Numerical computation of invariant objects with wavelets

Lluís Alsedà

in collaboration with D. Romero

Departament de Matemàtiques
Universitat Autònoma de Barcelona

http://www.mat.uab.cat/∼alseda

GDHAM15 Santuari de Núria, Girona, 28 June - 4 July 2015

UAB Universitat Autònoma de Barcelona

DEPARTAMENT DE **MATEMÀTIQUES**

Outline

Motivation

We are interested in studying complicate objects semianalitically (obtaining expansions in a truncated base) to be able to predict and understand changes in the geometry or dynamical properties as reducibility and others.

To fix ideas let us describe a couple of models that we use as toy models.

We consider skew products of the form

$$\begin{cases} \theta_{n+1} &= R(\theta_n) = \theta_n + \omega \pmod 1, \\ x_{n+1} &= T(\theta, x) \end{cases} \tag{1}$$

where $x \in \mathbb{R}^+, \theta \in \mathbb{S}^1$, $\omega \in \mathbb{R} \setminus \mathbb{Q}$ and $T(\theta, x)$ is of the form either $f(x)g(\theta)$ or $f(x) + g(\theta)$.

The [GOPY]-Keller model

In the system (1) we take $T(\theta, x) = f(x)g(\theta)$ with

1. $f : [0, \infty) \longrightarrow [0, \infty) \in \mathcal{C}^1$, bounded, strictly increasing, strictly concave and verifying $f(0) = 0$ (to fix ideas take $f(x) = 2\sigma \tanh(x)$ with $\sigma > 0$ as in the [GOPY] model). Thus, $x = 0$ will be invariant.
2. $g : \mathbb{S}^1 \longrightarrow [0, \infty)$ bounded and continuous (to fix ideas take $g(\theta) = \varepsilon + |\cos(2\pi\theta)|$ in a similar way to the [GOPY] model – except for ε and the absolute value).

We get:

$$\begin{cases} \theta_{n+1} &= \theta_n + \omega \pmod 1, \\ x_{n+1} &= 2\sigma \tanh(x_n)(\varepsilon + |\cos(2\pi\theta_n)|) \end{cases} \tag{2}$$

$\omega = \frac{\sqrt{5}+1}{2}$, $\sigma > 0$ and $\varepsilon \geq 0$.

The [GOPY]-Keller model

Remark

The attractor of the above system (if it exists) will be pinched if and only if $\varepsilon = 0$.

Pinching

There are big differences between the cases when g takes the value 0 at some point: the *pinched* case and the case when g is strictly positive.

In the pinched case any T–invariant set has to be 0 on a point and hence on a dense set because the circle $x \equiv 0$ is invariant and the θ-projection of every invariant object must be invariant under R.

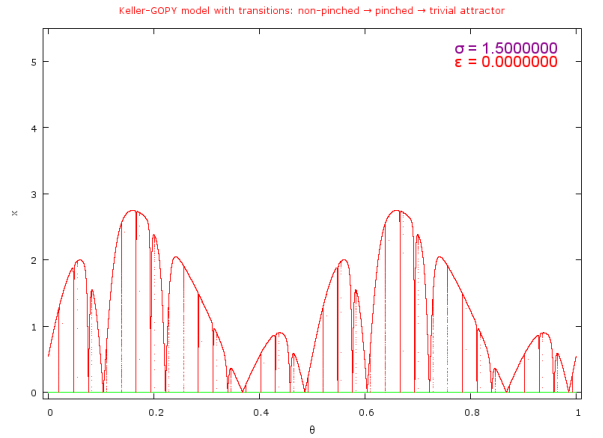
The [GOPY]-Keller model

The following theorem due to Keller [Kel] makes the above informal ideas rigorous. Before stating it we need to introduce the constant σ:

Since the line $x = 0$ is invariant, by using Birkhoff Ergodic Theorem, it turns out that

$$\sigma := f'(0) \exp\left(\int_{\mathbb{S}^1} \log g(\theta) d\theta \right) < \infty.$$

is the vertical Lyapunov exponent on the circle $x = 0$.



▸ Movie: A family of Keller Attractors

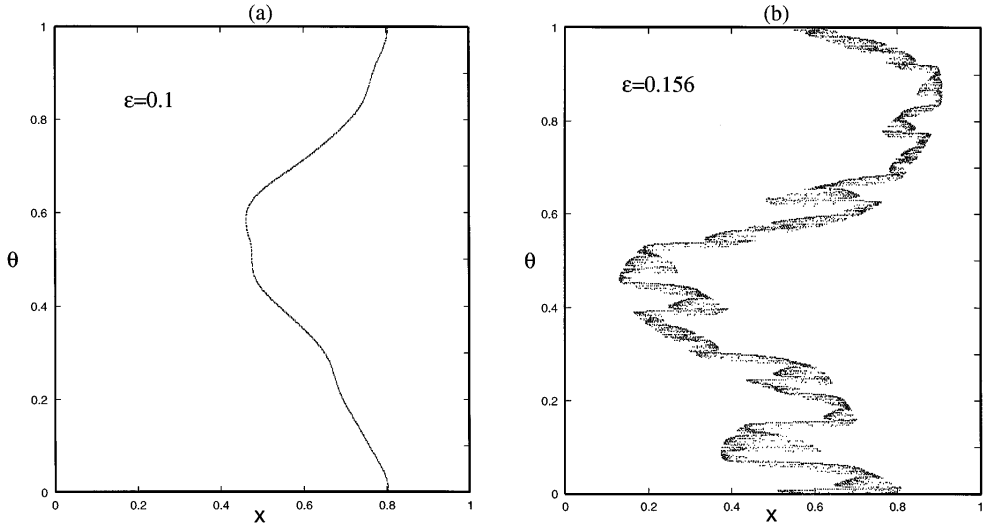
Keller Theorem

There exists an upper semicontinuous map $\phi\colon \mathbb{S}^1 \longrightarrow [0,\infty)$ whose graph is invariant under the Model (2). Moreover,

