ and $\lfloor \theta + \omega \rfloor = 1$. Consequently,

$$\big\lfloor 2R_\omega(\theta) \big\rfloor = \begin{cases} 1 & \text{if } \theta \in [0, 1 - \omega), \\ 0 & \text{if } \theta \in \big[1 - \omega, \frac{1}{2}\big), \end{cases}$$

and thus,

$$d\big(R_\omega(\theta)\big) = 1 - \big\lfloor 2R_\omega(\theta) \big\rfloor = \begin{cases} 0 & \text{if } \theta \in [0, 1 - \omega), \text{ and} \\ 1 & \text{if } \theta \in \big[1 - \omega, \frac{1}{2}\big). \end{cases}$$

Now we prove (b). To do this we need to repeat the above computation for $d\big(R_\omega\big(\theta + \frac{1}{2}\big)\big)$. Observe that $0 \leq \theta < \frac{1}{2}$ is equivalent to

$$1 < \tfrac{1}{2} + \omega \leq \theta + \tfrac{1}{2} + \omega < 1 + \omega < 2,$$

which implies that $\big\lfloor \theta + \frac{1}{2} + \omega \big\rfloor = 1$. Hence

$$\big\lfloor 2R_\omega\big(\theta + \tfrac{1}{2}\big) \big\rfloor = \big\lfloor 2(\theta + \tfrac{1}{2} + \omega) \big\rfloor - 2\big\lfloor \theta + \tfrac{1}{2} + \omega \big\rfloor = \lfloor 2(\theta + \omega) \rfloor - 1,$$

and

$$d\big(R_\omega\big(\theta + \tfrac{1}{2}\big)\big) = 1 - \big\lfloor 2R_\omega\big(\theta + \tfrac{1}{2}\big) \big\rfloor = 2 - \lfloor 2(\theta + \omega) \rfloor =$$
$$2 - \left\{ \begin{matrix} 1 & \text{if } \theta \in [0, 1 - \omega) \\ 2 & \text{if } \theta \in \big[1 - \omega, \frac{1}{2}\big) \end{matrix} \right\} = \big\lfloor 2R_\omega(\theta) \big\rfloor = 1 - d\big(R_\omega(\theta)\big).$$

Now, as before, let us denote by $u_i(x) := (-1)^i h[i + d(\theta)]$ the i-th component of the vector $u(\theta)$ with $i \in \{0, 1, \dots, 2p - 2\}$. Moreover, recall that $d(\theta + \ell) = d(\theta) - 2\ell$ for every $x \in \mathbb{R}$ and $\ell \in \mathbb{Z}$, and $d\big(R_\omega(\theta)\big) \in \{0, 1\}$. Then, by Remark 3.8(b),

$$\begin{aligned} \sum_{\ell \in \mathbb{Z}} u_i\big(R_\omega(x) + \ell\big) &= \sum_{\ell \in \mathbb{Z}} (-1)^i \, h\big[i + d\big(R_\omega(x) + \ell\big)\big] \\ &= (-1)^i \sum_{\ell \in \mathbb{Z}} h\big[i + d\big(R_\omega(x)\big) - 2\ell\big)\big] \\ &= (-1)^i \frac{\sqrt{2}}{2} \\ &= (-1)^i \sum_{\ell \in \mathbb{Z}} h\big[i + 1 - d\big(R_\omega(x)\big) - 2\ell\big)\big] \\ &= \sum_{\ell \in \mathbb{Z}} (-1)^i \, h\big[i + d\big(R_\omega\big(x + \tfrac{1}{2}\big)\big) - 2\ell\big] \\ &= \sum_{\ell \in \mathbb{Z}} (-1)^i \, h\big[i + d\big(R_\omega\big(x + \tfrac{1}{2}\big) + \ell\big)\big] \\ &= \sum_{\ell \in \mathbb{Z}} u_i\big(R_\omega\big(x + \tfrac{1}{2}\big) + \ell\big). \end{aligned}$$

Finally, (c) follows directly from (b), Corollary 5.27 and

$$\begin{aligned} &\big\{\!\!\big\{ 2\big(\theta + \tfrac{1}{2}\big) + \widetilde{\omega} \big\}\!\!\big\} = \{\!\{ 2\theta + 1 + \widetilde{\omega} \}\!\} = \{\!\{ 2\theta + \widetilde{\omega} \}\!\}, \text{ and} \\ &(-1)^{d\left(R_\omega\left(x + \frac{1}{2}\right)\right)} = (-1)^{1 - d\left(R_\omega(\theta)\right)} = -(-1)^{d\left(R_\omega(\theta)\right)}. \end{aligned}$$

□

Thanks to this proposition we can finally get explicit Daubechies-Lagarias formulae for a general $\psi^{\text{PER}}(R_\omega(\theta))$, similar to Corollary 5.27. Note that, as already mentioned in the introduction of this section, the fact that $1-\omega \in [0,1/2]$ forces us to consider two different cases.

Corollary 5.29. *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments. Assume that $\theta \in [0, 1-\omega)$. Then, $\{\{2\theta + \widetilde{\omega}\}\}= 2\theta + \widetilde{\omega}$ and*

$$\psi^{\text{PER}}\big(R_\omega(\theta)\big) = -\tfrac{1}{2p-1}\ \vec{\boldsymbol{ua}}^{\mathsf{T}}_{2p-1}\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}(2\theta+\widetilde{\omega},i)}\right)\vec{\mathbf{1}}.$$

Assume that $\theta \in \big[1-\omega, \frac{1}{2}\big)$. Then, $\{\{2\theta + \widetilde{\omega}\}\}= 2\theta + \widetilde{\omega} - 1$ and

$$\psi^{\text{PER}}\big(R_\omega(\theta)\big) = \tfrac{1}{2p-1}\ \vec{\boldsymbol{ua}}^{\mathsf{T}}_{2p-1}\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}(2\theta+\widetilde{\omega}-1,i)}\right)\vec{\mathbf{1}}.$$

Proof. When $\theta \in [0, 1-\omega)$ we have $\{\{2\theta + \widetilde{\omega}\}\}= 2\theta + \widetilde{\omega}$ because

$$0 < \widetilde{\omega} < 2\theta + \widetilde{\omega} < 2 - 2\omega + \widetilde{\omega} = 1.$$

On the other hand, when $\theta \in \big[1-\omega, \frac{1}{2}\big)$, $\{\{2\theta + \widetilde{\omega}\}\}= 2\theta + \widetilde{\omega} - 1$ because

$$1 = 2 - (\widetilde{\omega}+1) + \widetilde{\omega} = 2 - 2\omega + \widetilde{\omega} \leq 2\theta + \widetilde{\omega} < 1 + \widetilde{\omega} < 2.$$

Then, the corollary follows from Corollary 5.27 and Proposition 5.28(a,b). □

The next task is to obtain a version of Corollary 5.29 for dyadic values of θ. To do this we have to introduce the necessary notation that will be used to state and prove the result we are looking for. This will be done in the next subsection.

5.5.1.1. *On storing rotated dyadic numbers as an integers.* Before going on with the computation of the wavelet values, let us introduce with some general considerations on how we are going to efficiently deal with using integers to store dyadic numbers rotated by the irrational rotation R_ω. To archieve this, we need some notation.

First of all, since we will be dealing with irrational nubers, we would like to fix a certain number of digits to work with. This will be done by forcing our binary expansions (that in the dyadic world correspond to the dyadic expansions of θ) to have a fixed finite length.

Definition 5.30 (ℓ-binary expansion). Given a non-negative integer m we define the *ℓ-digit binary expansion* of m, for short *ℓ-binary expansion* of m, as the sequence $b_{\ell-1}b_{\ell-2}\ldots b_1b_0 \in \{0,1\}^\ell$ such that $m = \sum_{i=0}^{\ell-1} b_i 2^i$. The ℓ-digit binary expansion of m will be denoted by $\mathsf{binary}_\ell(m)$. Observe that $\ell \geq \mathsf{length}(\mathsf{binary}(m))$ and, hence,

$$\mathsf{binary}_\ell(m) = \mathbf{0}^{\ell-\mathsf{length}(\mathsf{binary}(m))}\,\mathsf{binary}(m).$$

■

In the following remark we will see how the dyadic expansion of a rotated dyadic number and the binary expansion of its integer part relate to one another.

Remark 5.31. We want to explore how one might deal with irrational numbers in a dyadic setting. Suppose we have $\alpha \in [0,1)$. In particular

$$\alpha = 0.\,\mathsf{dyad}(\alpha).$$

However, suppose we want to use integer arithmetic for the first $\nu^\star$ digits of the expansion, similarly to what we did when considering dyadic rationals using integer arithmetic. This can be done by defining $\bar{\alpha} = 2^{\nu^\star}\alpha$. Trivially $\alpha = \bar{\alpha}/2^{\nu^\star}$. It is easy to see that we get $\bar{\alpha} = \lfloor\bar{\alpha}\rfloor + \{\{\bar{\alpha}\}\}$, where $\lfloor\bar{\alpha}\rfloor < 2^{\nu^\star}$. In this setting, we can operate with the dyadic-as-integers framework on the integer $\lfloor\bar{\alpha}\rfloor$ while still taking into account the effects of $\{\!\{\bar{\alpha}\}\!\}$ (for example through the vectors $\vec{\mathbf{V}}_n^{(\widetilde{\omega})}$ we will define in Subsubsection 5.5.1.2). Moreover,

$$\begin{aligned}\mathsf{dyad}\left(\frac{\bar{\alpha}}{2^{\nu^\star}}\right) &= \mathsf{binary}_\ell\big(\lfloor\bar{\alpha}\rfloor\big)\,\mathsf{dyad}\big(\{\!\{\bar{\alpha}\}\!\}\big)\\ &= \mathbf{0}^{\ell-\mathsf{length}(\mathsf{binary}(\lfloor\alpha\rfloor))}\,\mathsf{binary}\big(\lfloor\bar{\alpha}\rfloor\big)\,\mathsf{dyad}\big(\{\!\{\bar{\alpha}\}\!\}\big)\\ &= \mathsf{dyad}(\alpha).\end{aligned}$$

This will be used in the next subsubsection to use the irrational number $\widetilde{\omega}$ as an integer to adapt the rotated case to a dyadic setting. ∎

Since we are interested in using matrices Ψ and Ψ_R with the largest possible dimension 2^ν we will assume that $\nu \geq 5$ from now on.

Let $\nu^\star > \nu$ be an upper bound for ν. This $\nu^\star$ should be understood as the length of the dyadic expansion of $\widetilde{\omega}$ we will store. This is to help with computing $\lim_{n\to\infty}\prod_{i=1}^n \mathbf{M}_{\mathsf{dyad}(\{\!\{2\theta+\widetilde{\omega}\}\!\},i)}$ for several values of ν without having to consider each case separately. In fact, computation-wise $\nu^\star$ should be taken as the bit length of the data type used to store the dyadic numerators and to perform all related arithmetic. Since we use the C programing language, we use the `unsigned int` type for this purpose. Hence

$$\nu^\star = 8\cdot\texttt{sizeof(unsigned int)} = 32.$$

Recall that the maximum number that can be stored in 32 bits is $2^{\nu^\star}-1 = 2^{32}-1$.

5.5.1.2. *On the effcient computation of the matrix products.* In this subsubsection, we would like to obtain a version of Corollary 5.29 focusing on dyadic values. To this end we will implement the strategy used so far to consider the dyadic numbers as integers divided by 2^ν, for some $\nu \in \mathbb{N}$ such that $\nu < \nu^\star$ ($\nu^\star$ being the maximum precision have we set). Moreover, we need to find a way to compute $\tau(\nu)$ as mentioned in the introduction to this section. In what follows we set

$$2^{\nu^\star}\widetilde{\omega} = \varpi + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\} \tag{5.20}$$

where $\varpi := \lfloor 2^{\nu^\star}\widetilde{\omega}\rfloor$. This is the first crucial step to deal with irrational numbers in an integer setting. Note that

$$\mathsf{binary}_{\nu^\star}(\varpi) := b_{\nu^\star-1}b_{\nu^\star-2}b_{\nu^\star-3}b_{\nu^\star-4}\ldots b_1b_0$$

corresponds to the dyadic expansion of $\widetilde{\omega}$ up to order $\nu^\star$. Moreover, for $r \in \{1,2,\ldots,\nu^\star\}$,

$$\widetilde{\varpi}^r := \sum_{i=0}^{r-1} b_i 2^i \in \mathbb{Z}.$$

Hence, $\mathsf{binary}_r\big(\widetilde{\varpi}^r\big) = b_{r-1}\ldots b_1b_0$. Therefore, the various $\widetilde{\varpi}^r$ correspond to the *tails* of the dyadic expression of ϖ:

$$\overbrace{b_{\nu^\star-1}b_{\nu^\star-2}\ldots\underbrace{b_r\ldots b_1b_0}_{\widetilde{\varpi}^r}}^{\varpi}$$

Observe that $\widetilde{\varpi}^{\nu^\star} = \varpi$ and $\widetilde{\varpi}^r < 2^r$. Hence, for $r \leq s \leq \nu^\star$,

$$\frac{\widetilde{\varpi}^s - \widetilde{\varpi}^r}{2^r} \leq \frac{\widetilde{\varpi}^s}{2^r} < \frac{\widetilde{\varpi}^s + 2^r - \widetilde{\varpi}^r}{2^r} = \frac{\widetilde{\varpi}^s - \widetilde{\varpi}^r}{2^r} + 1.$$

Moreover, from the definition of $\widetilde{\varpi}^r$ and $\widetilde{\varpi}^s$ it follows that $\frac{\widetilde{\varpi}^s - \widetilde{\varpi}^r}{2^r} \in \mathbb{Z}^+$. Thus,

$$\widetilde{\varpi}^s = 2^r\frac{\widetilde{\varpi}^s - \widetilde{\varpi}^r}{2^r} + \widetilde{\varpi}^r = 2^r\big\lfloor\tfrac{\widetilde{\varpi}^s}{2^r}\big\rfloor + \widetilde{\varpi}^r. \tag{5.21}$$

In particular, $\widetilde{\varpi}^r = \widetilde{\varpi}^s \pmod{2^r}$.

Moreover, let us denote $\nu^{\mathsf{d}} := \nu^\star + 1 - \nu \geq 2$. Note that ν^{d} corresponds to the shift one needs to apply to the binary expansion of length $\nu^\star$ in order to land on the first digit that gives an expression of length ν:

$$\mathsf{binary}(a) = \underbrace{a_{\nu^\star-1}a_{\nu^\star-2}a_{\nu^\star-3}\dots a_{\nu^{\mathsf{d}}}a_{\nu^{\mathsf{d}}-1}}_{\text{Binary expansion of the first }\nu\text{ digits}}\overbrace{\;a_{\nu^{\mathsf{d}}-2}\dots a_1a_0}^{\text{Shift of length }\nu^{\mathsf{d}}}$$

Finally, $\tau(\nu) := 2^{\nu-1} - \left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor \in \mathbb{Z}$.