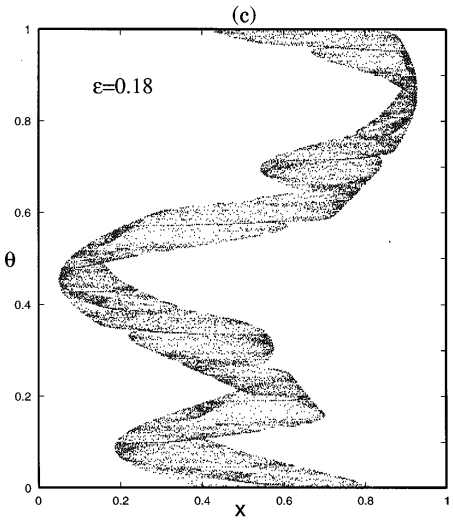
1. The Lebesgue measure on the circle, lifted to the graph of ϕ is a Sinai-Ruelle-Bowen measure,
2. if $\sigma \leq 1$ then $\phi \equiv 0$,
3. if $\sigma > 1$ then $\phi(\theta) > 0$ for almost every θ,
4. if $\sigma > 1$ and $g(\theta_0) = 0$ for some θ_0 then the set $\{\theta\colon \phi(\theta) > 0\}$ is meager and ϕ is almost everywhere discontinuous,
5. if $\sigma > 1$ and $g > 0$ then ϕ is positive and continuous; if g is $\mathcal{C}^1$ then so is ϕ,
6. if $\sigma \neq 1$ then $|x_n - \phi(\theta_n)| \to 0$ exponentially fast for almost every θ and every $x > 0$.

For this model we want to compute the attractor so that we can detect the pinching point.

The Nishikawa-Kaneko model. Plots with $a = 3.0$ (taken from the original paper)



$$\begin{cases} \theta_{n+1} &= \theta_n + \omega \pmod 1, \\ x_{n+1} &= a x(1-x) + \varepsilon \sin(2\pi\theta_n) \end{cases} \qquad (3)$$



where $x \in [0,1]$ and $\omega = \frac{\sqrt{5}+1}{2}$.

The Nishikawa-Kaneko model

In the Nishikawa-Kaneko paper it is described the *fractalization route* seen (that they illustrate in the pictures) with support on rough numerical computations.

Currently there is a strong debate about this route and about the fact that what we get at the end is really a fractal.

For this model we want to compute the attractors so that we can approximate their regularities and, perhaps, helping in deciding whether the final object is a fractal.

On the use of wavelets

As we have seen the invariant objects that we want to compute are expressed as graphs of functions (from $\mathbb{S}^1$ to $\mathbb{R}$).

The standard approach to compute invariant objects is to use finite Fourier approximations (trigonometric polynomials) to get expansions as:

$$\mathcal{F}(\theta) = a_0 + \sum_{n=1}^{N} \left(a_n \cos(n\theta) + b_n \sin(n\theta)\right).$$

However, the regularity and periodicity of the trigonometric basis makes clear that this approach is too complicate (or can be use at a very high cost) since, as we have seen, the topology and geometry of these objects is extremely complicate.

On the use of wavelets

In this case it seems more natural to use wavelets that adapt much better to oscillatory, irregular and highly discontinuous objects.

Our aim is to devise an algorithm to compute *massive* finite wavelet approximations for attractors with complicate geometry. We need these massive approximations because we want to be able to compute the regularities of the objects and this, as we will see, requires a massive computation of wavelet coefficients.

A (short) crash course on wavelets and regularity

Let us start by the definition of Multiresolution Analysis (MRA)

Definition

A sequence of closed subspaces of $\mathscr{L}^2(\mathbb{R})$, $\{\mathcal{V}_j\}_{j\in\mathbb{Z}}$, is a *Multiresolution Analysis* if it satisfies:

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

A (short) crash course on wavelets and regularity

Consider the bi-indexed family of maps obtained by dilations and translations of $\phi(x)$:

$$\phi_{j,n}(x) = \frac{1}{\sqrt{2^j}} \phi\left(\frac{x - 2^j n}{2^j}\right).$$

It can be shown that

1. $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$ is an orthonormal basis of $\mathcal{V}_j$ for each $j \in \mathbb{Z}$, and
2. $\phi(x)$ characterizes the whole MRA (see [Mal]).

[Mal] Mallat, Stéphane, *A wavelet tour of signal processing*, Academic Press Inc., San Diego, CA, 1998, xxiv+577.

A (short) crash course on wavelets and regularity

If we fix an MRA, we know that $\mathcal{V}_j \subset \mathcal{V}_{j-1}$. Then we define the subspace *$\mathcal{W}_j$ as the orthogonal complement of $\mathcal{V}_j$ on $\mathcal{V}_{j-1}$*.
That is

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

We are looking for an orthonormal basis of $\mathcal{W}_j$, *which verifies a relation with $\phi(x)$*, (the *wavelets*). This basis is given from a function called the *mother wavelet* $\psi(x)$ by the formula

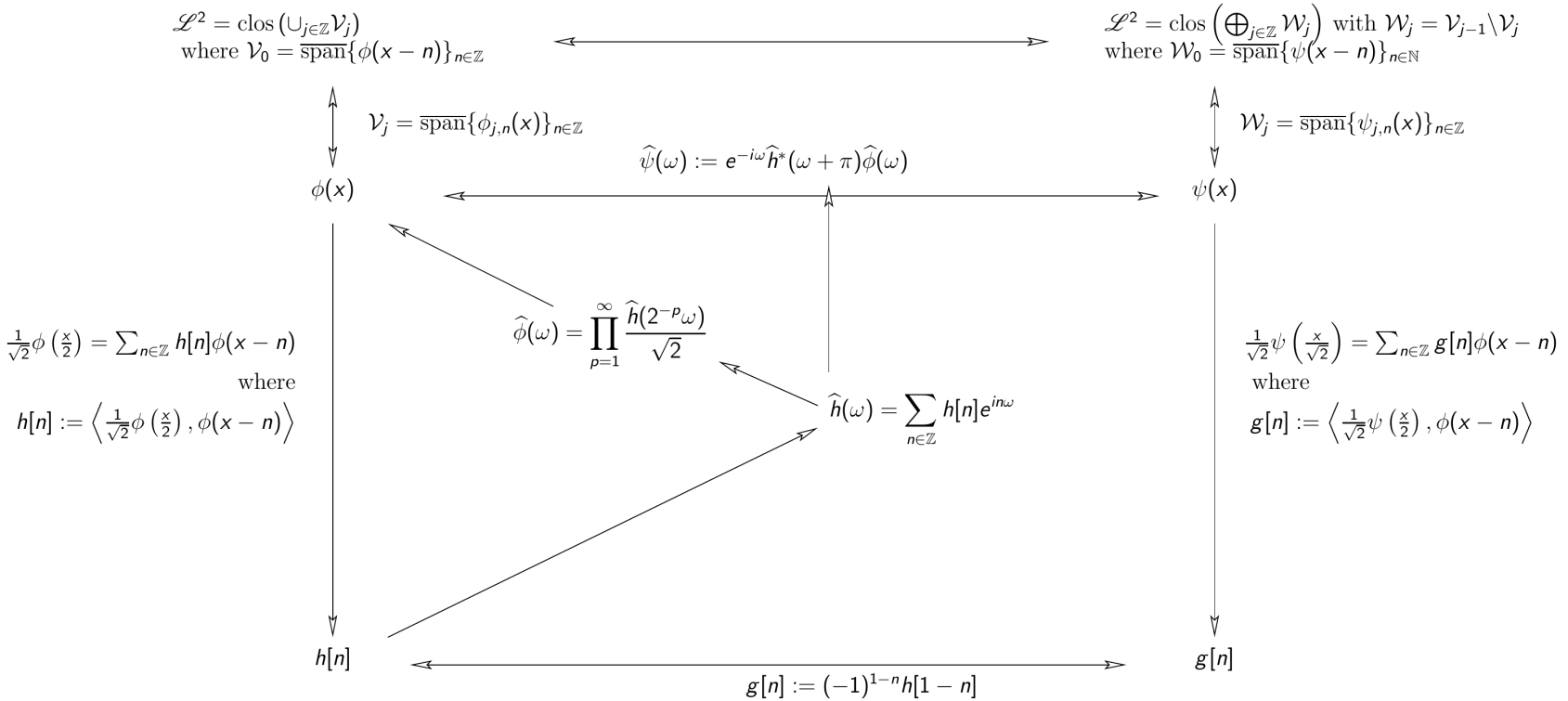
$$\psi_{j,n}(x) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{x - 2^j n}{2^j}\right).$$

In [Mal] it is shown that:

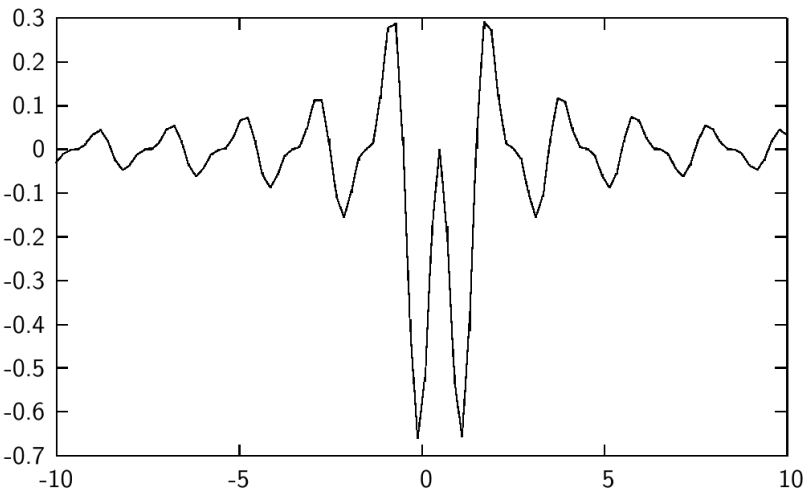
Mallat and Meyer Theorem

- For every $j \in \mathbb{Z}$ the family $\{\psi_{j,n}\}_{n\in\mathbb{Z}}$ is an orthonormal basis of each $\mathcal{W}_j$,
- The wavelets $\{\psi_{j,n}\}_{(j,n)\in\mathbb{Z}\times\mathbb{Z}}$ are an orthonormal basis of $\mathscr{L}^2(\mathbb{R})$ for all $j, n \in \mathbb{Z}$.

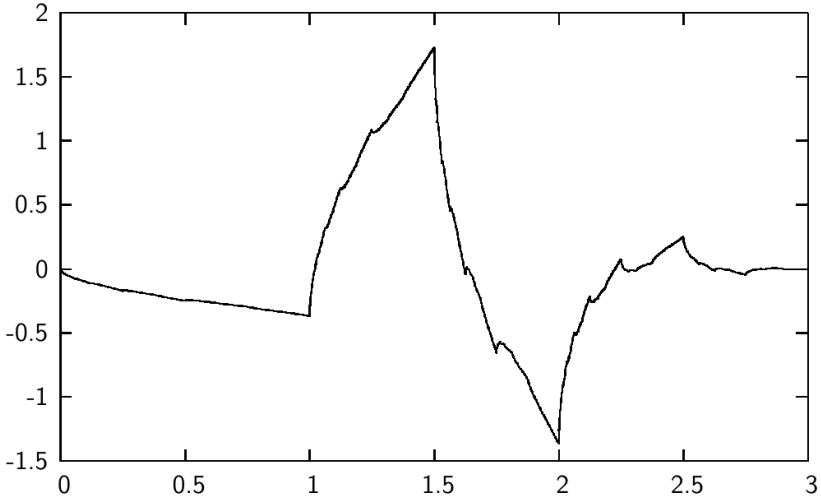
A (short) crash course on wavelets and regularity
How to compute the mother wavelet



A (short) crash course on wavelets and regularity
Examples of mother wavelets



Shannon wavelet (no compact support)



Daubechies wavelet (compact support)

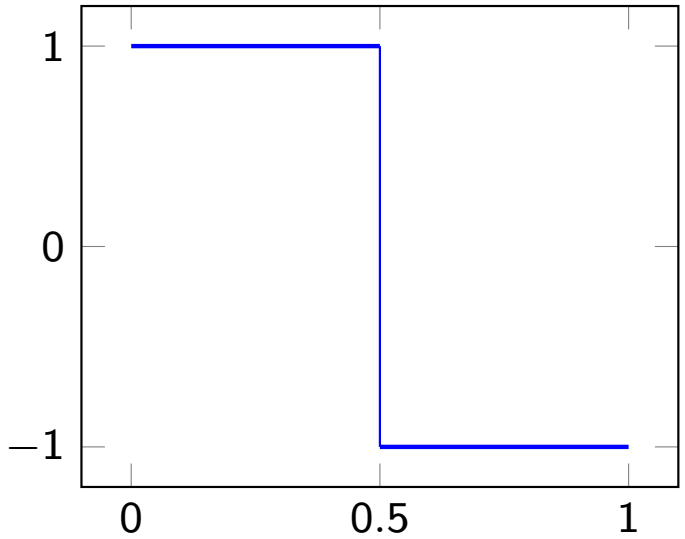
$$\psi(x) = \frac{\sin(2\pi(x-1/2))}{2\pi(x-1/2)} - \frac{\sin(\pi(x-1/2))}{\pi(x-1/2)}$$

$$h[n] = \begin{cases} \frac{\sqrt{2}}{2} & \text{if } n = 0, \\ \sqrt{2}\frac{-1^{(n-1)/2}}{\pi n} & \text{if } n \text{ odd}, \\ 0 & \text{otherwise.} \end{cases}$$