Remark 5.32. By (5.21),

$$\left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor 2^{\nu^{\mathsf{d}}} \leq \varpi = \lfloor 2^{\nu^\star}\widetilde{\omega} \rfloor < 2^{\nu^\star}\widetilde{\omega} =$$
$$\varpi + \{\{2^{\nu^\star}\widetilde{\omega}\}\} = 2^{\nu^{\mathsf{d}}}\left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor + \widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\{2^{\nu^\star}\widetilde{\omega}\}\} \leq$$
$$2^{\nu^{\mathsf{d}}}\left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor + \left(2^{\nu^{\mathsf{d}}} - 1\right) + \{\{2^{\nu^\star}\widetilde{\omega}\}\} < \left(\left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor + 1\right)2^{\nu^{\mathsf{d}}}.$$

Hence,

$$\tau(\nu) - 1 = 2^{\nu-1} - \left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor - 1 < 2^{\nu-1} - \frac{\varpi + \{\{2^{\nu^\star}\widetilde{\omega}\}\}}{2^{\nu^{\mathsf{d}}}} =$$
$$2^{\nu-1} - \frac{2^{\nu-1}}{2^{\nu^\star}}\left(\varpi + \{\{2^{\nu^\star}\widetilde{\omega}\}\}\right) = 2^{\nu-1}\left(1 - \widetilde{\omega}\right) < 2^{\nu-1} - \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \leq \tau(\nu).$$

Consequently,

$$\tfrac{q}{2^\nu} > 1 - \omega = \tfrac{1-\widetilde{\omega}}{2} \quad \text{is equivalent to} \quad q \geq \tau(\nu).$$

Observe also that, since $\nu \geq 5$,

$$0 < 2^{\nu-2} < 2^{\nu-1}\left(1 - \widetilde{\omega}\right) < \tau(\nu) < 2^{\nu-1}\left(1 - \widetilde{\omega}\right) + 1 < 4\tfrac{2^{\nu-1}}{5} + \tfrac{2^{\nu-1}}{5} = 2^{\nu-1}.$$

■

Now we will define a series of vectors $\overrightarrow{\mathbf{V}}_k^{(\widetilde{\omega})}$, $k \in \{0, \dots, \nu^\star\}$ that will play an analogous role that the vector $\overrightarrow{\mathbf{V}}^{(0)}$ had in the non rotated case. However, since we are dealing with an irrational rotation, in this case we cannot simply assume that there is an infinite series of zeroes on the right hand side. Hence, we need to define different vectors depending on the value of ν we will use.

$$\overrightarrow{\mathbf{V}}_0^{(\widetilde{\omega})} := \lim_{n\to\infty}\left(\mathbf{M}_{\mathsf{dyad}(\{\{2^{\nu^\star}\widetilde{\omega}\}\},1)} \cdot \mathbf{M}_{\mathsf{dyad}(\{\{2^{\nu^\star}\widetilde{\omega}\}\},2)} \cdots \mathbf{M}_{\mathsf{dyad}(\{\{2^{\nu^\star}\widetilde{\omega}\}\},n)} \cdot \overrightarrow{\mathbf{1}}\right) =$$
$$\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}(\{\{2^{\nu^\star}\widetilde{\omega}\}\},i)}\right)\overrightarrow{\mathbf{1}},$$

and for $k = 1, 2, \dots, \nu^\star$,

$$\overrightarrow{\mathbf{V}}_k^{(\widetilde{\omega})} := M_{b_{k-1}}\overrightarrow{\mathbf{V}}_{k-1}^{(\widetilde{\omega})}.$$

The inequality $\widetilde{\varpi}^k < 2^k$, implies $\frac{\widetilde{\varpi}^k + \{\{2^{\nu^\star}\widetilde{\omega}\}\}}{2^k} \in [0,1)$ and hence,

$$\mathsf{dyad}\left(\frac{\widetilde{\varpi}^k + \{\{2^{\nu^\star}\widetilde{\omega}\}\}}{2^k}\right) = \mathsf{binary}_k\left(\widetilde{\varpi}^k\right)\mathsf{dyad}\left(\{\{2^{\nu^\star}\widetilde{\omega}\}\}\right).$$

So,

$$\overrightarrow{\mathbf{V}}_k^{(\widetilde{\omega})} = \left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}\left(\frac{\widetilde{\varpi}^k + \{\{2^{\nu^\star}\widetilde{\omega}\}\}}{2^k},i\right)}\right)\overrightarrow{\mathbf{1}}. \tag{5.22}$$

In particular,

$$
\begin{aligned}
\vec{\mathbf{V}}^{(\widetilde{\omega})}_{\nu^\star} = \left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}\left(\frac{\varpi^{\nu^\star}+\{\{2^{\nu^\star}\widetilde{\omega}\}\}}{2^{\nu^\star}},i\right)}\right)\vec{\mathbf{1}} = \\
\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}\left(\frac{\varpi+\{\{2^{\nu^\star}\widetilde{\omega}\}\}}{2^{\nu^\star}},i\right)}\right)\vec{\mathbf{1}} = \\
\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}\left(\widetilde{\omega},i\right)}\right)\vec{\mathbf{1}}.
\end{aligned}
\tag{5.23}
$$

Where the last equation follows from Equation 5.20.

Corollary 5.33 (Corollary 5.29 - Dyadic version). *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments. Then,*

$$\psi^{\mathrm{PER}}\big(R_\omega(0)\big) = \tfrac{1}{2p-1}\ \vec{\boldsymbol{ua}}^{\mathsf{T}}_{2p-1}\vec{\mathbf{V}}^{(\widetilde{\omega})}_{\nu^\star},$$

and

$$\psi^{\mathrm{PER}}\left(R_\omega\left(\tfrac{\tau(\nu)}{2^\nu}\right)\right) = -\tfrac{1}{2p-1}\ \vec{\boldsymbol{ua}}^{\mathsf{T}}_{2p-1}\left(\mathbf{M}_0^{\nu-1}\cdot\vec{\mathbf{V}}^{(\widetilde{\omega})}_{\nu^{\mathsf{d}}}\right).$$

Moreover, for $x = \frac{q}{2^\nu} < \frac{1}{2}$ with $q \in \mathbb{N}\setminus\{\tau(\nu)\}$,

$$
\begin{aligned}
\psi^{\mathrm{PER}}\big(R_\omega\big(\tfrac{q}{2^\nu}\big)\big) = (-1)^d\tfrac{1}{2p-1} \\
\vec{\boldsymbol{ua}}^{\mathsf{T}}_{2p-1}\left(\mathbf{M}_0^{\nu-1-\mathsf{lbq}}\cdot\mathbf{M}_1\cdot\mathbf{M}_{a_{\mathsf{lbq}-2}}\cdot\mathbf{M}_{a_{\mathsf{lbq}-3}}\cdots\mathbf{M}_{a_1}\cdot\mathbf{M}_{a_0}\cdot\vec{\mathbf{V}}^{(\widetilde{\omega})}_{\nu^{\mathsf{d}}}\right),
\end{aligned}
$$

with

$$d = 0 \quad \textit{and} \quad \mathsf{binary}\left(q + \left\lfloor\frac{\varpi}{2^{\nu^{\mathsf{d}}}}\right\rfloor\right) = 1a_{\mathsf{lbq}-2}a_{\mathsf{lbq}-3}\cdots a_1 a_0$$

for $q < \tau(\nu)$, and

$$d = 1 \quad \textit{and} \quad \mathsf{binary}(q-\tau(\nu)) = 1a_{\mathsf{lbq}-2}a_{\mathsf{lbq}-3}\cdots a_1 a_0$$

whenever $\tau(\nu) < q < 2^{\nu-1}$.

Remark 5.34. When $\tau(\nu) \leq q < 2^{\nu-1}$, by Remark 5.32 we have

$$0 \leq q - \tau(\nu) < 2^{\nu-1} - \tau(\nu) = \left\lfloor\frac{\varpi}{2^{\nu^{\mathsf{d}}}}\right\rfloor < 2^{\nu^\star}\widetilde{\omega} < 2^{\nu^\star}.$$

Consequently, $q - \tau(\nu)$ fits in a $\nu^\star$−bit unsigned integer.

When $0 < q < \tau(\nu)$ we have

$$0 < \left\lfloor\frac{\varpi}{2^{\nu^{\mathsf{d}}}}\right\rfloor < q + \left\lfloor\frac{\varpi}{2^{\nu^{\mathsf{d}}}}\right\rfloor < \tau(\nu) + \left\lfloor\frac{\varpi}{2^{\nu^{\mathsf{d}}}}\right\rfloor = 2^{\nu-1} \leq 2^{\nu^\star-1}.$$

Consequently, $q + \left\lfloor\frac{\varpi}{2^{\nu^{\mathsf{d}}}}\right\rfloor$ also fits in a $\nu^\star$−bit unsigned integer. ∎

Proof of Corollary 5.29 dyadic version. The first statement (case $x = 0$) follows from Corollary 5.29and (5.23).

Now, in view of Corollary 5.29 and Remark 5.32 we essentially need to compute the full dyadic expression of $\{\{2x + \widetilde{\omega}\}\}$ for the dyadic rationals

$$x = \frac{q}{2^\nu} < \frac{1}{2} = \frac{2^{\nu-1}}{2^\nu}$$

with $q \in \{1, 2, \dots, 2^{\nu-1}-1\}$ (observe that since $q < 2^{\nu-1}$, $\mathsf{length}(\mathsf{binary}(q)) \leq \nu-1$).

We consider two cases:

Case 1: $q \leq \tau(\nu) - 1$. That is by Remark 5.32, $\theta = q \cdot 2^{-\nu} \in (0, 1-\omega)$. In this case, by Corollary 5.29 and (5.21),

$$1 > \{\!\{2x + \widetilde{\omega}\}\!\} = 2x + \widetilde{\omega} = 2\left(q \cdot 2^{-\nu}\right) + \widetilde{\omega} = \frac{2^{\nu^{\mathsf{d}}} q + \varpi + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^\star}} =$$

$$\frac{1}{2^{\nu-1}}\left(\frac{2^{\nu^{\mathsf{d}}}\left(q + \left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor\right)}{2^{\nu^{\mathsf{d}}}} + \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right) =$$

$$\frac{1}{2^{\nu-1}}\left(q + \left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor + \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right).$$

Consequently, since $\{\!\{2x + \widetilde{\omega}\}\!\}, \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}, \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}} \in [0,1)$, by Remark 5.31 for $\alpha = q + \left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor + \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}} < 2^{\nu-1}$,

$$\mathsf{dyad}\left(\{\!\{2x + \widetilde{\omega}\}\!\}\right) =$$

$$\mathsf{dyad}\left(\frac{1}{2^{\nu-1}}\left(q + \left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor + \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right)\right) =$$

$$\mathbf{0}^{\nu-1-\mathsf{lbq}}\,\mathsf{binary}\left(q + \left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor\right)\mathsf{dyad}\left(\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right),$$

where $\mathsf{lbq} = \mathsf{length}\left(\mathsf{binary}\left(q + \left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor\right)\right)$. Thus, by (5.22), for $x = q \cdot 2^{-\nu} \in (0, 1-\omega)$,

$$\left(\lim_{n\to\infty}\prod_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}(\{\!\{2x+\widetilde{\omega}\}\!\},i)}\right)\vec{\mathbf{1}} =$$

$$\mathbf{M}_0^{\nu-1-\mathsf{lbq}} \cdot \mathbf{M}_1 \cdot \mathbf{M}_{a_{\mathsf{lbq}-2}} \cdot \mathbf{M}_{a_{\mathsf{lbq}-3}} \cdots \mathbf{M}_{a_1} \cdot \mathbf{M}_{a_0} \cdot$$

$$\left(\lim_{n\to\infty}\prod_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}\left(\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}},i\right)}\right)\vec{\mathbf{1}} =$$

$$\mathbf{M}_0^{\nu-1-\mathsf{lbq}} \cdot \mathbf{M}_1 \cdot \mathbf{M}_{a_{\mathsf{lbq}-2}} \cdot \mathbf{M}_{a_{\mathsf{lbq}-3}} \cdots \mathbf{M}_{a_1} \cdot \mathbf{M}_{a_0} \cdot \vec{\mathbf{V}}_{\nu^{\mathsf{d}}}^{(\widetilde{\omega})},$$

where $\mathsf{binary}\left(q + \left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor\right) = 1a_{\mathsf{lbq}-2}a_{\mathsf{lbq}-3}\cdots a_1 a_0$.

Then, the statement for the case $1 \leq q \leq \tau(\nu) - 1$ follows from Corollary 5.29 for the case $x = q \cdot 2^{-\nu} \in (0, 1-\omega)$.

Case 2: $\tau(\nu) \leq q < 2^{\nu-1}$. That is $x = q \cdot 2^{-\nu} \in \left(1-\omega, \frac{1}{2}\right)$. By Corollary 5.29 and (5.21),

$$[0,1) \ni \{\!\{2x + \widetilde{\omega}\}\!\} = 2x + \widetilde{\omega} - 1 = \frac{2^{\nu^{\mathsf{d}}} q + \varpi - 2^{\nu^\star} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^\star}} =$$

$$\frac{1}{2^{\nu-1}}\left(q + \left\lfloor \frac{\varpi}{2^{\nu^{\mathsf{d}}}} \right\rfloor - 2^{\nu-1} + \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right) =$$

$$\frac{1}{2^{\nu-1}}\left((q - \tau(\nu)) + \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right).$$

Since $q = \tau(\nu)$ (i.e. $x = \tau(\nu) \cdot 2^{-\nu}$), by Remark 5.31 we have

$$\mathsf{dyad}\big(2x + \widetilde{\omega} - 1\big) = \mathsf{dyad}\left(\frac{1}{2^{\nu-1}} \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^{\star}}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right) =$$

$$\mathsf{binary}_{\nu-1}(0)\,\mathsf{dyad}\left(\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^{\star}}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right) = \mathbf{0}^{\nu-1}\,\mathsf{dyad}\left(\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^{\star}}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right).$$

Hence, by (5.22),

$$\left(\lim_{n\to\infty} \prod\nolimits_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}(2x+\widetilde{\omega}-1,i)}\right)\vec{\mathbf{1}} =$$

$$\mathbf{M}_0^{\nu-1} \cdot \left(\lim_{n\to\infty} \prod\nolimits_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}\left(\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^{\star}}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}},i\right)}\right)\vec{\mathbf{1}} = \mathbf{M}_0^{\nu-1} \cdot \vec{\mathbf{V}}_{\nu^{\mathsf{d}}}^{(\widetilde{\omega})},$$

and the second statement of the corollary follows from Corollary 5.29 for the case $x = q \cdot 2^{-\nu} \in \big(1-\omega, \frac{1}{2}\big)$.

When $q > \tau(\nu)$, again by Remark 5.31,

$$\mathsf{dyad}\big(2x + \widetilde{\omega} - 1\big) = \mathsf{dyad}\left(\frac{1}{2^{\nu-1}}\left((q-\tau(\nu)) + \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^{\star}}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right)\right) =$$

$$\mathsf{binary}_{\nu-1}(q-\tau(\nu))\,\mathsf{dyad}\left(\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^{\star}}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right) =$$

$$\mathbf{0}^{\nu-1-\mathsf{lbq}}\,\mathsf{binary}(q-\tau(\nu))\,\mathsf{dyad}\left(\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^{\star}}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right).$$

Thus, again by (5.22),

$$\left(\lim_{n\to\infty} \prod\nolimits_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}(\{\!\{2x+\widetilde{\omega}\}\!\},i)}\right)\vec{\mathbf{1}} =$$

$$\mathbf{M}_0^{\nu-1-\mathsf{lbq}} \cdot \mathbf{M}_1 \cdot \mathbf{M}_{a_{\mathsf{lbq}-2}} \cdot \mathbf{M}_{a_{\mathsf{lbq}-3}} \cdots \mathbf{M}_{a_1} \cdot \mathbf{M}_{a_0} \cdot$$

$$\left(\lim_{n\to\infty} \prod\nolimits_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}\left(\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^{\star}}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}},i\right)}\right)\vec{\mathbf{1}} =$$

$$\mathbf{M}_0^{\nu-1-\mathsf{lbq}} \cdot \mathbf{M}_1 \cdot \mathbf{M}_{a_{\mathsf{lbq}-2}} \cdot \mathbf{M}_{a_{\mathsf{lbq}-3}} \cdots \mathbf{M}_{a_1} \cdot \mathbf{M}_{a_0} \cdot \vec{\mathbf{V}}_{\nu^{\mathsf{d}}}^{(\widetilde{\omega})},$$

where $\mathsf{binary}(q-\tau(\nu)) = 1a_{\mathsf{lbq}-2}a_{\mathsf{lbq}-3}\cdots a_1 a_0$ has length lbq.

This shows the statement of the corollary for the case $\tau(\nu) < q < 2^{\nu-1}$, by using again Corollary 5.29 for the case $x = q \cdot 2^{-\nu} \in \big(1-\omega, \frac{1}{2}\big)$. □

5.5.2. Computation of the Rotated $\psi_{j,n}^{\mathrm{PER}}$ with $j < 0$. In this subection we will finally tackle the computation of $\psi_{j,n}^{\mathrm{PER}}$ on rotated dyadic points. It serves as a worthy end to the chapter, as it is by far the most complex computation of all. At the end of the day, it all boils down to using the techniques developed in Subsections 5.4.2 and 5.5.1. However, when combining both techniques the results become quite more involved. If one recalls Corollary 5.20, even though we wanted to compute the wavelet value for $\theta \in \big[0, \frac{1}{2^j}\big)$, the interval had to be divided into two, $\theta \in \big[0, \frac{1}{2^{j+1}}\big)$ and $\theta \in \big[\frac{1}{2^{j+1}}, \frac{1}{2^j}\big)$. On the other hand, we have just seen that the fact that we are applying an irrational rotation with $\omega > \frac{1}{2}$, also divides the interval of computable numbers in two, separated by $1-\omega$. Hence, in this case we will find that the number of cases will be three, stemming from the combination of the previous two. One cannot say which gives rise to which, but both effects need to be taken into account simultaneously.

As we have already stated, we are interested in finding an efficient implementation of the Daubechies-Lagarias algorithm for

$$\psi_{j,n}^{\mathrm{PER}}\big(R_\omega(\theta)\big) \quad \text{with} \quad j \geq 1,\ n \in \{0, 1, \ldots, 2^j - 1\}$$

when $\theta \in [0, 1)$ is dyadic.

From the definition of periodized wavelet in Equation (3.18) applied in our case we can write:

$$\psi_{j,n}^{\mathrm{PER}}\big(R_\omega(\theta)\big) = \sum_{\ell \in \mathbb{Z}} \psi_{j,n}\big(\theta + \omega + \ell\big) = 2^{-j/2} \sum_{\ell \in \mathbb{Z}} \psi\left(\frac{\big(\theta + \omega + \ell\big) - 2^j n}{2^j}\right) \tag{5.24}$$

for every $\theta \in [0, 1)$, and $j, n \in \mathbb{Z}$.

In a very similar manner to Proposition 3.22 we can find self-similarities within the wavelet values with the same j.

Corollary 5.35 (of Proposition 3.22). *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments. Then, for every $\theta \in \big[0, \frac{1}{2^j}\big)$, $j \in \{1, \ldots, J\}$, and $n \in \{0, 1, \ldots, 2^j - 1\}$,*

$$\psi_{-j,n}^{\mathrm{PER}}\big(R_\omega(\theta)\big) = \psi_{-j,\mathsf{mod}\left(n+k,\,2^j\right)}^{\mathrm{PER}}\left(R_\omega\left(\theta + \tfrac{k}{2^j}\right)\right)$$

for every $k = 1, 2, \ldots, 2^j - 1$.