No closed formula

$$h[n] = \begin{cases} 0.4829629131445341\ldots & \text{if } n = 0, \\ 0.8365163037378079\ldots & \text{if } n = 1, \\ 0.2241438680420134\ldots & \text{if } n = 2, \\ -0.1294095225512604\ldots & \text{if } n = 3, \\ 0 & \text{otherwise.} \end{cases}$$

A (short) crash course on wavelets and regularity
Examples of mother wavelets



Haar wavelet (compact support)

$$\psi(x) := \mathbf{1}_{[0,\frac{1}{2})}(x) - \mathbf{1}_{[\frac{1}{2},1)}(x) \quad \text{where} \quad \mathbf{1}_{[a,b)}(x) = \begin{cases} 1 & \text{if } x \in [a,b), \\ 0 & \text{otherwise.} \end{cases}$$

A (short) crash course on wavelets and regularity
Wavelet expansions

We know that given a function $f \in \mathscr{L}^2(\mathbb{R})$ and an MRA, f can be expanded in the wavelet basis:

$$f(x) = \sum_{j\in\mathbb{Z}}\sum_{n\in\mathbb{Z}} \langle f, \psi_{j,n}\rangle \psi_{j,n}(x) = \sum_{j\in\mathbb{Z}}\sum_{n\in\mathbb{Z}} d_j[n]\psi_{j,n}(x)$$

We look for finite (truncated) wavelet approximations of f of the type:

$$f \sim \sum_{j=0}^{J}\sum_{n=0}^{2^j-1} \langle f, \psi_{j,n}\rangle \psi_{j,n} = \sum_{j=0}^{J}\sum_{n=0}^{2^j-1} d_j[n]\psi_{j,n}(x)$$

To obtain such approximations one uses the Fast Wavelet Transform (FWT).

A (short) crash course on wavelets and regularity
The FWT

With $J > 0$ fixed and an MRA $\{\mathcal{V}_j\}_{j\in\mathbb{Z}}$, we can decompose

$$\mathscr{L}^2(\mathbb{R}) = \mathcal{V}_{-J} \oplus \left(\mathscr{L}^2(\mathbb{R})\backslash\mathcal{V}_{-J}\right).$$

That is, we can write the projection of f to $\mathcal{V}_{-J}$ as

$$\mathrm{P}_{\mathcal{V}_{-J}}(f) = \sum_{n\in\mathbb{Z}} \langle f, \phi_{-J,n}\rangle \phi_{-J,n} \in \mathcal{V}_{-J}$$

and

$$f = \mathrm{P}_{\mathcal{V}_{-J}}(f) + (f - \mathrm{P}_{\mathcal{V}_{-J}}(f)),$$

with $(f - \mathrm{P}_{\mathcal{V}_{-J}})(f) \in \left(\mathscr{L}^2(\mathbb{R})\backslash\mathcal{V}_{-J}\right)$.

A (short) crash course on wavelets and regularity
The FWT

Then we can truncate $\mathrm{P}_{\mathcal{V}_{-J}}(f)$ to the finite dimensional version of $\mathcal{V}_{-J}$ to get

$$f \sim \sum_{n=0}^{2^J-1} \langle f, \phi_{-J,n} \rangle \phi_{-J,n} = \sum_{n=0}^{2^J-1} a_{-J}[n] \phi_{-J,n}.$$

But, recall that $\mathcal{V}_{-J} = \mathcal{V}_{-J+1} \oplus \mathcal{W}_{-J+1}$. Therefore,

$$\begin{aligned} f &\sim \sum_{n=0}^{2^J-1} a_{-J}[n] \phi_{-J,n} \\ &= \sum_{n=0}^{2^{J-1}-1} \langle f, \phi_{-J+1,n} \rangle \phi_{-J+1,n} + \sum_{n=0}^{2^{J-1}-1} \langle f, \psi_{-J+1,n} \rangle \psi_{-J+1,n} \\ &= \sum_{n=0}^{2^{J-1}-1} a_{-J+1}[n] \phi_{-J+1,n} + \sum_{n=0}^{2^{J-1}-1} d_{-J+1}[n] \psi_{-J+1,n}. \end{aligned}$$

A (short) crash course on wavelets and regularity
The FWT

By iterating this process we get:

$$f \sim \phi_{0,0} + \sum_{j=0}^{J} \sum_{n=0}^{2^j-1} \langle f, \psi_{j,n} \rangle \psi_{j,n} = \phi_{0,0} + \sum_{j=0}^{J} \sum_{n=0}^{2^j-1} d_{-J+j}[n] \psi_{j,n}(x)$$

with $\phi_{0,0} = \langle f, \phi \rangle$. This will give a good approximation for f provided that J is big enough (so $\phi_{0,0} + \sum_{n=0}^{2^J-1} a_{-J}[n] \phi_{-J,n}$ is a sufficiently good approximation of f).

But additionally we need a formula to compute the coefficients $a_{j+1}[n]$ and $d_{j+1}[n]$ from the coefficients $a_j[n]$.

A (short) crash course on wavelets and regularity
The FWT

This formula is given by the

Mallat Theorem (see [Mal])

Let $\{\mathcal{V}_j\}_{j\in\mathbb{Z}}$ be an MRA and let $a_j[n] := \langle f, \phi_{j,n}\rangle$ denote the *scaling coefficients* and let $d_j[n] := \langle f, \psi_{j,n}\rangle$ denote the *wavelet coefficients*. Then, the following recursive formulas hold.

- At the *decomposition*:

$$a_{j+1}[p] = \sum_{n\in\mathbb{N}} h[n-2p]a_j[n] \text{ and } d_{j+1}[p] = \sum_{n\in\mathbb{N}} g[n-2p]a_j[n].$$

- At the *reconstruction*:

$$a_j[p] = \sum_{n\in\mathbb{N}} h[p-2n]a_{j+1}[n] + \sum_{n\in\mathbb{N}} g[p-2n]d_{j+1}[n].$$

A (short) crash course on wavelets and regularity
The FWT

To start the FWT we need the scaling coefficients $a_J[n]$. Under certain *reasonable* conditions, which includes that f is Lipschitz, in [Fra] it is shown:

Lemma

$$|\langle f, \phi_{j,n}\rangle - 2^{-j/2} f(2^j n)| < C 2^{-j(\alpha+1/2)}.$$

Therefore,

$$2^{J/2} a_J[n] \approx f(2^J n).$$

[Fra] Frazier, Michael W., *An introduction to wavelets through linear algebra*, Springer, 1999.

A (short) crash course on wavelets and regularity
Summarizing the FWT

1. Fix a J large enough such that $\sum_{n=0}^{2^J-1} a_{-J}[n]\phi_{-J,n}$ is a good approximation of f and $\mathrm{C}2^{-J(\alpha+1/2)}$ is small enough.
2. With such J, take $f(2^J n)$ to be an approximation of $2^{J/2}a_J[n]$
3. Apply

$$a_{j+1}[p] = \sum_{n\in\mathbb{N}} h[n-2p]a_j[n] \text{ and } d_{j+1}[p] = \sum_{n\in\mathbb{N}} g[n-2p]a_j[n].$$

Computing regularities

We start by defining the regularity spaces that we will use

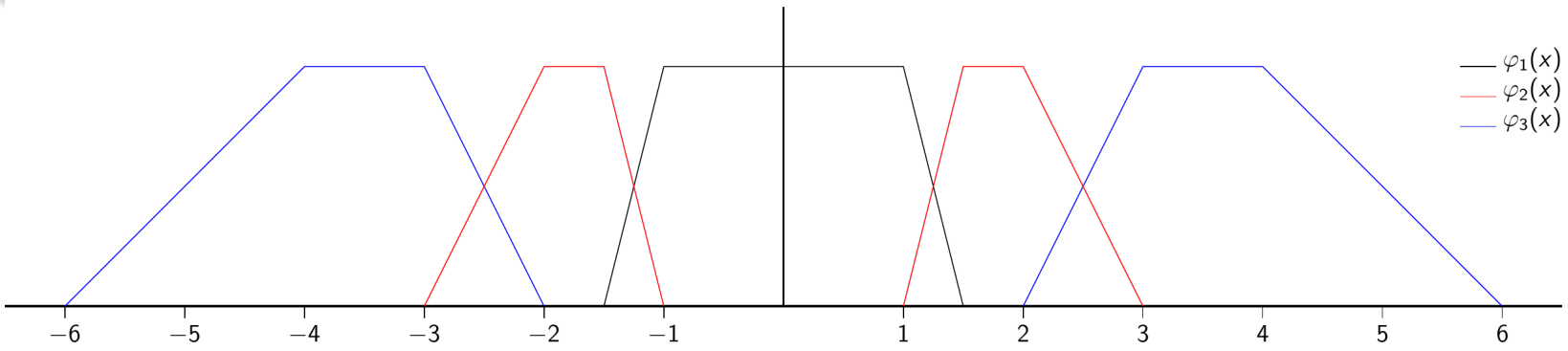
Definition

Let $\varphi = \{\varphi_j\}_{j=0}^{\infty}$ be a *dyadic resolution of unity* (see the figure below). We define the *Besov* spaces $\mathcal{B}^s_{\infty,\infty}$ with $s \in \mathbb{R}$ by

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

with

$$\|f\|_{\infty,\infty,s} = \sup_{j\geq 0} \; 2^{js} \left(\|(\varphi_j \widehat{f})^{\vee}\|_{\infty} \right).$$



The graph of some $\varphi_j(x)$ from φ.

Computing regularities

From [Tri2] and [Coh]

Theorem

Let $s \in \mathbb{R} \setminus \{0\}$ and let ψ be a mother Daubechies wavelet with more than $\max(s, 5/2 - s)$ vanishing moments. Then $f \in \mathcal{B}^s_{\infty,\infty}$ if and only if there exists $C > 0$ such that

$$\sup_{n\in\mathbb{Z}} |\langle f, \psi_{j,n}\rangle| \leq C 2^{\tau j} \qquad \text{with} \qquad \tau = \begin{cases} s + \frac{1}{2} & \text{if } s > 0, \\ s - \frac{1}{2} & \text{if } s < 0, \end{cases}$$

for all $j \leq 0$.

We will use this result to estimate the regularity of the Keller-GOPY attractor.

[Coh] Cohen, Albert, *Numerical analysis of wavelet methods*, North-Holland, 2003.

Using the FWT to compute wavelet coefficients

we consider the following parametrization of the system (2):

$$\begin{cases} \theta_{n+1} &= \theta_n + \omega \pmod 1, \\ x_{n+1} &= 2\sigma \tanh(x_n)(\varepsilon(\sigma) + |\cos(2\pi\theta_n)|) \end{cases} \tag{4}$$

with $\omega = \frac{\sqrt{5}+1}{2}$, $\sigma \in [1, 2]$ and

$$\varepsilon(\sigma) = \begin{cases} (\sigma - 1.5)^2 & \text{when } 1.5 \leq \sigma \leq 2, \\ 0 & \text{when } 1 \leq \sigma \leq 1.5. \end{cases}.$$

In this way the system is pinched if and only if $\sigma \leq 1.5$.

Following de la Llave and Petrov, take a Daubechies wavelet with $k > \max\left(s, \frac{5}{2} - s\right)$ vanishing moments and fix a positive integer J.

Using the FWT to compute wavelet coefficients

To compute an estimate of the Hölder exponent of the attractor we will perform the following steps:

Step 0 Approximate the attractor by iterating with a transient to obtain a signal $(\theta_i, \lambda(\theta_i))$ (use *Simo's principle*: if we have an attractor let it work (attract) for us). Since Keller Theorem says that we have exponential contraction the approximation will be moderately good.

Step 1 Calculate $a_{-J}[n] \approx \langle \lambda, \phi_{-J,n} \rangle$, where $0 \leq n \leq 2^J - 1$.

Step 2 Calculate, by means of the FWT the coefficients

$$d_j[n] = \langle \lambda, \psi_{j,n} \rangle$$

where $0 \leq j \leq J$ and, for each, j $0 \leq n \leq 2^j - 1$.

Using the FWT to compute wavelet coefficients

Step 3 For $0 \leq j \leq J$, calculate

$$s_j = \log_2 \left(\sup_{0 \leq n \leq 2^j - 1} |d_j[n]| \right).$$

Step 4 Make a linear regression to estimate the slope τ of the graph of the pairs (j, s_j) with $j = 0, -1, -2, \ldots, -J$ and use the above theorem to get s provided that the wavelet used had more than $\max(s, 5/2 - s)$ vanishing moments.