Now we need a rotated version of Corollary 5.16 to get an explicit expression for $\psi_{-j,n}^{\mathrm{PER}}$ for general values of j, n and $\theta \in \mathbb{S}^1$, in terms of these parameters and variable. This is also the analogue of Corollary 5.20 for non-dyadic values of x. To do this we will introduce some useful notation.

Recall that $\widetilde{\omega} := \{\!\{2\omega\}\!\}$, and $2\omega = 1 + \widetilde{\omega}$. Now we set,

$$\begin{aligned} {}^{j}\omega &:= 2^j \cdot 2 \cdot \omega = 2^j + 2^j\widetilde{\omega} \text{ and} \\ {}^{j}\varpi &:= \big\lfloor {}^{j}\omega \big\rfloor = 2^j + \big\lfloor 2^j\widetilde{\omega} \big\rfloor . \end{aligned}$$

Clearly, with this notation, $\{\!\{{}^{j}\omega\}\!\}= \{\!\{2^j\widetilde{\omega}\}\!\}$, and

$${}^{j}\omega = {}^{j}\varpi + \{\!\{{}^{j}\omega\}\!\}= 2^j + \big\lfloor 2^j\widetilde{\omega} \big\rfloor + \{\!\{2^j\widetilde{\omega}\}\!\}.$$

Note that ${}^{j}\varpi$ corresponds to a shift to the left of length $\nu^\star - j$ of the binary expression of ϖ.

On the other hand, given an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments and $j \in \{1, \ldots, J\}$, we denote

$$\kappa_j^\omega := (-1)^{\left\lfloor \frac{\varpi}{2^{\nu^\star - j}} \right\rfloor} \frac{\sqrt{2}^{j+1}}{2p - 1}.$$

Also, for every $\theta \in \big[0, \frac{1}{2^j}\big)$, we denote

$$\eta_j(\theta) := \big\lfloor 2^{j+1}\theta + \{\!\{2^j\widetilde{\omega}\}\!\} \big\rfloor .$$

Note that $0 \leq 2^{j+1}\theta + \{\!\{2^j\widetilde{\omega}\}\!\} < 3$, hence $\eta_j(\theta) \in \{0, 1, 2\}$. The avid reader will have already noticed that η_j corresponds to the divisions of the interval $\big[0, \frac{1}{2^{j+1}}\big)$ mentioned in the introduction of this sub-section.

Finally, we will define the analogous to the vector $\overrightarrow{\mathbf{uj}}(m)$ for the rotated case

$$\overrightarrow{\mathbf{uj}}^{\boldsymbol{\omega}}(m) := \sum_{\ell \in \mathbb{Z}} u\Big(\tfrac{{}^{j}\varpi}{2} + 2^j\ell - m\Big)$$

COROLLARY 5.36. *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments, and let $\nu \in \mathbb{N} \setminus \{1\}$, $j \in \{1, \dots, \nu - 1\}$ and $n \in \{0, 1, \dots, 2^j - 1\}$. Then, for every $\theta \in \big[0, \frac{1}{2^j}\big)$,*

$$\psi^{\text{PER}}_{-j,n}\big(R_\omega(\theta)\big) = (-1)^{\eta_j(x)} \kappa^\omega_j$$

$$\vec{\boldsymbol{uj}}^{\boldsymbol{\omega}}\big(n - \eta_j(\theta)\tfrac{1}{2}\big)^\intercal \left(\lim_{n\to\infty} \prod\nolimits_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}\left(2^{j+1}\theta + \{\{2^j\widetilde{\omega}\}\} - \eta_j(\theta), i\right)}\right) \vec{\mathbf{1}}.$$

PROOF. We recall that for every $\theta \in \mathbb{R}$ and $k \in \mathbb{Z}$ we have

$$\{\{2(\theta + k)\}\} = \{\{2\theta\}\} \quad \text{and} \quad d(\theta + k) = 1 - \lfloor 2(\theta + k) \rfloor = d(\theta) - 2k.$$

Hence, $(-1)^{d(\theta+k)} = (-1)^{d(\theta)}$. So, by Equation (3.18) and Corollary 5.6,

$$\begin{aligned}
\psi^{\text{PER}}_{-j,n}\big(R_\omega(\theta)\big) &= \sum_{\ell\in\mathbb{Z}} \psi_{-j,n}\big(\theta + \omega + \ell\big) = 2^{j/2} \sum_{\ell\in\mathbb{Z}} \psi\left(\frac{\big(\theta + \omega + \ell\big) - 2^{-j}n}{2^{-j}}\right) \\
&= 2^{j/2} \sum_{\ell\in\mathbb{Z}} \psi\big(2^j(\theta + \omega) + (2^j\ell - n)\big) \\
&= (-1)^{d(2^j(\theta+\omega))} \frac{\sqrt{2}^{j+1}}{2p - 1} \left(\sum_{\ell\in\mathbb{Z}} u\big(2^j(\theta + \omega) + (2^j\ell - n)\big)\right)^\intercal \\
&\qquad \left(\lim_{n\to\infty} \prod\nolimits_{i=1}^{n} \mathbf{M}_{\mathsf{dyad}(\{\{2^{j+1}(\theta+\omega)\}\}, i)}\right) \vec{\mathbf{1}}.
\end{aligned}$$

Now observe that

$$2^{j+1}(\theta + \omega) = 2^{j+1}\theta + {}^j\omega = 2^{j+1}\theta + \{\{2^j\widetilde{\omega}\}\} + \big(2^j + \lfloor 2^j\widetilde{\omega} \rfloor\big). \tag{5.25}$$

Consequently,

$$\big\{\big\{2^{j+1}(\theta + \omega)\big\}\big\} = \big\{\big\{2^{j+1}\theta + \{\{2^j\widetilde{\omega}\}\}\big\}\big\} = 2^{j+1}\theta + \{\{2^j\widetilde{\omega}\}\} - \eta_j(\theta).$$

Now we claim that

$$\big\lfloor 2^{j+1}(\theta + \omega) \big\rfloor = \eta_j(\theta) + \ 2^j + \big\lfloor \tfrac{\varpi}{2^{\nu^\star - j}} \big\rfloor.$$

Observe that, if the claim holds,

$$\begin{aligned}
(-1)^{d(2^j(\theta+\omega))} &= (-1)^{1 - \lfloor 2^{j+1}(\theta+\omega) \rfloor} = -(-1)^{\lfloor 2^{j+1}(\theta+\omega) \rfloor} = \\
&\quad - (-1)^{\eta_j(\theta)} (-1)^{2^j} (-1)^{\left\lfloor \frac{\varpi}{2^{\nu^\star - j}} \right\rfloor} = (-1)^{\eta_j(\theta)} \left(-(-1)^{\left\lfloor \frac{\varpi}{2^{\nu^\star - j}} \right\rfloor}\right),
\end{aligned}$$

so that $(-1)^{d(2^j(\theta+\omega))} \frac{\sqrt{2}^{j+1}}{2p-1} = -(-1)^{\eta_j(\theta)} \kappa^\omega_j$.

By (5.25), $\big\lfloor 2^{j+1}(\theta + \omega) \big\rfloor = \eta_j(\theta) + \ 2^j + \big\lfloor 2^j\widetilde{\omega} \big\rfloor$. So, to prove the claim we have to show that

$$\big\lfloor 2^j\widetilde{\omega} \big\rfloor = \left\lfloor \frac{\varpi}{2^{\nu^\star - j}} \right\rfloor.$$

By (5.21),

$$\begin{aligned}
2^j\widetilde{\omega} = \frac{2^{\nu^\star}\widetilde{\omega}}{2^{\nu^\star - j}} = \frac{\varpi + \big\{\big\{2^{\nu^\star}\widetilde{\omega}\big\}\big\}}{2^{\nu^\star - j}} = \frac{2^{\nu^\star - j} \big\lfloor \frac{\varpi}{2^{\nu^\star - j}} \big\rfloor + \widetilde{\varpi}^{\nu^\star - j} + \big\{\big\{2^{\nu^\star}\widetilde{\omega}\big\}\big\}}{2^{\nu^\star - j}} = \\
\left\lfloor \frac{\varpi}{2^{\nu^\star - j}} \right\rfloor + \frac{\widetilde{\varpi}^{\nu^\star - j} + \big\{\big\{2^{\nu^\star}\widetilde{\omega}\big\}\big\}}{2^{\nu^\star - j}}.
\end{aligned}$$

Hence,

$$\{\{2^j\widetilde{\omega}\}\} = \left\{\!\!\left\{ \frac{\widetilde{\varpi}^{\nu^\star - j} + \{\{2^{\nu^\star}\widetilde{\omega}\}\}}{2^{\nu^\star - j}} \right\}\!\!\right\} = \frac{\widetilde{\varpi}^{\nu^\star - j} + \{\{2^{\nu^\star}\widetilde{\omega}\}\}}{2^{\nu^\star - j}}, \text{ and}$$
$$\lfloor 2^j\widetilde{\omega}\rfloor = \left\lfloor \frac{\varpi}{2^{\nu^\star - j}} \right\rfloor,$$

because $\widetilde{\varpi}^{\nu^\star - j} < 2^{\nu^\star - j}$ (in particular, $\frac{\widetilde{\varpi}^{\nu^\star - j} + \{\{2^{\nu^\star}\widetilde{\omega}\}\}}{2^{\nu^\star - j}} < 1$). This ends the proof of the claim.

To end the proof of the corollary we have to show that, for $\theta \in \big[0, \frac{1}{2^j}\big)$,

$$\sum_{\ell \in \mathbb{Z}} u\big(2^j(\theta + \omega) + (2^j\ell - n)\big) = \overrightarrow{\mathbf{uj}}^{\boldsymbol{\omega}}\big(n - \eta_j(\theta)\tfrac{1}{2}\big) = \sum_{\ell \in \mathbb{Z}} u\Big(\tfrac{{}^j\varpi + \eta_j(\theta)}{2} + (2^j\ell - n)\Big).$$

Recall that $u_i(\theta) := (-1)^i h[i + d(\theta)]$ denotes the i-th component of the vector $u(\theta)$ with $i \in \{0, 1, \ldots, 2p-2\}$. So, we need to prove that

$$d\big(2^j(\theta + \omega) + (2^j\ell - n)\big) = d\Big(\tfrac{{}^j\varpi + \eta_j(\theta)}{2} + (2^j\ell - n)\Big).$$

By using again that $d(\theta + k) = 1 - \lfloor 2(\theta + k)\rfloor = 1 - \lfloor 2\theta\rfloor - 2k$ for every $\theta \in \mathbb{R}$ and $k \in \mathbb{Z}$, the last equality is equivalent to

$$\big\lfloor 2^{j+1}(\theta + \omega)\big\rfloor = \big\lfloor {}^j\varpi + \eta_j(\theta)\big\rfloor = {}^j\varpi + \eta_j(\theta).$$

On the other hand, for $x \in \big[0, \frac{1}{2^j}\big)$, we have

$$\big\lfloor 2^{j+1}(\theta + \omega)\big\rfloor = \big\lfloor 2^{j+1}\theta + {}^j\varpi + \{\{{}^j\omega\}\}\big\rfloor = {}^j\varpi + \eta_j(\theta).$$

This ends the proof of the corollary. □

Now we need the dyadic version of Corollary 5.36.

For $\nu \in \mathbb{N} \setminus \{1\}$, $j \in \{1, \ldots, \nu - 1\}$ and $\ell \in \{0, 1\}$ we denote

$$\tau_\ell(j, \nu) := 2^{\nu - j - (1-\ell)} - \left\lfloor \frac{\widetilde{\varpi}^{\nu^\star - j}}{2^{\nu^{\mathrm{d}}}} \right\rfloor \in \mathbb{Z}.$$

Remark 5.37 (The dyadic version of $\eta_j(x)$). We would like to obtain an analogous result to 5.32 but for the $\psi^{\mathrm{PER}}_{j,n}$ case. Let $\nu \in \mathbb{N} \setminus \{1\}$ and $j \in \{1, \ldots, \nu - 1\}$. From the proof of Corollary 5.36 we know that

$$\{\{2^j\widetilde{\omega}\}\}= \frac{\widetilde{\varpi}^{\nu^\star - j} + \{\{2^{\nu^\star}\widetilde{\omega}\}\}}{2^{\nu^\star - j}}.$$

Hence,

$$\begin{aligned}\tau_\ell(j, \nu) - 1 = 2^{\nu - j - (1-\ell)} - \left\lfloor \frac{\widetilde{\varpi}^{\nu^\star - j}}{2^{\nu^{\mathrm{d}}}} \right\rfloor - 1 <& \\ 2^{\nu - j - (1-\ell)} - \frac{\widetilde{\varpi}^{\nu^\star - j} + \{\{2^{\nu^\star}\widetilde{\omega}\}\}}{2^{\nu^{\mathrm{d}}}} =& \\ 2^{\nu - j - (1-\ell)} - 2^{\nu - j - 1}\left(\frac{\widetilde{\varpi}^{\nu^\star - j} + \{\{2^{\nu^\star}\widetilde{\omega}\}\}}{2^{\nu^\star - j}}\right) =& \\ 2^{\nu - j - 1}\big(2^\ell - \{\{2^j\widetilde{\omega}\}\}\big) < 2^{\nu - j - (1-\ell)} - \frac{\widetilde{\varpi}^{\nu^\star - j}}{2^{\nu^{\mathrm{d}}}} \le \tau_\ell(j, \nu).&\end{aligned}$$

Consequently, for every $\theta = \frac{q}{2^\nu} \in \big(0, \frac{1}{2^j}\big)$,

$$\eta_j\big(\tfrac{q}{2^\nu}\big) = \big\lfloor 2^{j+1}\tfrac{q}{2^\nu} + \{\{2^j\widetilde{\omega}\}\}\big\rfloor = \big\lfloor \tfrac{q}{2^{\nu - j - 1}} + \{\{2^j\widetilde{\omega}\}\}\big\rfloor = 1$$

is equivalent to $2^{\nu-j-1}\big(1-\{\!\{2^j\widetilde{\omega}\}\!\}\big) \leq q < 2^{\nu-j-1}\big(2-\{\!\{2^j\widetilde{\omega}\}\!\}\big)$ which, in turn, is equivalent to

$$\tau_0(j,\nu) \leq q \leq \tau_1(j,\nu)-1.$$

Analogously,

$$\eta_j\big(\tfrac{q}{2^\nu}\big)=0 \quad \text{is equivalent to} \quad q \leq \tau_0(j,\nu)-1,$$

and

$$\eta_j\big(\tfrac{q}{2^\nu}\big)=2 \quad \text{is equivalent to} \quad q \geq \tau_1(j,\nu).$$

∎

Corollary 5.38 (Corollary 5.36 Dyadic version). *Let ψ be an $\mathbb{R}$-Daubechies wavelet with $p \geq 1$ vanishing moments, let $\nu \in \mathbb{N}\setminus\{1\}$, $j \in \{1,\dots,\nu-1\}$ and $n \in \{0,1,\dots,2^j-1\}$. Then, the following statements hold:*

(a)
$$\psi^{\text{PER}}_{-j,n}\big(R_\omega(0)\big)=\kappa^\omega_j \ \vec{\boldsymbol{uj}}^{\boldsymbol{\omega}}(n)^{\mathsf{T}}\vec{\mathbf{V}}^{(\widetilde{\omega})}_{\nu^\star-j},$$
and for every $q \in \big\{1,2,\dots,\tau_0(j,\nu)-1\big\}$,

$$\psi^{\text{PER}}_{-j,n}\big(R_\omega\big(\tfrac{q}{2^\nu}\big)\big)=\kappa^\omega_j$$
$$\vec{\boldsymbol{uj}}^{\boldsymbol{\omega}}(n)^{\mathsf{T}}\left(\mathbf{M}_0^{\nu-j-1-\mathsf{lbq}}\cdot\mathbf{M}_1\cdot\mathbf{M}_{a_{\mathsf{lbq}-2}}\cdot\mathbf{M}_{a_{\mathsf{lbq}-3}}\cdots\mathbf{M}_{a_1}\cdot\mathbf{M}_{a_0}\cdot\vec{\mathbf{V}}^{(\widetilde{\omega})}_{\nu^{\mathsf{d}}}\right),$$

where $\mathsf{binary}\left(q+\big\lfloor\frac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}}\big\rfloor\right)=1a_{\mathsf{lbq}-2}a_{\mathsf{lbq}-3}\cdots a_1a_0.$