This algorithm gives an effective way of computing wavelet coefficients and regularities. It has only two problems that must be solved:

Using the FWT to compute wavelet coefficients

Remark

Steps 3 and 4 before justify why we need massive computation on wavelets coefficients. To have J points in the above regression we need 2^{J+1} coefficients. Conversely, with 2^{30} coefficients we only have 29 values to estimate the regularity.

Using the FWT to compute wavelet coefficients: the problems

1. The function λ which defines the attractor is far from being Lipschitz. Thus, the estimate $a_{-J}[n] \approx \langle \lambda, \phi_{-J,n} \rangle$ in principle is not valid. However, using ideas from the proof of Keller Theorem and the dominated Convergence Theorem it can be shown that the approximation is valid provided that the transient in the computation of the algorithm is big enough.
2. The points θ_i that give the attractor are, a priori not equispaced. This can be solved by conjugating the attractor with a diffeomorphism of class $\mathcal{C}^2$ to a version of the attractor with points equispaced and sorting the signal to get the values $\lambda(\theta_i)$ in the right ordering. The conjugacy is not a problems since by using a theorem from de la Llave and Obaya one can prove that the regularity of both attractors is the same.

Using the FWT to compute wavelet coefficients

With these tricks we get the following regularity graph:

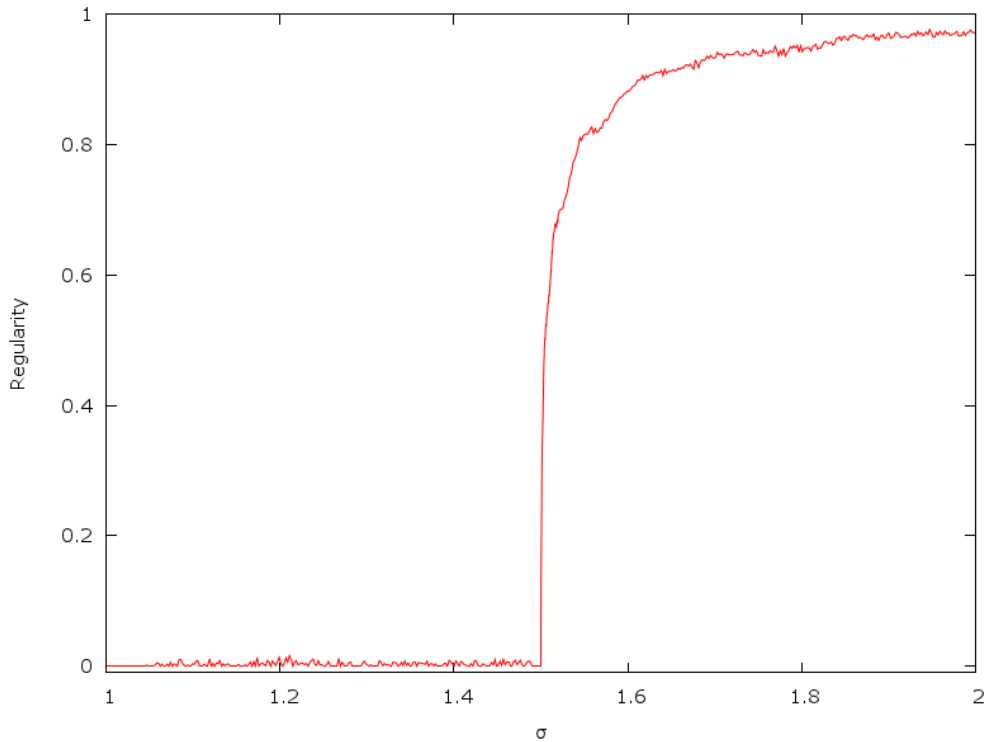


Figure: The estimate of the regularity of the (strange) attractor of System (2) for $\sigma \in [1, 2]$ and ε given by the parametrization $\varepsilon(\sigma)$. The results are obtained by using a sample of 2^{30} points (that is, $J = 30$), a transient $N_0 = 10^5$ and the Daubechies Wavelet with 16 vanishing moments.

Using the FWT to compute wavelet coefficients

This algorithm detects in a correct way the pinching point despite of the fact that the regularities for $\sigma \gtrapprox 1.5$ are wrong, as the following lemma shows:

Lemma

The upper semi continuous function $\lambda\colon \mathbb{S}^1 \longrightarrow \mathbb{R}^+$ whose graph is in φ is in $\mathscr{B}^0_{\infty,\infty}(\mathbb{S}^1)$ when $\varepsilon = 0$ and $\mathscr{B}^s_{\infty,\infty}(\mathbb{S}^1)$, with $s \in (0,1]$ when $\varepsilon > 0$.

This also explains why we have to use the Besov spaces instead of the Hölder ones: to allow regularity zero that is not attainable with Hölder spaces.

The problem in this algorithm comes from the fact that the initial signal (the attractor) does not have enough quality (precision) due to the extremely complicate geometry

Using the the invariance equation to compute wavelet coefficients

To overcome the above method its better to use the invariance equation to fix the attractor and solve numerically the problem.

Also we will use Haar wavelets that have better properties than Daubechies ones.

The invariance equation

The study of the above systems is based on the iteration of the *Transfer Operator*. Let $\mathscr{P}$ be the space of all functions (not necessarily continuous) from $\mathbb{S}^1$ to $\mathbb{R}$. If we look for a functional version of the System (2) in the space $\mathscr{P}$ one can define the *Transfer Operator* $\mathfrak{T}\colon \mathscr{P} \longrightarrow \mathscr{P}$ as

$$\mathfrak{T}(\varphi)(\theta) = f(\varphi(R_\omega^{-1}(\theta))) \cdot g(R_\omega^{-1}(\theta)).$$

Notice that the graph of a function $\varphi\colon \mathbb{S}^1 \longrightarrow \mathbb{R}$ is invariant for the System (2) if and only if

$$f(\varphi(R_\omega^{-1}(\theta))) \cdot g(R_\omega^{-1}(\theta)) = \mathfrak{T}(\varphi)(\theta) = \varphi(\theta)$$

or equivalently we get the *invariance equation*

$$f(\varphi(\theta)) \cdot g(\theta) = \varphi(R_\omega(\theta)).$$

The invariance equation

To solve the above functional equation we write the attractor as

$$\varphi(\theta) = \phi_{0,0} + \sum_{j=0}^{J} \sum_{n=0}^{2^j-1} d_j[n]\psi_{j,n}(\theta)$$

where the coefficients $\phi_{0,0}$ and $d_j[n]$ are the unknowns.