(b)
$$\psi^{\text{PER}}_{-j,n}\big(R_\omega\big(\tfrac{\tau_0(j,\nu)}{2^\nu}\big)\big)=-\kappa^\omega_j \ \vec{\boldsymbol{uj}}^{\boldsymbol{\omega}}\big(n-\tfrac{1}{2}\big)^{\mathsf{T}}\left(\mathbf{M}_0^{\nu-j-1}\cdot\vec{\mathbf{V}}^{(\widetilde{\omega})}_{\nu^{\mathsf{d}}}\right),$$
and for every $q \in \{\tau_0(j,\nu)+1,\tau_0(j,\nu)+2,\dots,\tau_1(j,\nu)-1\}$,

$$\psi^{\text{PER}}_{-j,n}\big(R_\omega\big(\tfrac{q}{2^\nu}\big)\big)=-\kappa^\omega_j$$
$$\vec{\boldsymbol{uj}}^{\boldsymbol{\omega}}\big(n-\tfrac{1}{2}\big)^{\mathsf{T}}\left(\mathbf{M}_0^{\nu-j-1-\mathsf{lbq}}\cdot\mathbf{M}_1\cdot\mathbf{M}_{a_{\mathsf{lbq}-2}}\cdot\mathbf{M}_{a_{\mathsf{lbq}-3}}\cdots\mathbf{M}_{a_1}\cdot\mathbf{M}_{a_0}\cdot\vec{\mathbf{V}}^{(\widetilde{\omega})}_{\nu^{\mathsf{d}}}\right),$$

where $\mathsf{binary}\big(q-\tau_0(j,\nu)\big)=1a_{\mathsf{lbq}-2}a_{\mathsf{lbq}-3}\cdots a_1a_0.$

(c)
$$\psi^{\text{PER}}_{-j,n}\big(R_\omega\big(\tfrac{\tau_1(j,\nu)}{2^\nu}\big)\big)=\kappa^\omega_j \ \vec{\boldsymbol{uj}}^{\boldsymbol{\omega}}(n-1)^{\mathsf{T}}\left(\mathbf{M}_0^{\nu-j-1}\cdot\vec{\mathbf{V}}^{(\widetilde{\omega})}_{\nu^{\mathsf{d}}}\right),$$
and for every $q \in \{\tau_1(j,\nu)+1,\tau_1(j,\nu)+2,\dots,2^{\nu-j}-1\}$,

$$\psi^{\text{PER}}_{-j,n}\big(R_\omega\big(\tfrac{q}{2^\nu}\big)\big)=\kappa^\omega_j$$
$$\vec{\boldsymbol{uj}}^{\boldsymbol{\omega}}(n-1)^{\mathsf{T}}\left(\mathbf{M}_0^{\nu-j-1-\mathsf{lbq}}\cdot\mathbf{M}_1\cdot\mathbf{M}_{a_{\mathsf{lbq}-2}}\cdot\mathbf{M}_{a_{\mathsf{lbq}-3}}\cdots\mathbf{M}_{a_1}\cdot\mathbf{M}_{a_0}\cdot\vec{\mathbf{V}}^{(\widetilde{\omega})}_{\nu^{\mathsf{d}}}\right),$$

where $\mathsf{binary}\big(q-\tau_1(j,\nu)\big)=1a_{\mathsf{lbq}-2}a_{\mathsf{lbq}-3}\cdots a_1a_0.$

Remark 5.39. For every $q \leq \tau_0(j,\nu)-1$, we have

$$q+\Big\lfloor\frac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}}\Big\rfloor < \tau_0(j,\nu)+\Big\lfloor\frac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}}\Big\rfloor = 2^{\nu-j-1}-\Big\lfloor\frac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}}\Big\rfloor+\Big\lfloor\frac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}}\Big\rfloor < 2^{\nu^\star-2}$$

because $\nu^\star > \nu$ and $j \geq 1$. Consequently, $q+\big\lfloor\frac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}}\big\rfloor$ fits in a $\nu^\star-$bit unsigned integer.

Now let $\tau_0(j,\nu) \leq q \leq \tau_1(j,\nu)-1$. In this case we have

$$q-\tau_0(j,\nu) < \tau_1(j,\nu)-\tau_0(j,\nu) = 2^{\nu-j}-2^{\nu-j-1}=2^{\nu-j-1}<2^{\nu^\star-2},$$

and again $q-\tau_0(j,\nu)$ fits in a $\nu^\star-$bit unsigned integer.

Finally, when $\tau_1(j,\nu) \le q \le 2^{\nu-j}-1$,

$$q-\tau_1(j,\nu) < 2^{\nu-j}-\tau_1(j,\nu) = \left\lfloor \frac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}} \right\rfloor \le \frac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}} \le \\ \varpi = \lfloor 2^{\nu^\star}\widetilde{\omega} \rfloor < 2^{\nu^\star}\widetilde{\omega} < 2^{\nu^\star}\cdot\frac{1}{4} = 2^{\nu^\star-2}.$$

So, $q-\tau_1(j,\nu)\big)$ also fits in a $\nu^\star-$bit unsigned integer. ∎

As before, to obtain implementable and efficient formulae for $\psi^{\mathrm{PER}}_{-j,n}\big(R_\omega(\theta)\big)$, we additionally need explicit expressions for the vectors $\overrightarrow{\mathsf{uj}}^{\omega}(n)$, $\overrightarrow{\mathsf{uj}}^{\omega}\big(n-\frac{1}{2}\big)$, and $\overrightarrow{\mathsf{uj}}^{\omega}(n{-}1)$. This will be done in the next subsection, after the proof of Corollary 5.36.

Proof of Corollary 5.38. From the proof of Corollary 5.36we know that

$$\{\!\{2^j\widetilde{\omega}\}\!\}= \frac{\widetilde{\varpi}^{\nu^\star-j}+\{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^\star-j}}.$$

Hence, since $\eta_j(0) = \big\lfloor \{\!\{2^j\widetilde{\omega}\}\!\} \big\rfloor = 0$, the first statement of (a) follows from Corollary 5.36 and (5.22).

Now we prove the second statement of (a). By Remark 5.37, we know that $q \in \big\{1,2,\dots,\tau_0(j,\nu)-1\big\}$ is equivalent to $\eta_j\big(\frac{q}{2^\nu}\big) = \big\lfloor \frac{q}{2^{\nu-j-1}} + \{\!\{2^j\widetilde{\omega}\}\!\} \big\rfloor = 0$. On the other hand, by (5.21),

$$1 > 2^{j+1}\theta + \{\!\{2^j\widetilde{\omega}\}\!\} = \frac{q}{2^{\nu-j-1}} + \{\!\{2^j\widetilde{\omega}\}\!\} = \\ \frac{q}{2^{\nu-j-1}} + \frac{\widetilde{\varpi}^{\nu^\star-j}+\{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^\star-j}} = \\ \frac{1}{2^{\nu-j-1}}\left(q + \frac{2^{\nu^{\mathsf{d}}}\big\lfloor\frac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}}\big\rfloor + \widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}} \right) = \\ \frac{1}{2^{\nu-j-1}}\left(q + \big\lfloor\tfrac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}}\big\rfloor + \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}} \right).$$

In particular,

$$q + \big\lfloor\tfrac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}}\big\rfloor + \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}} < 2^{\nu-j-1}.$$

So, since $\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}}+\{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}} \in [0,1)$, by Remark 5.31 for $q + \big\lfloor\frac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}}\big\rfloor + \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}}+\{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}$,

$$\mathsf{dyad}\left(\frac{q}{2^{\nu-j-1}} + \{\!\{2^j\widetilde{\omega}\}\!\}\right) = \\ \mathsf{dyad}\left(\frac{1}{2^{\nu-j-1}}\left(q + \big\lfloor\tfrac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}}\big\rfloor + \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}} \right)\right) = \\ \mathbf{0}^{\nu-j-1-\mathsf{lbq}}\,\mathsf{binary}\left(q + \big\lfloor\tfrac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}}\big\rfloor\right)\mathsf{dyad}\left(\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right),$$

where $\mathsf{lbq} = \mathsf{length}\left(\mathsf{binary}\left(q + \lfloor\frac{\widetilde{\varpi}^{\nu^\star - j}}{2^{\nu^{\mathsf{d}}}}\rfloor\right)\right)$. Thus, by (5.22),

$$\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}\left(\frac{q}{2^{\nu-j-1}}+\{\!\{2^j\widetilde{\omega}\}\!\},i\right)}\right)\vec{\mathbf{1}} =$$

$$\mathbf{M}_0^{\nu-j-1-\mathsf{lbq}}\cdot\mathbf{M}_1\cdot\mathbf{M}_{a_{\mathsf{lbq}-2}}\cdot\mathbf{M}_{a_{\mathsf{lbq}-3}}\cdots\mathbf{M}_{a_1}\cdot\mathbf{M}_{a_0}\cdot$$

$$\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}\left(\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}}+\{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right),i)}\right)\vec{\mathbf{1}} =$$

$$\mathbf{M}_0^{\nu-j-1-\mathsf{lbq}}\cdot\mathbf{M}_1\cdot\mathbf{M}_{a_{\mathsf{lbq}-2}}\cdot\mathbf{M}_{a_{\mathsf{lbq}-3}}\cdots\mathbf{M}_{a_1}\cdot\mathbf{M}_{a_0}\cdot\vec{\mathbf{V}}_{\nu^{\mathsf{d}}}^{(\widetilde{\omega})},$$

where $\mathsf{binary}\left(q + \lfloor\frac{\widetilde{\varpi}^{\nu^\star - j}}{2^{\nu^{\mathsf{d}}}}\rfloor\right) = 1a_{\mathsf{lbq}-2}a_{\mathsf{lbq}-3}\cdots a_1 a_0$. So, the second statement of (a) follows from by Corollary 5.36.

Now we prove (b). By Remark 5.37, we know that $\tau_0(j,\nu) \le q \le \tau_1(j,\nu) - 1$ is equivalent to $\eta_j\left(\frac{q}{2^\nu}\right) = \left\lfloor\frac{q}{2^{\nu-j-1}} + \{\!\{2^j\widetilde{\omega}\}\!\}\right\rfloor = 1$. So, since

$$j \le \nu - 1 \quad\Longleftrightarrow\quad \nu^{\mathsf{d}} \le \nu^\star - j,$$

by (5.21) we get

$$1 > 2^{j+1}x + \{\!\{2^j\widetilde{\omega}\}\!\} - 1 = \frac{q}{2^{\nu-j-1}} + \{\!\{2^j\widetilde{\omega}\}\!\} - 1 =$$

$$\frac{q - 2^{\nu-j-1}}{2^{\nu-j-1}} + \frac{\widetilde{\varpi}^{\nu^\star-j} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^\star-j}} =$$

$$\frac{q - \tau_0(j,\nu)}{2^{\nu-j-1}} + \frac{\widetilde{\varpi}^{\nu^\star-j} - 2^{\nu^{\mathsf{d}}}\left\lfloor\frac{\widetilde{\varpi}^{\nu^\star-j}}{2^{\nu^{\mathsf{d}}}}\right\rfloor + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu-j-1}\cdot 2^{\nu^{\mathsf{d}}}} =$$

$$\frac{1}{2^{\nu-j-1}}\left(q - \tau_0(j,\nu) + \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right).$$

When $q = \tau_0(j,\nu)$, by Remark 5.31 we have

$$\mathsf{dyad}\left(\frac{q}{2^{\nu-j-1}} + \{\!\{2^j\widetilde{\omega}\}\!\} - 1\right) = \mathsf{dyad}\left(\frac{1}{2^{\nu-j-1}}\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right) =$$

$$\mathbf{0}^{\nu-j-1}\,\mathsf{dyad}\left(\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right).$$

Hence, by (5.22),

$$\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}\left(\frac{q}{2^{\nu-j-1}}+\{\!\{2^j\widetilde{\omega}\}\!\}-1,i\right)}\right)\vec{\mathbf{1}} =$$

$$\mathbf{M}_0^{\nu-j-1}\cdot\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}\left(\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}}+\{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}},i\right)}\right)\vec{\mathbf{1}} = \mathbf{M}_0^{\nu-j-1}\cdot\vec{\mathbf{V}}_{\nu^{\mathsf{d}}}^{(\widetilde{\omega})},$$

and the first statement of (b) follows from Corollary 5.36.

When $q > \tau_0(j,\nu)$, again by Remark 5.31,

$$\mathsf{dyad}\left(\frac{q}{2^{\nu-j-1}} + \{\!\{2^j\widetilde{\omega}\}\!\} - 1\right) =$$

$$\mathsf{dyad}\left(\frac{1}{2^{\nu-j-1}}\left(q - \tau_0(j,\nu) + \frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right)\right) =$$

$$\mathbf{0}^{\nu-j-1-\mathsf{lbq}}\,\mathsf{binary}\big(q - \tau_0(j,\nu)\big)\,\mathsf{dyad}\left(\frac{\widetilde{\varpi}^{\nu^{\mathsf{d}}} + \{\!\{2^{\nu^\star}\widetilde{\omega}\}\!\}}{2^{\nu^{\mathsf{d}}}}\right)$$

and, again by (5.22),

$$\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}\left(\frac{q}{2^{\nu-j-1}}+\{\{2^j\widetilde{\omega}\}\}\,-1,i\right)}\right)\vec{\mathbf{1}} =$$
$$\mathbf{M}_0^{\nu-j-1-\mathsf{lbq}}\cdot\mathbf{M}_1\cdot\mathbf{M}_{a_{\mathsf{lbq}-2}}\cdot\mathbf{M}_{a_{\mathsf{lbq}-3}}\cdots\mathbf{M}_{a_1}\cdot\mathbf{M}_{a_0}\cdot$$
$$\left(\lim_{n\to\infty}\prod\nolimits_{i=1}^{n}\mathbf{M}_{\mathsf{dyad}\left(\frac{\widetilde{\varpi}\,{}^{\nu^{\mathsf{d}}}+\{\{2^{\nu^\star}\widetilde{\omega}\}\}}{2^{\nu^{\mathsf{d}}}},i\right)}\right)\vec{\mathbf{1}} =$$
$$\mathbf{M}_0^{\nu-j-1-\mathsf{lbq}}\cdot\mathbf{M}_1\cdot\mathbf{M}_{a_{\mathsf{lbq}-2}}\cdot\mathbf{M}_{a_{\mathsf{lbq}-3}}\cdots\mathbf{M}_{a_1}\cdot\mathbf{M}_{a_0}\cdot\vec{\mathbf{V}}_{\nu^{\mathsf{d}}}^{(\widetilde{\omega})},$$

where $\mathsf{binary}\big(q-\tau_0(j,\nu)\big) = 1a_{\mathsf{lbq}-2}a_{\mathsf{lbq}-3}\cdots a_1a_0$ has length lbq. Thus, by using Corollary 5.36 the second statement of (b) follows.

To prove (c) note that $\tau_1(j,\nu) \le q \le 2^{\nu-j}-1$ is equivalent to $\frac{q}{2^\nu} < \frac{1}{2^j}$ and $\eta_j\left(\frac{q}{2^\nu}\right) = \left\lfloor\frac{q}{2^{\nu-j-1}} + \{\{2^j\widetilde{\omega}\}\}\right\rfloor = 2$ by Remark 5.37. Then the proof of (c) follows "mutatis mutandis" the proof of (b), and replacing $\tau_0(j,\nu)$ by $\tau_1(j,\nu)$. □

5.5.2.1. *Efficient computation of the vectors* $\overrightarrow{\boldsymbol{uj}}^{\boldsymbol{\omega}}(n)$, $\overrightarrow{\boldsymbol{uj}}^{\boldsymbol{\omega}}\left(n-\frac{1}{2}\right)$. We will present an analogous of Lemma 5.21 for the rotated case. Please keep in mind the definition of base(k) as given in Equation 5.15.

LEMMA 5.40. *Let* $\ell\in\mathbb{Z}$ *then*

$$\text{base}\left(u\left(\frac{{}^j\varpi}{2}+2^j\ell-n+\frac{\delta}{2}\right)\right)\cap\text{supp}(h)\neq\emptyset,$$

if and only if ℓ *is such that*

$$\ell_{\min} := \left\lceil\frac{1-p+n-\frac{{}^j\varpi+\delta}{2}}{2^j}\right\rceil \le \ell \le \left\lfloor\frac{p+n-\frac{{}^j\varpi+\delta+1}{2}}{2^j}\right\rfloor =: \ell_{\max}. \tag{5.26}$$

PROOF. The proof is completely analogous to the one for Lemma 5.21. □

REMARK 5.41. Analogously to Remark 5.22 this $\ell_{\max}$ and $\ell_{\min}$ allow us to actually compute the vectors $\overrightarrow{\mathsf{uj}}^{\boldsymbol{\omega}}(n)$, $\overrightarrow{\mathsf{uj}}^{\boldsymbol{\omega}}\left(n-\frac{1}{2}\right)$, since we have limited the values of ℓ in their definition to a finite set. That is,

$$\overrightarrow{\mathsf{uj}}^{\boldsymbol{\omega}}(n) = \sum_{\ell\in\mathbb{Z}} u\left(\frac{{}^j\varpi}{2}+2^j\ell-n\right) = \sum_{\ell_{\min}}^{\ell_{\max}} u\left(\frac{{}^j\varpi}{2}+2^j\ell-n\right),$$
$$\overrightarrow{\mathsf{uj}}^{\boldsymbol{\omega}}\left(n-\tfrac{1}{2}\right) = \sum_{\ell\in\mathbb{Z}} u\left(\frac{{}^j\varpi+\delta}{2}+2^j\ell-n\right) = \sum_{\ell_{\min}}^{\ell_{\max}} u\left(\frac{{}^j\varpi+\delta}{2}+2^j\ell-n\right).$$

■

Note that again in this case we have that for $2^j > p$ that for most values of n $\ell_{\min} > \ell_{max}$, hence $\overrightarrow{\mathsf{uj}}^{\boldsymbol{\omega}}\left(n-\frac{\delta}{2}\right) = 0$, in particular we get

$$\ell_{\min} = \begin{cases} 0 & \text{if } n \le p-1+\frac{{}^j\varpi+\delta+1}{2}, \\ 1 & \text{if } n > p-1+\frac{{}^j\varpi+\delta+1}{2}, \end{cases} \quad\text{and,}$$
$$\ell_{\max} = \begin{cases} -1 & \text{if } n \le 1-p+\frac{{}^j\varpi+\delta}{2}, \\ 0 & \text{if } n > 1-p+\frac{{}^j\varpi+\delta}{2}. \end{cases} \tag{5.27}$$

Moreover, since ${}^j\varpi < 2^j$, we get that both

$$p-1+\frac{{}^j\varpi+\delta+1}{2} < 2^j-1 \quad\text{and}\quad 1-p+\frac{{}^j\varpi+\delta}{2}2^j-1 < 2^j-1.$$

In particular, we get similar sparsity results as in Remark 5.25, with the areas $\overrightarrow{\mathsf{uj}}^{\boldsymbol{\omega}}\left(n-\frac{\delta}{2}\right)\neq 0$ can be seen represented in Figure 5.2.

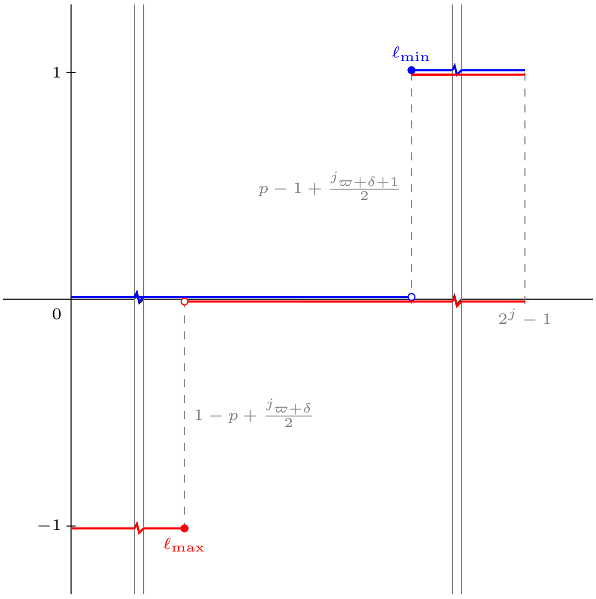


FIGURE 5.2. Graphs of $\ell_{\min}$ and $\ell_{\max}$ as functions of n when $2^j > p$ for the rotated case. Note that we have $\ell_{\max} < \ell_{\min}$ except for a region in the centre.

PROPOSITION 5.42. *Let*

$$\ell_{\min} := \left\lceil \frac{1-p+n-\frac{{}^{j}\varpi+\delta}{2}}{2^j} \right\rceil \leq \ell \leq \left\lfloor \frac{p+n-\frac{{}^{j}\varpi+\delta+1}{2}}{2^j} \right\rfloor =: \ell_{\max},$$

be as in Lemma 5.40. Then the following statements hold:

(a) For every $j \in \mathbb{N}$, $n \in \{0,1,\dots,2^j-1\}$ and $i \in \{0,1,2,\dots,2p-2\}$,

$$\left[\overrightarrow{\boldsymbol{uj}}^{\boldsymbol{\omega}}(n)\right]_i = (-1)^i \sum_{\substack{\ell=\ell_{\min} \\ i+1+2n-(2^{j+1}\ell+{}^{j}\varpi)\in\operatorname{supp}(h)}}^{\ell_{\max}} h\left[i+1+2n-\left(2^{j+1}\ell+{}^{j}\varpi\right)\right].$$

(b) For every $j \in \mathbb{N}$ and $n \in \{0,1,\dots,2^j-1\}$,

$$\left[\overrightarrow{\boldsymbol{uj}}^{\boldsymbol{\omega}}\left(n-\tfrac{1}{2}\right)\right]_0 = \sum_{\ell=\left\lceil \frac{2n-2p+1-{}^{j}\varpi}{2^{j+1}} \right\rceil}^{\left\lfloor \frac{2n-{}^{j}\varpi}{2^{j+1}} \right\rfloor} h\left[2n-\left(2^{j+1}\ell+{}^{j}\varpi\right)\right].$$

(c) For the special case $n=-1$.

$$\left[\overrightarrow{\boldsymbol{u1}}^{\boldsymbol{\omega}}(-1)\right]_0 = \sum_{\left\lceil \frac{-2p-2+1-{}^{j}\varpi}{2^{j+1}} \right\rceil}^{\left\lfloor \frac{-3-{}^{j}\varpi}{2^{j+1}} \right\rfloor} h\left[2n-3-4\ell\right].$$

Furthermore, for every $j \in \mathbb{N} \setminus \{1\}$ and $n \in \{0,1,\dots,2^j-1\}$,

$$\left[\overrightarrow{\boldsymbol{uj}}^{\boldsymbol{\omega}}(n-1)\right]_0 = \sum_{\ell=1+\left\lfloor -\frac{2p+1+{}^{j}\varpi}{2^{j+1}} \right\rfloor}^{0} h\left[2n-1-\left(2^{j+1}\ell+{}^{j}\varpi\right)\right].$$

(d) For every $j \in \mathbb{N}$, $n \in \{0, 1, \ldots, 2^j - 1\}$, $t \in \{1, 2\}$ and $i \in \{t, t+1, \ldots, 2p-2\}$,

$$\big[\overrightarrow{\boldsymbol{uj}}^{\omega}\big(n - t\tfrac{1}{2}\big)\big]_i = (-1)^t \big[\overrightarrow{\boldsymbol{uj}}^{\omega}(n)\big]_{i-t}.$$

(e) For every $j \in \mathbb{N}$ and $n \in \{0, 1, \ldots, 2^j - 1\}$,

$$\big[\overrightarrow{\boldsymbol{uj}}^{\omega}(n-1)\big]_1 = -\big[\overrightarrow{\boldsymbol{uj}}^{\omega}\big(n - \tfrac{1}{2}\big)\big]_0.$$

Proof. For every $t \in \mathbb{Z}^+$ and $x \in \mathbb{R}$ we have

$$d\big(x + \tfrac{t}{2}\big) = 1 - \big\lfloor 2\big(x + \tfrac{t}{2}\big)\big\rfloor = 1 - \lfloor 2x \rfloor - t = d(x) - t.$$

Consequently,

$$d\Big(\tfrac{{}^{j}\varpi}{2} + (2^j\ell - n) + \tfrac{t}{2}\Big) = d\Big(\tfrac{{}^{j}\varpi}{2} + (2^j\ell - n)\Big) - t. \tag{5.28}$$

We start by proving statements (d) and (e):

$$\begin{aligned}
\big[\overrightarrow{\mathbf{uj}}^{\omega}\big(n - t\tfrac{1}{2}\big)\big]_i = \left[\sum_{\ell\in\mathbb{Z}} u\Big(\tfrac{{}^{j}\varpi + t}{2} + (2^j\ell - n)\Big)\right]_i = \sum_{\ell\in\mathbb{Z}} \Big[u\Big(\tfrac{{}^{j}\varpi + t}{2} + (2^j\ell - n)\Big)\Big]_i = \\
(-1)^i \sum_{\ell\in\mathbb{Z}} h\Big[i + d\Big(\tfrac{{}^{j}\varpi + t}{2} + (2^j\ell - n)\Big)\Big] = \\
(-1)^t(-1)^{i-t} \sum_{\ell\in\mathbb{Z}} h\Big[(i-t) + d\Big(\tfrac{{}^{j}\varpi}{2} + (2^j\ell - n)\Big)\Big] = (-1)^t \big[\overrightarrow{\mathbf{uj}}^{\omega}(n)\big]_{i-t}.
\end{aligned}$$

Also, from the previous equalities,

$$\begin{aligned}
\big[\overrightarrow{\mathbf{uj}}^{\omega}(n-1)\big]_1 &= -\sum_{\ell\in\mathbb{Z}} h\Big[1 + d\Big(\tfrac{{}^{j}\varpi}{2} + (2^j\ell - n) + \tfrac{2}{2}\Big)\Big] \\
&= -\sum_{\ell\in\mathbb{Z}} h\Big[1 + d\Big(\tfrac{{}^{j}\varpi}{2} + (2^j\ell - n)\Big) - 2\Big] \\
&= -\sum_{\ell\in\mathbb{Z}} h\Big[d\Big(\tfrac{{}^{j}\varpi}{2} + (2^j\ell - n) + \tfrac{1}{2}\Big)\Big] = -\big[\overrightarrow{\mathbf{uj}}^{\omega}\big(n - \tfrac{1}{2}\big)\big]_0.
\end{aligned}$$

The proof of (a) follows directly from the definitions of $\ell_{\min}$ and $\ell_{\max}$ for the rotated case. (b) and (c), follow from the construction of $\overrightarrow{\mathbf{uj}}^{\omega}$ and the particularization to $i = 0$. □

5.6. Specialization to $p = 10$

The reader will note that all the results from Chapter 3 have been obtained using Daubechies Wavelets with 10 vanishing moments. For the sake of completeness, we will write the specialization of Propositions 5.23 and 5.42 to $p = 10$ without proof.

Corollary 5.43 (Specialisation of Proposition 5.23 to $p = 10$). *For $p = 10$ the following statements hold:*

(1) For $j = 1$ and $n \in \{0, 1\}$ we have

$$\big[\overrightarrow{\boldsymbol{u1}}(n)\big]_i = (-1)^i \sum_{k=0}^{4} h\big[\alpha^{1,n}(i) + 4k\big]$$

for every $i \in \{0, 1, 2, \ldots, 18\}$. Moreover,

$$\big[\overrightarrow{\boldsymbol{u1}}\big(n - \tfrac{1}{2}\big)\big]_0 = \sum_{k=0}^{4} h\big[2n + 4k\big].$$

(2) *For $j = 2$ and $n \in \{0,1,2,3\}$ we have*

$$\left[\overrightarrow{\boldsymbol{u2}}(n)\right]_i = (-1)^i \sum_{k=0}^{\left\lfloor \frac{19-\alpha^{2,n}(i)}{8} \right\rfloor} h\left[\alpha^{2,n}(i) + 8k\right],$$

for every $i \in \{0,1,2,\dots,18\}$. Moreover,

$$\left[\overrightarrow{\boldsymbol{u2}}(n - \tfrac{1}{2})\right]_0 = \sum_{k=0}^{\left\lfloor \frac{19-2n}{8} \right\rfloor} h\left[2n + 8k\right].$$

(3) *For $j = 3$ and $n \in \{0,1,2,3,4,5,6,7\}$ we have*

$$\left[\overrightarrow{\boldsymbol{u3}}(n)\right]_i = (-1)^i \begin{cases} h\left[\alpha^{3,n}(i)\right] & \textit{if } \alpha^{3,n}(i) > 3, \textit{ and} \\ h\left[\alpha^{3,n}(i)\right] + h\left[\alpha^{3,n}(i) + 16\right] & \textit{otherwise,} \end{cases}$$

for every $i \in \{0,1,2,\dots,18\}$. Moreover,

$$\left[\overrightarrow{\boldsymbol{u3}}^{\omega}(n - \tfrac{1}{2})\right]_0 = \begin{cases} h[\mathsf{mod}(2n+7,\,16)] & \textit{if } \mathsf{mod}(2n+7,\,16) > 3, \textit{ and} \\ h[\mathsf{mod}(2n+7,\,16)] + h[\mathsf{mod}(2n+7,\,16) + 16] & \textit{otherwise,} \end{cases}$$

$$\left[\overrightarrow{\boldsymbol{u3}}^{\omega}(n - 1)\right]_0 = \begin{cases} h[\mathsf{mod}(2n+6,\,16)] & \textit{if } \mathsf{mod}(2n+6,\,16) > 3, \textit{ and} \\ h[\mathsf{mod}(2n+6,\,16)] + h[\mathsf{mod}(2n+6,\,16) + 16] & \textit{otherwise,} \end{cases}$$

and

$$\left[\overrightarrow{\boldsymbol{u3}}^{\omega}(n - 1)\right]_1 = -\begin{cases} h[\mathsf{mod}(2n+7,\,16)] & \textit{if } \mathsf{mod}(2n+7,\,16) > 3, \textit{ and} \\ h[\mathsf{mod}(2n+7,\,16)] + h[\mathsf{mod}(2n+7,\,16) + 16] & \textit{otherwise.} \end{cases}$$

(4) *For $j = 4$ and $n \in \{0,1,\dots,15\}$ we have*

$$\left[\overrightarrow{\boldsymbol{u4}}(n)\right]_i = \begin{cases} (-1)^i\, h[i+1] \quad \textit{for } i \in \{0,1,2,\dots,18\} & \textit{when } n = 0, \\ \left\{\begin{array}{ll} (-1)^i\, h[i+1+2n] & \textit{for } i \in [0, 18-2n] \cap \mathbb{Z} \\ 0 & \textit{for } i \in [19-2n, 18] \cap \mathbb{Z} \end{array}\right\} & \textit{when } 1 \le n \le 6, \\ \left\{\begin{array}{ll} (-1)^i\, h[i+1+2n] & \textit{for } i \in [0, 18-2n] \cap \mathbb{Z} \\ 0 & \textit{for } i \in [19-2n, 30-2n] \cap \mathbb{Z} \\ (-1)^i\, h[i+2n-31] & \textit{for } i \in [31-2n, 18] \cap \mathbb{Z} \end{array}\right\} & \textit{when } n \in \{7,8,9\}, \\ \left\{\begin{array}{ll} 0 & \textit{for } i \in [0, 30-2n] \cap \mathbb{Z} \\ (-1)^i\, h[i+2n-31] & \textit{for } i \in [31-2n, 18] \cap \mathbb{Z} \end{array}\right\} & \textit{when } n \ge 10. \end{cases}$$

Moreover,

$$\left[\overrightarrow{\boldsymbol{u4}}(n - \tfrac{1}{2})\right]_0 = \begin{cases} h[2n] & \textit{for } n \le 9, \textit{ and} \\ 0 & \textit{otherwise.} \end{cases}$$

(5) *For $j \ge 5$ and $n \in \{0,1,\dots,2^j - 1\}$ we have*

$$\left[\overrightarrow{\boldsymbol{uj}}(n)\right]_i = \begin{cases} (-1)^i\, h[i+1] \quad \textit{for } i \in \{0,1,2,\dots,18\} & \textit{when } n = 0, \\ \left\{\begin{array}{ll} (-1)^i\, h[i+1+2n] & \textit{for } i \in [0, 18-2n] \cap \mathbb{Z} \\ 0 & \textit{for } i \in [19-2n, 18] \cap \mathbb{Z} \end{array}\right\} & \textit{when } 1 \le n \le 9, \\ 0 \quad \textit{for } i \in \{0,1,2,\dots,18\} & \textit{when } 10 \le n \le 2^j - 10, \\ \left\{\begin{array}{ll} 0 & \textit{for } i \in \left[0, \widetilde{n} - 1\right] \cap \mathbb{Z} \\ (-1)^i\, h\left[i - \widetilde{n}\right] & \textit{for } i \in \left[\widetilde{n}, 18\right] \cap \mathbb{Z} \end{array}\right\} & \textit{when } n \ge 2^j - 9, \end{cases}$$

where $\widetilde{n} := 2^{j+1} - 2n - 1 \in [1, 17] \cap \mathbb{Z}$. *Moreover,*

$$\big[\overrightarrow{\boldsymbol{uj}}(n - \tfrac{1}{2})\big]_0 = \begin{cases} h[2n] & \textit{for } n \leq 9, \textit{ and} \\ 0 & \textit{otherwise.} \end{cases}$$

Corollary 5.44 (Specialisation of Proposition 5.42 to $p = 10$). *For* $p = 10$ *the following statements hold:*

(1) For $j = 1$ *and* $n \in \{0, 1\}$ *we have*

$$\big[\overrightarrow{\boldsymbol{u1}}^{\boldsymbol{\omega}}(n)\big]_i = (-1)^i \sum_{\ell=0}^{4} h\big[\mathsf{mod}(i + 2n + 3,\, 4) + 4\ell\big]$$

for every $i \in \{0, 1, 2, \ldots, 18\}$. *Moreover,*

$$\big[\overrightarrow{\boldsymbol{u1}}^{\boldsymbol{\omega}}(n - \tfrac{1}{2})\big]_0 = \sum_{k=0}^{4} h\big[\mathsf{mod}(2(n+1),\, 4) + 4k\big], \textit{ and}$$

$$\big[\overrightarrow{\boldsymbol{u1}}^{\boldsymbol{\omega}}(n - 1)\big]_0 = \sum_{k=0}^{4} h\big[\mathsf{mod}(2n + 1,\, 4) + 4k\big].$$