To make this formula easier we set $N = 2^{J+1}$, $\phi_{0,0} = d_0$ and we set $\ell = \ell(j,n) = 2^j + n$, which gives a linear indexing of $d_j[n]$ and $\psi_{j,n}(\theta)$. With this notation the formula becomes:

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

The invariance equation

As usual we plug this expression into the invariance equation and we get:

$$d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell(R_\omega(\theta)) = f\left(d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell(\theta)\right) \cdot g(\theta).$$

To be able to solve it we discretize the variable θ into N dyadic points in the circle $\theta_i = \frac{i}{N}$ for $i = 0, 1, \ldots, N-1$ and we impose that the invariance equation is verified on the dyadic points θ_i. We get

$$d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell(R_\omega(\theta_i)) - f\left(d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell(\theta_i)\right) \cdot g(\theta_i) = 0$$

for every $i = 0, 1, \ldots, N-1$.

Thus we get a non-linear system of N equations with N unknowns.

The invariance equation

We will use the Newton Method to solve this system.

To do this we are facing two problems:

- Find a good initial seed
- Solve a huge linear system that, as we will see, is ill conditioned.

The solution to the first problem is easy. We find an approximation to the attractor *at the prescribed points* θ_i (for this we have to start at the appropriate pre-image of each θ_i) and compute the wavelet coefficients. This process need not be very precise since this will not give the final solution but it will give a good initial seed.

The Jacobian matrix

The solution to the second problem consists in understanding well the properties of the values $\psi_\ell(R_\omega(\theta_i))$ and $\psi_\ell(\theta_i)$.

The Jacobian of the system, denoted $\Phi = (\Phi_{i,\ell})$ is

$$\Phi_{i,\ell} = \begin{cases} 1 - f'\left(d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell(\theta_i)\right) \cdot g(\theta_i) & \text{if } \ell = 0, \\ \psi_\ell(R_\omega(\theta_i)) - f'\left(d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell(\theta_i)\right) \cdot g(\theta_i) \cdot \psi_\ell(\theta_i) & \text{if } \ell \neq 1, \end{cases}.$$

The Jacobian matrix

It is useful to have the Jacobian of the system in matrix form. So, we introduce some vectors and matrices.

- $\delta := (d_0, d_1, \ldots, d_{N-1})^\top$
- $\Psi = (\Psi_{i,\ell})$ is the matrix $N \times N$ such that

$$\Psi_{i,\ell} = \begin{cases} 1 & \text{if } \ell = 0 \\ \psi_\ell(\theta_i) & \text{if } \ell \neq 0 \end{cases}$$

- $\Psi_R = \left(\Psi^R_{i,\ell}\right)$ is the matrix such that

$$\Psi^R_{i,\ell} = \begin{cases} 1 & \text{if } \ell = 0 \\ \psi_\ell(R_\omega(\theta_i)) & \text{if } \ell \neq 0 \end{cases}$$

- Λ is the diagonal matrix that has $f'\left([\Psi\delta]_0\right) \cdot g(\theta_0), f'\left([\Psi\delta]_1\right) \cdot g(\theta_1), \ldots, f'\left([\Psi\delta]_{N-1}\right) \cdot g(\theta_{N-1})$ as the elements in the diagonal, where $[\alpha]_\ell$ denotes the component ℓ of the vector α.

The Jacobian matrix

Observe that

$$[\Psi\delta]_i = d_0 + \sum_{\ell=1}^{N-1} d_\ell \psi_\ell(\theta)\varphi(\theta_i) = \varphi(\theta_i)$$

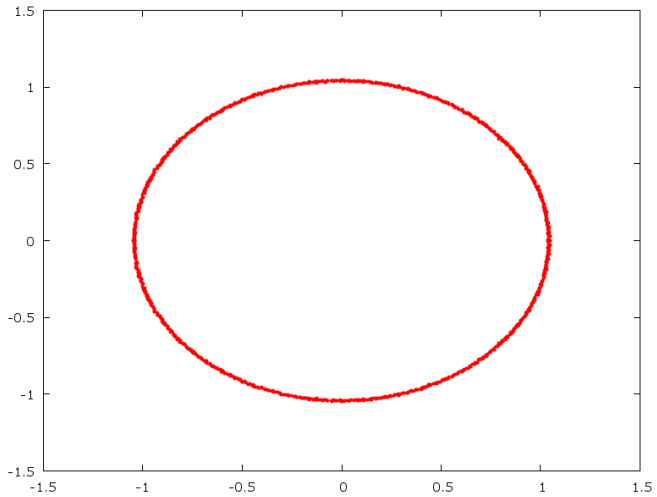
With all this notation the Jacobian matrix can be written as

$$\Phi = \Psi_R - \Lambda\Psi.$$

At each iteration of the Newton Method we have to solve the linear system with Φ as a matrix and whose independent term is the evaluation of the invariance equation.

The Jacobian matrix

In the figure below we show a typical spectrum of this matrix which explains why all standard methods to solve the system fail.



There is a good solution to this problem which is *preconditioning*. But for this we need to know better the wavelet matrices Ψ_R and Ψ

The wavelet matrices

First we consider the case of Haar wavelets defined before.

Lemma (Properties of the Haar wavelet matrices)

The following statement hold:

1. The wavelets matrix is very simple:

$$\psi_{j,n}(\ell/N) = \begin{cases} \frac{1}{\sqrt{N}} 2^{-j/2} & \text{for } 0 \leq \ell - ns < s/2, \\ -\frac{1}{\sqrt{N}} 2^{-j/2} & \text{for } s/2 \leq \ell - ns < s, \\ 0 & \text{if } \ell - ns \geq 0. \end{cases}$$

   with $s = 2^{J-j}$.

2. The matrix Ψ is orthogonal.
3. Set $r = \lfloor \omega N \rfloor$ and let $P = (p_{i,j})$ be the permutation matrix such that $p_{i,j} = 1$ if and only if $j = i + r \pmod N$. Then, $\boxed{\Psi_R = P\Psi}$.

The Jacobian matrix revisited: preconditioning

We have to solve (many times) the system

$$(\Psi_R - \Lambda\Psi)z = b$$

where b is the evaluation of the invariance equation.