(2) For $j = 2$ *and* $n \in \{0, 1, 2, 3\}$ *we have*

$$\big[\overrightarrow{\boldsymbol{u2}}^{\boldsymbol{\omega}}(n)\big]_i = (-1)^i \sum_{\ell=0}^{\left\lfloor \frac{19 - \mathsf{mod}(i+2n+5,\, 8)}{8} \right\rfloor} h\big[\mathsf{mod}(i + 2n + 5,\, 8) + 8\ell\big],$$

for every $i \in \{0, 1, 2, \ldots, 18\}$. *Moreover,*

$$\big[\overrightarrow{\boldsymbol{u2}}^{\boldsymbol{\omega}}(n - \tfrac{1}{2})\big]_0 = \sum_{k=0}^{\left\lfloor \frac{19 - \mathsf{mod}(2n+4,\, 8)}{8} \right\rfloor} h\big[\mathsf{mod}(2n + 4,\, 8) + 8k\big], \textit{ and}$$

$$\big[\overrightarrow{\boldsymbol{u2}}^{\boldsymbol{\omega}}(n - 1)\big]_0 = \sum_{k=0}^{\left\lfloor \frac{19 - \mathsf{mod}(2n+3,\, 8)}{8} \right\rfloor} h\big[\mathsf{mod}(2n + 3,\, 8) + 8k\big].$$

(3) For $j = 3$, $n \in \{0, 1, 2, 3, 4, 5, 6, 7\}$ *and* $i \in \{0, 1, 2, \ldots, 18\}$ *we have*

$$\big[\overrightarrow{\boldsymbol{u3}}(n)\big]_i = (-1)^i \begin{cases} h\big[\alpha^{3,n}(i)\big] & \textit{if } \alpha^{3,n}(i) > 3, \textit{ and} \\ h\big[\alpha^{3,n}(i)\big] + h\big[\alpha^{3,n}(i) + 16\big] & \textit{otherwise,} \end{cases}$$

where, $\alpha^{3,n}(i) := \mathsf{mod}(i + 2n + 8,\, 16) \in \{0, 1, \ldots, 15\}$. *Moreover,*

$$\big[\overrightarrow{\boldsymbol{u3}}^{\boldsymbol{\omega}}(n - \tfrac{1}{2})\big]_0 = \begin{cases} h[\mathsf{mod}(2n + 7,\, 16)] & \textit{if } \mathsf{mod}(2n + 7,\, 16) > 3, \textit{ and} \\ h[\mathsf{mod}(2n + 7,\, 16)] + h[\mathsf{mod}(2n + 7,\, 16) + 16] & \textit{otherwise,} \end{cases}$$

and

$$\big[\overrightarrow{\boldsymbol{u3}}^{\boldsymbol{\omega}}(n - 1)\big]_0 = \begin{cases} h[\mathsf{mod}(2n + 6,\, 16)] & \textit{if } \mathsf{mod}(2n + 6,\, 16) > 3, \textit{ and} \\ h[\mathsf{mod}(2n + 6,\, 16)] + h[\mathsf{mod}(2n + 6,\, 16) + 16] & \textit{otherwise.} \end{cases}$$

(4) For $j = 4$ and $n \in \{0, 1, \dots, 15\}$ we have

$$\left[\overrightarrow{\boldsymbol{u4}}^{\omega}(n)\right]_i = (-1)^i \begin{cases} \left\{\begin{array}{ll} h[i+2n+14] & \text{for } i \in [0, 5-2n] \cap \mathbb{Z} \\ 0 & \text{for } i \in [6-2n, 17-2n] \cap \mathbb{Z} \\ h[i+2n-18] & \text{for } i \in [18-2n, 18] \cap \mathbb{Z} \end{array}\right\} & \text{when } n \in \{0,1,2\}, \\ \left\{\begin{array}{ll} 0 & \text{for } i \in [0, 17-2n] \cap \mathbb{Z} \\ h[i+2n-18] & \text{for } i \in [18-2n, 18] \cap \mathbb{Z} \end{array}\right\} & \text{when } 3 \le n \le 8, \\ h[i] \qquad\qquad\quad \text{for } i \in \{0,1,2,\dots,18\} & \text{when } n = 9, \\ \left\{\begin{array}{ll} h[i+2n-18] & \text{for } i \in [0, 37-2n] \cap \mathbb{Z} \\ 0 & \text{for } i \in [38-2n, 18] \cap \mathbb{Z} \end{array}\right\} & \text{when } n \ge 10. \end{cases}$$

Moreover,

$$\left[\overrightarrow{\boldsymbol{u4}}^{\omega}\left(n - \tfrac{1}{2}\right)\right]_0 = \begin{cases} h[2n+13] & \text{if } n \le 3, \\ 0 & \text{if } 4 \le n \le 9, \\ h[2n-19] & \text{if } n \ge 10, \end{cases}$$

and

$$\left[\overrightarrow{\boldsymbol{u4}}^{\omega}(n-1)\right]_0 = \begin{cases} h[2n+12] & \text{if } n \le 3, \\ 0 & \text{if } 4 \le n \le 9, \, . \\ h[2n-20] & \text{if } n \ge 10, \end{cases}$$

(5) For $j \ge 5$ and $n \in \{0, 1, \dots, 2^j - 1\}$ we have

$$\left[\overrightarrow{\boldsymbol{uj}}^{\omega}(n)\right]_i = \begin{cases} 0 & \text{for } i \in \{0,1,2,\dots,18\} \text{ when } 2n < {}^{j}\varpi - 19, \\ (-1)^i \, h\big[i + 2n - ({}^{j}\varpi - 1)\big] & \begin{cases} \text{for } i \in \big[{}^{j}\varpi - (2n+1), 18\big] \cap \mathbb{Z} \\ \qquad \text{when } {}^{j}\varpi - 19 \le 2n < {}^{j}\varpi - 1, \\ \text{for } i \in \{0,1,2,\dots,18\} \\ \qquad \text{when } 2n \in \big\{{}^{j}\varpi - 1, {}^{j}\varpi\big\}, \\ \text{for } i \in \big[0, 18 + {}^{j}\varpi - 2n\big] \cap \mathbb{Z} \\ \qquad \text{when } {}^{j}\varpi < 2n \le 18 + {}^{j}\varpi, \end{cases} \\ 0 & \text{for } i \in \{0,1,2,\dots,18\} \text{ when } 2n > 18 + {}^{j}\varpi. \end{cases}$$

Moreover,

$$\left[\overrightarrow{\boldsymbol{uj}}^{\omega}\left(n - \tfrac{1}{2}\right)\right]_0 = \begin{cases} 0 & \text{if } 2n < {}^{j}\varpi, \\ h\big[2n - {}^{j}\varpi\big] & \text{if } {}^{j}\varpi \le 2n \le {}^{j}\varpi + 19, \\ 0 & \text{if } 2n > {}^{j}\varpi + 19, \end{cases}$$

and

$$\left[\overrightarrow{\boldsymbol{uj}}^{\omega}(n-1)\right]_0 = \begin{cases} 0 & \text{if } 2n < {}^{j}\varpi + 1, \\ h\big[2n - 1 - {}^{j}\varpi\big] & \text{if } {}^{j}\varpi + 1 \le 2n \le {}^{j}\varpi + 20, \\ 0 & \text{if } 2n > {}^{j}\varpi + 20. \end{cases}$$

Remark 5.45. Note that in the case when $j \ge 5$ in Corollary 5.44, we have that most values of $\left[\overrightarrow{\mathbf{uj}}^{\omega}(n)\right]_i$ will be zero. In particular, the vector $\overrightarrow{\mathbf{uj}}^{\omega}(n)$ will be zero whenever $n < \left\lfloor \frac{{}^{j}\varpi - 19}{2} \right\rfloor$.

Note however, that for the non-zero cases the values depend only on the parity of ${}^{j}\varpi$. That is, the values computed for $\overrightarrow{\mathbf{uj}}^{\omega}(n)$ depend only on wether ${}^{j}\varpi$ is odd or even. Hence, one only needs to compute the non-zero coefficients of ${}^{j}\varpi$ for two cases.

Finally note that in this case the effect of $\big[\overrightarrow{\mathbf{uj}}^{\boldsymbol{\omega}}\big(-\frac{1}{2}\big)\big]_0$ and $\big[\overrightarrow{\mathbf{uj}}^{\boldsymbol{\omega}}(-1)\big]_0$ should not be underestimated. It might well happen that $\big[\overrightarrow{\mathbf{uj}}^{\boldsymbol{\omega}}(n)\big]_i = 0$ for all i but $\big[\overrightarrow{\mathbf{uj}}^{\boldsymbol{\omega}}(-1)\big]_0 \neq 0$. $\blacksquare$

Conclusions

This are the conclusions corresponding to the second part of the thesis. We have successfully extended and consolidated the results obtained in [**AMR16**], which were expanded in David Romero's PhD thesis [**RiS15**], to which this present thesis hopes to be a worthy continuation.

Using the invariance equation, we have successfully presented a method that allows to compute the truncated wavelet expansion of an attractor φ of a quasi-periodic forced skew product. That is, $2^{\nu+1}$ coefficients $\left\{d_{-j,n} \approx \langle\varphi, \psi^{\text{PER}}_{-j,n}\rangle\right\}_{j,n}$ such that

$$\varphi(\theta) \approx a_0 + \sum_{j=0}^{\nu} \sum_{n=0}^{2^j-1} d_{-j,n}\psi^{\text{PER}}_{-j,n}(\theta).$$

Moreover, this method has shown an outstanding robustness, as it has been used to compute the expansion for many different types of systems. It clearly performs better with regular systems, but it even has been able to give good approximations strange attractors and capture their behaviour. The fact that the subsequent recovery of the attractor from its wavelet expansion so closely mimics the behaviour of the approximation by forward iteration also deserves notice.

Another remarkable achievement is to have been able to obtain these expansions with as many as 2^{30} coefficients, that is, $\nu = 29$. We believe that this is noteworthy, especially if one considers that Daubechies wavelets are very complicated to compute, becoming harder as the number of vanishing moments increase. Hence, we believe that having succeeded in this endeavour with $p = 10$ vanishing moments is undoubtedly a reason of self-satisfaction.

What is more, we have been able to use these truncated expansions to compute the Besov regularities of the attractors for different values of the parameters. In Section 4.3, we have tested the method with the Keller-GOPY and Alsedà-Misiurewicz systems, for which there exist analytic results concerning their strangeness. The regularity obtained with the method successfully mimics the expected analytic behaviour. In Section 4.4 we have profitably scouted the situation for the Nishikawa-Kaneko system. Even though we cannot call the results conclusive, we believe that the insight we have gained from our computations is valuable.

Finally, in Chapter 5 we have given a very thorough description on how one can evaluate (periodized) Daubechies wavelets extremely efficiently. The subsequent implementation of the various algorithms devised have allowed us to obtain the wavelet matrices to be used in Chapter 4 using very reasonable computing resources and in a manageable amount of time.

All in all, we have developed a semi-analytic method that allows us to better understand the strangeness of attractors as (lack of) regularity. This method should complement other semi-analytical methods such as the computation of the arc length of the curve [**JT08**] or the computation of the Hausdorff dimension [**GJ13**].

Further research

There are three main areas where one can keep investigating: control of the error, improving the numerical limitations, and further applications of the wavelet coefficients.

Firstly, a comprehensive theoretical study of the error is required. However, this is harrowing endeavour. First and foremost, because we have no a priory results on how *good* a truncated wavelet approximation might be, specially if our goal is to use Theorem 3.35 to obtain regularities. A clear example of this is the Nishikawa-Kaneko case, were the behaviour of the computed coefficients is not as clear cut as in the Keller-GOPY or the Alsedà-Misiurewicz case. In this sense, the desire to use as many coefficients as possible collides with the capabilities available hardware. Hence, giving an appropriate estimate of the error for this sort of cases seems wishful thinking. We are quite confident that the error cannot be too large, though. Mostly because Figures 4.4, 4.9 and 4.14 show that the wavelet approximations obtained are of very good quality.

The main issues with obtaining results are laid out in Section 4.5. In this sense, the quest for better preconditioners is probably what can put the finishing touches to the method explained in this thesis. A better preconditioner than the one we have been using so far might help with the cases of zero regularity, as well as with the Nishikawa-Kaneko system. We have tried to use the matrices that are, in a way, *natural* to the system: $\frac{1}{N}\Psi$, $\frac{1}{N}\Psi^\intercal$, $\frac{1}{N}\Psi_R$, $\frac{1}{N}\Psi_R^\intercal$, having seen that $\frac{1}{N}\Psi_R^\intercal$ is optimal for the Keller-GOPY and the Alsedà-Misiurewicz systems, but not for the Nishikawa-Kaneko case. Moreover, with better preconditioners, one might use some other iterative solver that might perform better then TFQMR. Having said that, we believe that we have obtained optimal results in the computing and storage of the wavelet matrices and in the the implementation of the matrix operations.

When it comes to further applications, Theorem 3.35 has a more general version that allows to compute the Besov regularity for general $\mathscr{B}^s_{p,q}$, which might allow for some finer tuning when detecting irregularities that might not be discerned within the spaces $\mathscr{B}^s_{\infty,\infty}$. However, we are quite sceptical when it comes to the opportunities for finer detection following this path due to the inclusivity results presented in Section 3.4.

A path that may lead to more interesting results if done carefully is the use of *two-microlocal spaces* $\mathcal{C}^{s,s'}_{x_0}$, which roughly measure the smoothness of a function at a given point x_0 when compared to the surrounding points. Jaffard shows that there is a correlation between the regularities s and s' and the wavelet expansion of a function in [**Jaf91**]. The main result is quite similar to Theorem 3.35.

THEOREM 5.46 ([**Jaf91**] Theorem 2). *A distribution u belongs to $\mathcal{C}^{s,s'}_{x_0}(\mathbb{R}^k)$ if and only if*

$$\left|\langle u, \psi_{j,n}\rangle\right| \leq C2^{(-k/2+s)j}\frac{1}{\left(1+|k-2^j x_0|\right)^{s'}}$$

Note that when $s' = 0$ this theorem corresponds with Theorem 3.35, which makes sense because if $s > 0$ and $s' = 0$ then $\mathcal{C}^{s,s'}_{x_0}$ corresponds to the classical Holder space. However, in this case the regularity depends on two parameters, s and s' which can no longer be solved by a simple linear fitting. Moreover, the role of each parameter must be properly understood within the context of the duality strangeness and regularity. This is not trivial and requires a deeper theoretical understanding.

Finally, when it comes to further applications an interesting area of study could be finding (probably numerical) dependencies between the wavelets coefficients and the attractors. This could allow us to study bifurcations directly from the wavelet expansion. This, however, is a beast of its own to tackle, since Daubechies wavelets are very hard to evaluate, hence the coefficients computed cannot be neither easily nor intuitively understood.

APPENDIX A

Some notes on the numerical implementations

This very technical Annex wants to be a natural continuation to Chapter 5. In it we will focus on showing the actual implementation of the wavelet matrices in the C programming language. In particular, we will present the date types used to store the matrices and how one can program the operations corresponding to multiplying a matrix by a vector and the transposed matrix by a vector. Since this Annex reflexes the algorithms and code used to obtain the results from Chapter 4, we present the results for the particular case when we are using Daubechies wavelets with $p = 10$ vanishing moments.

A.1. Data types for the Wavelet Matrices

In this section we will present the data types used for the storage of the wavelet matrices. Recall that in Remark 4.10 we saw that by Proposition 3.22 we had that fixing j, all the columns corresponding to $\psi^{\text{PER}}_{-j,n}$, $n = 0, \dots, 2^j - 1$ were actually permutations of one another. To this end, the natural way of reducing storage becomes storing only one of the columns for the whole block of size $N \times 2^j$. Which column will depend on the situation, though. Notice that this allows a massive reduction on the amount of memory required to store all the information. If we are considering matrices of size $N \times N$, now the size required is reduced to $N \times \log_2(N)$. As we shall see, we can further reduce the memory required to store the matrices by using the results regarding $\ell_{\max}$ and $\ell_{\min}$ from Remark 5.25 and equation (5.27) in Chapter 5. Using this results, we can avoid storing all the unnecessary zeroes for the blocks of j such that $2^j > p$. For the particular case we have used, $p = 10$, $j = 5$ is the smallest such value. Hence we have non-sparse blocks for $j =\leq 4$ and sparse blocks whenever $j \geq 5$.

A.1.1. Some general considerations regarding the data types. For the implementation, we wold like to be able to deal with the whole wavelet matrix as single entity. To this end, it is interesting to realize that there are in fact four types of blocks within a wavelet matrix.

Type 1 The first type corresponds to the first column, which corresponds to the vector $\overrightarrow{\mathbf{1}}$, i.e, all the entries are 1. Hence, we do not to store this at all, just taking this fact into account when implementing the matrix operations is enough.

Type 2 The second type of block corresponds to the second column, the one that stores the values of $\psi^{\text{PER}} = \psi^{\text{PER}}_{0,0}$. Recall that by Propositions 5.15 and 5.28 these column satisfies that the first $N/2$ entries and the last $N/2$ entries have the same absolute value but have different signs. Knowing this, we only need to store the first $N/2$ entries and change their signs when implementing matrix operations.

Type 3 The third type of block corresponds to the ones where j is such that $2^j \leq 2p - 1$. In this case, all the entries should be considered different than zero. However, by Proposition 3.22 we know that all the columns inside one j-block are simply shifts of the first column. Hence, if one takes this shifts into account, for each j-block with 2^j columns, only one needs to be stored.

Type 4 The fourth and final type of column is the one corresponding to j levels such that $2^j > 2p-1$. By the finiteness of the support many entries on this column will simply be zero. In Chapter 5, we have even been able to localize where the zeroes are thanks to the definitions of $\ell_{\max}$ and $\ell_{\min}$. Hence, for each j-block we have to take into account both the self similarities (and hence store only one column) and the sparsity (therefore storing only the non-zero values).

Therefore, when it comes to storing these different types of columns one needs to do it carefully so that what is stored is both enough to recover all the information and minimal in the sense that we would not want to use memory superfluously. As we said before, for the first column, one does not need to store any information. For the second column, an array of size $N/2$ suffices. As for the j-blocks with j such that $2^j \leq 2p-1$, we have defines the following data type in C that allows us to store all the information we need:

```
typedef struct {
  unsigned dimrows, dimcols;
  double  *base_column;
} Self_Similar_Matrix_EncodedByColumn;
```

Where dimrows corresponds to the division of the column in groupings that will shift around (see Examples 4.11 and 4.12, where the groupings of size dimrows are coloured the same). On the other hand, dimcols corresponds to 2^j, the number of columns in this j-block. In the array base_column we store the whole first column in the block. All the others can be reconstructed knowing dimrows and dimcols.

Finally, for the sparse column, we have defined another data type:

```
typedef struct {
  unsigned dimrows, dimcols;
  unsigned numnonzeroblocks, nonzerodimension;
  unsigned base_non_interruptus_column_number;
  double  *base_non_interruptus_column;
} Self_Similar_Matrix_Interruptus_EncodedByNonInterruptusColumn;
```

In here, dimcols and dimrows have the same definition as in Self_Similar_Matrix_EncodedByColumn. On the other hand, numnonzeroblocks corresponds to the number of blocs that are different from zero in a given column. Simmilarly, nonzerodimension correspond to the number of entries that are different from zero. We are aware that nonzerodimension = numnonzeroblocks·dimrows, but since we are going to use it repeteadly we considered that is was worth storing it in the data type. What is a bit more interesting is the integer base_non_interruptus_column_number. This corresponds to the column within the j-block that does not start with a zero (that is in the overall position within the matrix 2^j+base_non_interruptus_column_number). Note that due to the matrix structure there is only one such column within the j-block. Finally, in the array base_non_interruptus_column we store the values different from zero of the column in position base_non_interruptus_column_number.

One of the goals of the current implementations are that we want the programming to be as simple as possible. Therefore, note that the data types for each type of column can be used far either the Wavelet Matrix Ψ and the Rotated Wavelet Matrix Ψ_R. The main change will be in the definition of the integer base_non_interruptus_column_number. The reason behind this stems from the different definitions of $\ell_{\max}$ and $\ell_{\min}$ and can be clearly seen in Figures 5.1 and 5.2.

Using the two data types described above, we can create a new data type called Wavelet_Matrix to store both the non-rotated and the rotated wavelet matrices.

```
typedef struct {
  unsigned short nu;
  unsigned N, Nhalf;
  double *psi_PER_0_vals;
  Self_Similar_Matrix_EncodedByColumn *psi_PER;
  Self_Similar_Matrix_Interruptus_EncodedByNonInterruptusColumn
  *psi_PER_big;
} Wavelet_Matrix;
```

In this case remains to explain the integers nu, N and Nhalf and the array psi_PER_0_vals. As the name implies nu corresponds to ν, the number of j-blocks used. $\mathtt{N} = 2^\nu$ corresponds to the dimension of the matrix. $\mathtt{Nhalf} = \mathtt{N}/2$ is stored because it is often used. Finally, the array psi_PER_0_vals corresponds to the storing of the first half od the second column. Note that both the psi_PER and psi_PER_big are considered to be arrays, since we have an entry in the array for each j-block.

A.2. Matrix operations

In this section we will show the implementations of the matrix operations we are going to use, that is $\Psi \cdot \vec{v}$, $\Psi^\mathsf{T} \cdot \vec{v}$, $\Psi_R \cdot \vec{v}$, and $\Psi_R^\mathsf{T} \cdot \vec{v}$. First, however, we will need a couple of auxiliary functions. All of them are based on their BLAS library equivalent.

The first one is dotproductsdif, which given three vectors $\vec{v}$, $\vec{u}$ and $\vec{w}$ computes $\vec{v} \cdot \vec{u} - \vec{v} \cdot \vec{w}$.

```
double dotproductsdif(unsigned size, double *c, double *a, double *b){
  double dtemp;
  unsigned char rem = size % 5;
  switch (rem) {
    case 0: dtemp = 0.0; break;
    case 1: dtemp = c[0]*(a[0] - b[0]); break;
    case 2: dtemp = c[0]*(a[0] - b[0]) +
                    c[1]*(a[1] - b[1]); break;
    case 3: dtemp = c[0]*(a[0] - b[0]) +
                    c[1]*(a[1] - b[1]) +
                    c[2]*(a[2] - b[2]); break;
    case 4: dtemp = c[0]*(a[0] - b[0]) +
                    c[1]*(a[1] - b[1]) +
                    c[2]*(a[2] - b[2]) +
                    c[3]*(a[3] - b[3]); break;
  }
  for(unsigned i=rem; i < size ; i += 5){
    dtemp += c[i]*(a[i] - b[i]) +
             c[i+1]*(a[i+1] - b[i+1]) +
             c[i+2]*(a[i+2] - b[i+2]) +
             c[i+3]*(a[i+3] - b[i+3]) +
             c[i+4]*(a[i+4] - b[i+4]);
  }
  return dtemp;
}
```

The next function, dotproduct computes the dot product between two vectors.

```
double dotproduct(unsigned size, double *a, double *b){
  double dtemp;
  unsigned char rem = size % 5;
  switch (rem) {
    case 0: dtemp = 0.0; break;
    case 1: dtemp = a[0]*b[0]; break;
    case 2: dtemp = a[0]*b[0] + a[1]*b[1]; break;
    case 3: dtemp = a[0]*b[0] + a[1]*b[1] +
                    a[2]*b[2]; break;
    case 4: dtemp = a[0]*b[0] + a[1]*b[1] +
                    a[2]*b[2] + a[3]*b[3];
  }
  for(unsigned i=rem; i < size ; i += 5){
    dtemp += a[i]*b[i] + a[i+1]*b[i+1] + a[i+2]*b[i+2] +
             a[i+3]*b[i+3] + a[i+4]*b[i+4];
  }
  return dtemp;
}
```

Now, we have the function `vectsum`. It simply computes the sum of all the entries of an array.

```
double vectsum(unsigned size, double *a){
  double dtemp = 0.0; register unsigned i;
  unsigned char rem = size % 6;
  for(i=0; i < rem; i++) dtemp += a[i];
  for( ; i < size ; i += 6) {
    dtemp += a[i] + a[i+1] + a[i+2] +
             a[i+3] + a[i+4] + a[i+5];
  }
  return dtemp;
}
```

The penultimate auxiliary function simply computes $a\vec{v} + \vec{u}$ and stores it in the memory corresponding to vector $\vec{u}$.

```
void DAXPY(unsigned size, double alpha, double *a, double *b){
  if(fabs(alpha) < 1.e-16) return;
  for(unsigned i=0 ; i < size ; i++) b[i] += alpha*a[i];
}
```

The final function of this section simply sets all the entries of a vector to a fixed number a.

```
void set_constant_vector(unsigned size, double a, double *b){
  b[0] = b[1] = b[2] = b[3] = b[4] = a;
  for(unsigned i=size % 5; i < size ; i += 5) {
    b[i] = a; b[i+1] = a; b[i+2] = a;
    b[i+3] = a; b[i+4] = a;
  }
}
```

To this point, all of the data types and functions presented have been general in scope. From this point onwards, we will focus only in the specialization Daubechies wavelets with $p = 10$ vanishing moments.

A.2.2. Functions corresponding to the Wavelet Matrix. In this section we will present the code for the functions we have used for the products $\Psi \cdot \vec{v}$ and $\Psi^{\mathsf{T}}\vec{v}$. For all the functions the inputs are the matrix to be multiplied, the array `x`, which corresponds to the vector $\vec{v}$ and the array `b`, where the result is stored.

A.2.2.1. *Matrix times vector.* The function is based in the fact that one can understand the product of a matrix and a vector columnwise as follows:

$$\begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N-1,0} & a_{N-1,1} & \cdots & a_{N-1,N-1} \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{N-1} \end{pmatrix} =$$

$$v_0 \begin{pmatrix} a_{0,0} \\ a_{1,0} \\ \vdots \\ a_{N-1,0} \end{pmatrix} + v_1 \begin{pmatrix} a_{0,1} \\ a_{1,1} \\ \vdots \\ a_{N-1,1} \end{pmatrix} + \cdots + v_{N-1} \begin{pmatrix} a_{N-1,0} \\ a_{N-1,1} \\ \vdots \\ a_{N-1,N-1} \end{pmatrix}$$

Using this principle and the internal structure of the matrix we get the following code for the matrix multiplication.

```
void Wavelet_Matrix_times_vector (Wavelet_Matrix *Wm,
                                  double *x, double *b){
  set_constant_vector(Wm->N, x[0], b);
  DAXPY(Wm->Nhalf, x[1], Wm->psi_PER_0_vals, b);
  DAXPY(Wm->Nhalf, -x[1], Wm->psi_PER_0_vals, b+Wm->Nhalf);

  for(unsigned j=1; j<5; j++){
    DAXPY(Wm->N, x[Wm->psi_PER[j].dimcols],
          Wm->psi_PER[j].base_column, b);
    for(unsigned n=1, shift=Wm->psi_PER[j].dimrows;
                 n<Wm->psi_PER[j].dimcols;
                 n++, shift+=Wm->psi_PER[j].dimrows){
      DAXPY(shift, x[Wm->psi_PER[j].dimcols+n],
            Wm->psi_PER[j].base_column+(Wm->N-shift), b);
      DAXPY(Wm->N-shift, x[Wm->psi_PER[j].dimcols+n],
            Wm->psi_PER[j].base_column, b+shift);
    }
  }
  for(unsigned j=5 ; j < Wm->nu ; j++){
    register unsigned n, shift;
    //Type I columns
    for(n=0,
        shift=Wm->psi_PER_big[j].base_non_interruptus_column_number*
        Wm->psi_PER_big[j].dimrows;
        n < Wm->psi_PER_big[j].base_non_interruptus_column_number;
        n++, shift -= Wm->psi_PER_big[j].dimrows){
          DAXPY(shift, x[Wm->psi_PER_big[j].dimcols+n],
                Wm->psi_PER_big[j].base_non_interruptus_column,
                b+(Wm->N-shift));
          DAXPY(Wm->psi_PER_big[j].nonzerodimension-shift,
                x[Wm->psi_PER_big[j].dimcols+n],
                Wm->psi_PER_big[j].base_non_interruptus_column+shift,
                b);
    }
    //Column base_non_iterruptus_column-number
    DAXPY(Wm->psi_PER_big[j].nonzerodimension,
          x[Wm->psi_PER_big[j].dimcols+
          Wm->psi_PER_big[j].base_non_interruptus_column_number],
          Wm->psi_PER_big[j].base_non_interruptus_column, b);
    //Type II columns
    for(n=Wm->psi_PER_big[j].base_non_interruptus_column_number+1,
        shift=Wm->psi_PER_big[j].dimrows;
        n <= Wm->psi_PER_big[j].dimcols +
        Wm->psi_PER_big[j].base_non_interruptus_column_number -
        Wm->psi_PER_big[j].numnonzeroblocks;
        n++, shift += Wm->psi_PER_big[j].dimrows){
          DAXPY(Wm->psi_PER_big[j].nonzerodimension,
                x[Wm->psi_PER_big[j].dimcols+n],
                Wm->psi_PER_big[j].base_non_interruptus_column,
                b+shift);
    }
    //Type I columns
    for(shift = Wm->psi_PER_big[j].dimrows;
        n < Wm->psi_PER_big[j].dimcols;
        n++, shift += Wm->psi_PER_big[j].dimrows){
          DAXPY(shift, x[Wm->psi_PER_big[j].dimcols+n],
                Wm->psi_PER_big[j].base_non_interruptus_column +
                (Wm->psi_PER_big[j].nonzerodimension - shift), b);
          DAXPY(Wm->psi_PER_big[j].nonzerodimension-shift,
                x[Wm->psi_PER_big[j].dimcols+n],
                Wm->psi_PER_big[j].base_non_interruptus_column,
                b+(Wm->N+shift-Wm->psi_PER_big[j].nonzerodimension));
    }
  }
}
```

A couple of notes in the implementation. Due to the Definitions of $\ell_{\max}$ and $\ell_{\min}$ in Remark 5.25, it is easy to see that there are two types of columns for this matrix:

$$\text{Type I: } \begin{pmatrix} \star \\ \vdots \\ \star \\ 0 \\ \vdots \\ 0 \\ \star \\ \vdots \\ \star \end{pmatrix} \qquad \text{Type II: } \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \star \\ \vdots \\ \star \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{A.1}$$

In the non rotated case we have that the columns begin with non-zero coefficients, and hence are of Type I. Moreover, the last columns are also of Type I. In between there is a big blob in the middle of Type II. In particular, for $p = 10$ we have 9 columns of Type I in the beginning and 10 columns in the end with $2^j - 19$ Type II columns in the middle.

A.2.2.2. *Transposed Matrix times Vector.* Similarly to the previous section, if we decompose the product of a transposed matrix by a vector in column-based operations we get that

$$\Psi^\intercal \vec{v} = \vec{b}$$

can be rewritten as

$$b[i] = \vec{v} \cdot \left(\Psi\right)_{*,i},$$

where $\left(\Psi\right)_{*,i}$ denotes the i-th column of the matrix Ψ. Hence, we can proceed to an almost identical way as for the regular product of matrix times vector basically changing the DAXPYs for dotproduct.

```
void Wavelet_Matrix_transposed_times_vector (Wavelet_Matrix *Wm,
                                            double *x, double *b){
  b[1] = dotproductsdif(Wm->Nhalf, Wm->psi_PER_0_vals, x,
                      x+Wm->Nhalf);
b[0] = vectsum(Wm->N, x);

for(unsigned j=1; j<5; j++){
  b[Wm->psi_PER[j].dimcols] =
        dotproduct(Wm->N, Wm->psi_PER[j].base_column, x);
  for(unsigned n=1, shift=Wm->psi_PER[j].dimrows;
      n < Wm->psi_PER[j].dimcols;
      n++, shift += Wm->psi_PER[j].dimrows){
        b[Wm->psi_PER[j].dimcols + n] =
            dotproduct(shift,
                  Wm->psi_PER[j].base_column+(Wm->N-shift), x) +
            dotproduct(Wm->N-shift,
                  Wm->psi_PER[j].base_column, x+shift);
  }
}

for(unsigned j=5 ; j < Wm->nu ; j++){
  register unsigned n, shift;
  double * bstrip = b + Wm->psi_PER_big[j].dimcols;
  //Type I columns
  for(n=0,
      shift=Wm->psi_PER_big[j].base_non_interruptus_column_number*
      Wm->psi_PER_big[j].dimrows;
      n < Wm->psi_PER_big[j].base_non_interruptus_column_number;
      n++, shift -= Wm->psi_PER_big[j].dimrows){
        bstrip[n] = dotproduct(shift,
          Wm->psi_PER_big[j].base_non_interruptus_column,
          x+(Wm->N-shift)) +
          dotproduct(Wm->psi_PER_big[j].nonzerodimension-shift,
          Wm->psi_PER_big[j].base_non_interruptus_column+shift, x);
  }
  //Column base_non_iterruptus_column-number
  bstrip[Wm->psi_PER_big[j].base_non_interruptus_column_number] =
      dotproduct(Wm->psi_PER_big[j].nonzerodimension,
                 Wm->psi_PER_big[j].base_non_interruptus_column, x);
  //Type II columns
  for(n=Wm->psi_PER_big[j].base_non_interruptus_column_number+1,
      shift=Wm->psi_PER_big[j].dimrows;
      n <= Wm->psi_PER_big[j].dimcols +
      Wm->psi_PER_big[j].base_non_interruptus_column_number -
      Wm->psi_PER_big[j].numnonzeroblocks;
      n++, shift += Wm->psi_PER_big[j].dimrows){
        bstrip[n] = dotproduct(Wm->psi_PER_big[j].nonzerodimension,
                    Wm->psi_PER_big[j].base_non_interruptus_column,
                    x+shift);
  }
  //Type I columns
  for( ; n < Wm->psi_PER_big[j].dimcols ;
      n++, shift += Wm->psi_PER_big[j].dimrows){
        bstrip[n] = dotproduct(Wm->N-shift,
                  Wm->psi_PER_big[j].base_non_interruptus_column,
                  x+shift) +
              dotproduct(Wm->psi_PER_big[j].nonzerodimension -
                    (Wm->N-shift),
                    Wm->psi_PER_big[j].base_non_interruptus_column+
                    (Wm->N-shift),x);
```

```
    }
  }
}
```

A.2.3. Functions corresponding to the Rotated Wavelet Matrix. In this subsection we will present the code for the products for the Ψ_R, the rotated wavelet matrix. We had to develop slightly different functions for this case. This mostly stems from the fact that within each j-block with $j \geq 5$ the Type I and Type II types of columns from Equation (A.1) are reversed. In particular, we have 19 consecutive columns of Type I leading up to the column `base_non_interruptus_column_number`, while the rest of the columns are of Type II. Hence the loop from $j \geq 5$ has been modified accordingly.

A.2.3.1. *Rotated Matrix times Vector.* The case is identical to the non-rotated case for $j \leq 4$. For $j \geq 5$ we simply need to take into account that the first columns are of Type II as in Equation (A.1).

```
void Rotated_Wavelet_Matrix_times_vector (Wavelet_Matrix *Wm,
                                      double *x, double *b){
  set_constant_vector(Wm->N, x[0], b);
  DAXPY(Wm->Nhalf, x[1], Wm->psi_PER_0_vals, b);
  DAXPY(Wm->Nhalf, -x[1], Wm->psi_PER_0_vals, b+Wm->Nhalf);
  for(unsigned j=1; j<5; j++){
    DAXPY(Wm->N, x[Wm->psi_PER[j].dimcols],
          Wm->psi_PER[j].base_column, b);
    for(unsigned n=1, shift=Wm->psi_PER[j].dimrows ;
        n < Wm->psi_PER[j].dimcols;
        n++, shift += Wm->psi_PER[j].dimrows){
      DAXPY(shift, x[Wm->psi_PER[j].dimcols+n],
            Wm->psi_PER[j].base_column+(Wm->N-shift), b );
      DAXPY(Wm->N-shift, x[Wm->psi_PER[j].dimcols+n],
            Wm->psi_PER[j].base_column, b+shift);
    }
  }
  for(unsigned j=5 ; j < Wm->nu ; j++){
    register unsigned n, shift;
    //Type II columns
    for(n=0, shift=(Wm->psi_PER_big[j].dimcols -
        Wm->psi_PER_big[j].base_non_interruptus_column_number)*
        Wm->psi_PER_big[j].dimrows;
        n <= Wm->psi_PER_big[j].base_non_interruptus_column_number-
        Wm->psi_PER_big[j].numnonzeroblocks;
        n++, shift += Wm->psi_PER_big[j].dimrows){
      DAXPY(Wm->psi_PER_big[j].nonzerodimension,
            x[Wm->psi_PER_big[j].dimcols+n],
            Wm->psi_PER_big[j].base_non_interruptus_column,
            b+shift);
    }
    //Type I column
    for(n=Wm->psi_PER_big[j].base_non_interruptus_column_number -
        Wm->psi_PER_big[j].numnonzeroblocks+1U,
        shift= Wm->psi_PER_big[j].nonzerodimension -
        Wm->psi_PER_big[j].dimrows;
        n < Wm->psi_PER_big[j].base_non_interruptus_column_number;
        n++, shift -= Wm->psi_PER_big[j].dimrows){
      DAXPY(shift, x[Wm->psi_PER_big[j].dimcols+n],
            Wm->psi_PER_big[j].base_non_interruptus_column,
            b+(Wm->N-shift));
          DAXPY(Wm->psi_PER_big[j].nonzerodimension-shift,
                x[Wm->psi_PER_big[j].dimcols+n],
                Wm->psi_PER_big[j].base_non_interruptus_column+shift,
                b);
    }
    //Column base_non_iterruptus_column-number
    DAXPY(Wm->psi_PER_big[j].nonzerodimension,
          x[Wm->psi_PER_big[j].dimcols+
          Wm->psi_PER_big[j].base_non_interruptus_column_number],
          Wm->psi_PER_big[j].base_non_interruptus_column, b);
    //Type II columns
    for(n=Wm->psi_PER_big[j].base_non_interruptus_column_number+1,
        shift=Wm->psi_PER_big[j].dimrows ;
        n < Wm->psi_PER_big[j].dimcols; n++,
        shift += Wm->psi_PER_big[j].dimrows){
          DAXPY(Wm->psi_PER_big[j].nonzerodimension,
                x[Wm->psi_PER_big[j].dimcols+n],
                Wm->psi_PER_big[j].base_non_interruptus_column,
                b+shift);
    }
  }
}
```

A.2.3.2. *Transposed Rotated Matrix times Vector.* Finally, this implementation follows the same structure as the non-transposed product but changing the DAXPYs for dotproducts, as it was done in the non-rotated case.

```
void Rotated_Wavelet_Matrix_transposed_times_vector
                        (Wavelet_Matrix *Wm, double *x, double *b){
  b[1] = dotproductsdif(Wm->Nhalf, Wm->psi_PER_0_vals,
                        x, x+Wm->Nhalf);
  b[0] = vectsum(Wm->N, x);
  for(unsigned j=1; j<5; j++){
    b[Wm->psi_PER[j].dimcols] =
                dotproduct(Wm->N, Wm->psi_PER[j].base_column, x);
    for(unsigned n=1, shift=Wm->psi_PER[j].dimrows ;
        n < Wm->psi_PER[j].dimcols ; n++,
        shift += Wm->psi_PER[j].dimrows){
          b[Wm->psi_PER[j].dimcols + n] =
              dotproduct(shift, Wm->psi_PER[j].base_column +
                        (Wm->N-shift), x) +
              dotproduct(Wm->N-shift, Wm->psi_PER[j].base_column,
                        x+shift);
    }
  }
  for(unsigned j=5 ; j < Wm->nu ; j++){
    register unsigned n, shift;
    double * bstrip = b + Wm->psi_PER_big[j].dimcols;
    Type II columns
    for(n=0, shift=(Wm->psi_PER_big[j].dimcols
        - Wm->psi_PER_big[j].base_non_interruptus_column_number)*
        Wm->psi_PER_big[j].dimrows; n <=
        Wm->psi_PER_big[j].base_non_interruptus_column_number-
        Wm->psi_PER_big[j].numnonzeroblocks;
        n++, shift += Wm->psi_PER_big[j].dimrows){
          bstrip[n] = dotproduct(Wm->psi_PER_big[j].nonzerodimension,
                      Wm->psi_PER_big[j].base_non_interruptus_column,
                      x+shift);
    }
    //Type I columns
    for(n=Wm->psi_PER_big[j].base_non_interruptus_column_number-
        Wm->psi_PER_big[j].numnonzeroblocks+1U,
        shift= Wm->psi_PER_big[j].nonzerodimension -
        Wm->psi_PER_big[j].dimrows;
        n < Wm->psi_PER_big[j].base_non_interruptus_column_number ;
        n++, shift -= Wm->psi_PER_big[j].dimrows){
          bstrip[n] = dotproduct(shift,
                        Wm->psi_PER_big[j].base_non_interruptus_column,
                        x+(Wm->N-shift)) +
                        dotproduct(
                          Wm->psi_PER_big[j].nonzerodimension-shift,
                          Wm->psi_PER_big[j].base_non_interruptus_column+
                          shift, x);
    }
    //Column base_non_iterruptus_column-number
    bstrip[Wm->psi_PER_big[j].base_non_interruptus_column_number] =
            dotproduct(Wm->psi_PER_big[j].nonzerodimension,
                          Wm->psi_PER_big[j].base_non_interruptus_column,
                          x);
    //Type II columns
    for(n=Wm->psi_PER_big[j].base_non_interruptus_column_number+1,
        shift=Wm->psi_PER_big[j].dimrows ;
        n < Wm->psi_PER_big[j].dimcols;
        n++, shift += Wm->psi_PER_big[j].dimrows){
          bstrip[n] = dotproduct(Wm->psi_PER_big[j].nonzerodimension,
                    Wm->psi_PER_big[j].base_non_interruptus_column,
                    x+shift);
    }
  }
}
```

Bibliography

[AC09] Lluís Alsedà and Sara Costa, *On the definition of strange nonchaotic attractor*, Fundamenta Mathematicae **206** (2009), no. 1, 23–39 (English).

[ALM00] Lluís Alsedà, Jaume Llibre, and Michał Misiurewicz, *Combinatorial dynamics and entropy in dimension one*, World Scientific, 2000.

[ALMM88] L. Alsedà, J. Llibre, F. Mañosas, and M. Misiurewicz, *Lower bounds of the topological entropy for continuous maps of the circle of degree one*, Nonlinearity **1** (1988), 463–479.

[AM08] Lluís Alsedà and Michał Misiurewicz, *Attractors for unimodal quasiperiodically forced maps*, J. Difference Equ. Appl. **14** (2008), no. 10-11, 1175–1196. MR 2447193 (2010c:37053)

[AM15] ———, *Skew product attractors and concavity*, Proceedings of the American Mathematical Society **143** (2015), no. 2, 703–716.

[AMR16] Lluís Alsedà, Josep M. Mondelo, and David Romero, *A numerical estimate of the regularity of a family of strange non–chaotic attractors*, submitted, April 2016.

[Bje09] Kristian Bjerklöv, *Sna's in the quasi-periodic quadratic family*, Communications Mathematical Physics (2009), no. 286, 137–161.

[Boo] *Boost C++ Library*, `https://www.boost.org/doc/libs/1_74_0/libs/math/doc/html/math_toolkit/daubechies.html`, Accessed: 2023-06-10.

[Boy86] P.L. Boyland, *Bifurcation of circle maps: Arnol'd tongues, bistability and rotation intervals*, Commun. Math. Phys. (1986), no. 106, 353–381.

[BS98] H. Broer and C. Simó, *Hill's equation with quasi-periodic forcing*, Boletim da Sociedade Brasileira de Matemática **29** (1998), no. 2, 253–293.

[Dau92] Ingrid Daubechies, *Ten lectures on wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 61, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992. MR 1162107 (93e:42045)

[DL91] Ingrid Daubechies and Jeffrey C. Lagarias, *Two-scale difference equations. I. Existence and global regularity of solutions*, SIAM J. Math. Anal. **22** (1991), no. 5, 1388–1410. MR 1112515 (92d:39001)

[DL92] ———, *Two-scale difference equations. II. Local regularity, infinite products of matrices and fractals*, SIAM J. Math. Anal. **23** (1992), no. 4, 1031–1079. MR 1166574 (93g:39001)

[dlLO99] R. de la Llave and R. Obaya, *Regularity of the composition operator in spaces of Hölder functions*, Discrete Contin. Dynam. Systems **5** (1999), no. 1, 157–184. MR MR1664481 (99m:47085)

[GJ13] M. Gröger and T. Jäger, *Dimensions of Attractors in Pinched Skew Products*, Comm. Math. Phys. **320** (2013), no. 1, 101–119. MR 3046991

[GOPY84] Celso Grebogi, Edward Ott, Steven Pelikan, and James A. Yorke, *Strange attractors that are not chaotic*, Phys. D **13** (1984), no. 1-2, 261–268. MR MR775290 (86g:58089)

[Har16] Godfrey H. Hardy, *Weierstrass's non-differentiable function*, Transactions of the American Mathematical Society **17** (1916), no. 3, 301–325.

[Har05] Carles Haro, Àlex and Simó, *To be or not to be an sna: That is the question*, Preprint: http://www.maia.ub.es/dsg/2005/0503haro.pdf (2005).

[HdlL06a] A. Haro and R. de la Llave, *A parameterization method for the computation of invariant tori and their whiskers in quasi-periodic maps: rigorous results*, J. Differential Equations **228** (2006), no. 2, 530–579. MR 2289544 (2007m:37066)

[HdlL06b] À. Haro and R. de la Llave, *A parameterization method for the computation of invariant tori and their whiskers in quasi-periodic maps: numerical algorithms*, Discrete Contin. Dyn. Syst. Ser. B **6** (2006), no. 6, 1261–1300. MR 2240743 (2007f:37090)

[HdlL07] A. Haro and R. de la Llave, *A parameterization method for the computation of invariant tori and their whiskers in quasi-periodic maps: explorations and mechanisms for the breakdown of hyperbolicity*, SIAM J. Appl. Dyn. Syst. **6** (2007), no. 1, 142–207. MR 2299977 (2008e:37057)

[Her79] M. Herman, *Sur la conjugaison différentiable des difféomorphismes du cercle à des rotations*, Publications Mathématiques de l'IHÉS **49** (1979), 5–233 (fr). MR 81h:58039

[HH94] J.F Heagy and S.M Hammel, *The birth of strange nonchaotic attractors*, Physica D: Nonlinear Phenomena **70** (1994), no. 1, 140–153.

[HW96] Eugenio Hernández and Guido Weiss, *A first course on wavelets*, Studies in Advanced Mathematics, CRC Press, Boca Raton, FL, 1996, With a foreword by Yves Meyer. MR MR1408902 (97i:42015)

[Ito81] R. Ito, *Rotation sets are closed*, Mathematical Proceedings of the Cambridge Philosophical Society **89** (1981), no. 1, 107–111.

[Jaf91] Stéphane Jaffard, *Pointwise smoothness, two-microlocalization and wavelet coefficients*, Publicacions Matemàtiques **35** (1991), no. 1.

[JO19] S. Janson and A. Öberg, *A piecewise contractive dynamical system and election methods*, Bulletin de la Société Mathématique de France **147** (2019), no. 3, 395–411.

[JS09] C. Simó J. Sánchez, M. Net, *Computation of invariant tori by newton–krylov methods in large-scale dissipative systems*, Physica D: Nonlinear Phenomena (2009), no. 239, 123–133.

[JT08] Àngel Jorba and Joan Carles Tatjer, *A mechanism for the fractalization of invariant curves in quasi-periodically forced 1-D maps*, Discrete Contin. Dyn. Syst. Ser. B **10** (2008), no. 2-3, 537–567. MR 2425056 (2010c:37140)

[Kan84] Kunihiko Kaneko, *Fractalization of torus*, Progress of Theoretical Physics **71** (1984), no. 5, 1112–1115.

[Kel96] Gerhard Keller, *A note on strange nonchaotic attractors*, Fund. Math. **151** (1996), no. 2, 139–148. MR MR1418993 (97j:58088)

[Mal98] Stéphane Mallat, *A wavelet tour of signal processing*, Academic Press Inc., San Diego, CA, 1998. MR MR1614527 (99m:94012)

[Mal09] ______, *A wavelet tour of signal processing*, Academic Press Inc., San Diego, CA, 2009. MR MR1614527 (99m:94012)

[Mil85] John Milnor, *On the concept of attractor*, Communications in Mathematical Physics **99** (1985), no. 2, 177 – 195.

[Mis82] M. Misiurewicz, *Periodic points of maps of degree one of a circle*, Ergodic Theory and Dynamical Systems **2** (1982), 221–227.

[Mis86] M. Misiurewicz, *Rotation intervals for a class of maps of the real line into itself*, Ergodic Theory and Dynamical Systems **6** (1986), no. 1, 117–132.

[Mis89] M. Misiurewicz, *Persistent rotation intervals for old maps*, Banach Center Publications (1989).

[MO15] X. Mora and M. Oliver, *Eleccions mitjançant el vot d'aprovació. el mètode de phragmén i algunes variants.*, Butlletí de la Societat Catalana de Matemàtiques **30** (2015), no. 1, 57–101.

[MSG20] L. Marangio, J Sedro, and S. et al. Galatolo, *Arnold maps with noise: Differentiability and non-monotonicity of the rotation number.*, J Stat Phys (2020), no. 179, 1594–1624.

[NK96] Takashi Nishikawa and Kunihiko Kaneko, *Fractalization of a torus as a strange nonchaotic attractor*, Phys. Rev. E **54** (1996), 6114–6124.

[Pav95] R. Pavani, *A numerical approximation of the rotation number*, Applied Mathematics and Computation (1995), no. 73, 191–201.

[Poi85] H. Poincaré, *Sur les courbes définies par les équations différentielles (iii)*, Journal de mathématiques pures et appliquées 4e série **1** (1885), 167–244.

[RiS15] David Romero i Sánchez, *Numerical computation of invariant objects with wavelets*, Ph.D. thesis, Universitat Autònoma de Barcelona, 2015.

[RT71] David Ruelle and Floris Takens, *On the nature of turbulence*, Les rencontres physiciens-mathématiciens de Strasbourg-RCP25 **12** (1971), 1–44.

[Saa03] Yousef Saad, *Iterative methods for sparse linear systems*, second ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003. MR 1990645 (2004h:65002)

[Sta97] Jaroslav Stark, *Invariant graphs for forced systems*, Phys. D **109** (1997), no. 1-2, 163–179, Physics and dynamics between chaos, order, and noise (Berlin, 1996). MR 1605989 (99c:58103)

[Sta99] ______, *Regularity of invariant graphs for forced systems*, Ergodic Theory Dynam. Systems **19** (1999), no. 1, 155–199. MR 1677161 (99k:58140)

[Ste70] Elias M Stein, *Singular integrals and differentiability properties of functions*, Princeton university press, 1970.

[SV06] T. M. Seara and J. Villanueva, *On the numerical computation of diophantine rotation numbers of analytic circle maps*, Physica D (2006), no. 217, 107–120.

[Tri83] Hans Triebel, *Theory of function spaces*, Monographs in Mathematics, no. 78, Birkhäuser, 1983.

[Tri92] ———, *Theory of function spaces ii / hans triebel*, Monographs in mathematics ; 84, Birkhäuser, Basel [etc, 1992 (eng).

[Tri06] ———, *Theory of function spaces. III*, Monographs in Mathematics, vol. 100, Birkhäuser Verlag, Basel, 2006. MR MR2250142 (2007k:46058)

[Tri10] ———, *Bases in function spaces, sampling, discrepancy, numerical integration*, EMS Tracts in Mathematics, vol. 11, European Mathematical Society (EMS), Zürich, 2010. MR 2667814 (2011j:46001)

[Vel88] M. Van Veldhuizen, *On the numerical approximation of the rotation number*, Journal of Computational and Applied Mathematics (1988), no. 21, 203–212.