Notice that

$$\Psi\Psi_R^\top = \Psi(P\Psi)^\top = \Psi\Psi^\top P^\top = P^\top,$$

$$\Psi_R\Psi_R^\top = P\Psi\Psi^\top P^\top = PP^\top = \mathrm{Id}\,.$$

Hence,

$$(\Psi_R - \Lambda\Psi)\Psi_R^\top = \mathrm{Id} - \Lambda P^\top$$

is the matrix that has 1's in the diagonal and the elements $\Lambda_{i,i}$ in the entry $i, i + N - r \pmod N$.

The Jacobian matrix revisited: preconditioning

So, if we make the change of variables $z = \Psi_R^\top y$. Then, the system $(\Psi_R - \Lambda\Psi)z = b$ becomes

$$(\mathrm{Id} - \Lambda P^\top)y = b.$$

By performing Gauss Method formally on the system we obtain an explicit recurrence that manages to solve the system in linear time with N. The only small difficulty is that there might be several dependence cycles in the substitution process of the y_i.

A word on the change of variables and its consequences

The previous change of variables suggest that we should do this change permanently and work with the *rotated wavelet coefficients* defined as $\boxed{c = \Psi_R \delta}$ (or equivalently $\delta = \Psi_R^\top c$).

Let us see that this has strong simplifying consequences.

A word on the change of variables and its consequences

Reconstruction

Above we have noted:

$$[\Psi\delta]_i = \varphi(\theta_i)$$

In other words,

$$\Psi\delta = (\varphi(\theta_0), \varphi(\theta_1), \ldots, \varphi(\theta_{N-1})^\top$$

allows us to get the attractor from the wavelet coefficients. But to get the attractor from the rotated wavelet coefficients we have

$$(\varphi(\theta_0), \varphi(\theta_1), \ldots, \varphi(\theta_{N-1})^\top = \Psi\delta = \Psi\Psi_R^\top c = P^\top c.$$

So, *the reconstruction is much easier done with the rotated coefficients than with the standard ones*.

A word on the change of variables and its consequences

Producing the initial seed

As we have said the initial seed is obtained by computing an approximation of the vector $(\varphi(\theta_0), \varphi(\theta_1), \ldots, \varphi(\theta_{N-1})^\top$. Since Ψ is orthogonal, we can get an initial seed for delta by

$$\delta = \Psi^\top \Psi \delta \approx \Psi^\top (\varphi(\theta_0), \varphi(\theta_1), \ldots, \varphi(\theta_{N-1})^\top.$$

But again, if we work with rotated coefficients we get

$$c = \Psi_R \delta \approx \Psi_R \Psi^\top (\varphi(\theta_0), \varphi(\theta_1), \ldots, \varphi(\theta_{N-1})^\top$$

and $\Psi_R \Psi^\top = (\Psi \Psi_R^\top)^\top = (P^\top)^\top = P$. Again, to obtain the initial seed *is much easier with the rotated coefficients than with the standard ones*.

A word on the change of variables and its consequences

The invariance equation with rotated coefficients

The invariance equation written algebraically is:

$$[\Psi_R \delta]_i - f\left([\Psi\delta]_i\right) \cdot g(\theta_i) = 0.$$

With rotated coefficients we have

$$\left[\Psi_R \Psi_R^\top c\right]_i - f\left(\left[\Psi \Psi_R^\top c\right]_i\right) \cdot g(\theta_i) = 0,$$

which is equivalent to

$$c_i - f\left(\left[P^\top c\right]_i\right) \cdot g(\theta_i) = 0,$$

And, once more, the invariance equation *is much easier with the rotated coefficients than with the standard ones*.

A word on the change of variables and its consequences

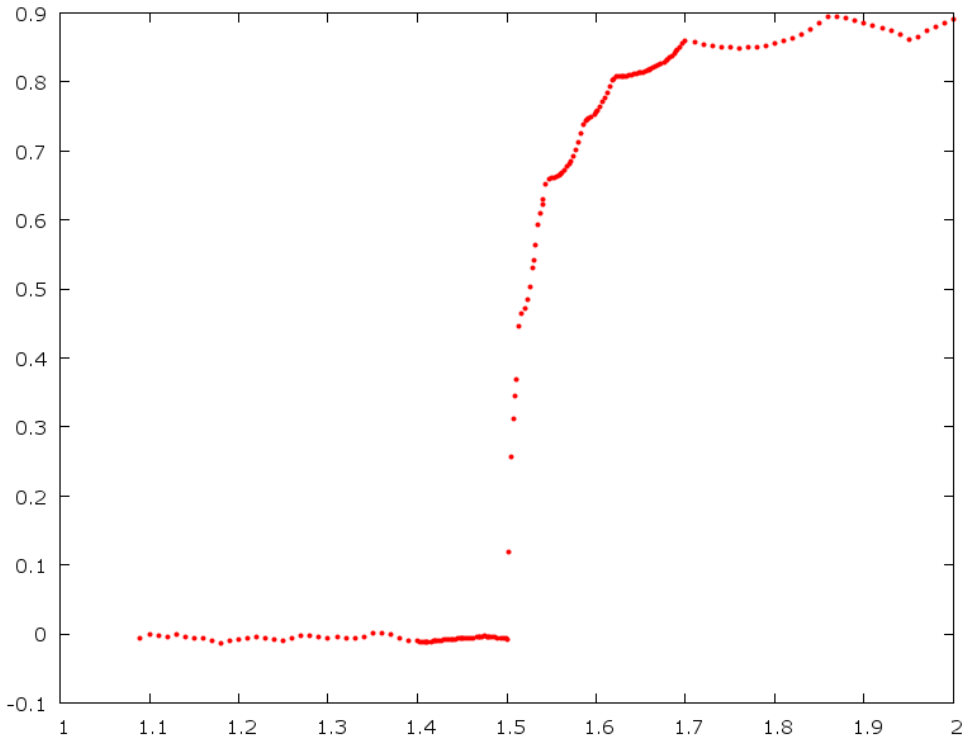
Implementing this algorithm leads to the following results in a pinched difficult case with Lyapunov exponent close to zero:

```
BigJ = 25; N = 33554432
\sigma = 1.090000
\epsilon = 0.0000000000000000
angle \omega = 0.61803398874989490253
Computation of \phi(theta_i = i/N): 1787.088863 CPU secs..
Computation of g(theta_i = i/N): 2.561167 CPU secs.
     ..................
Newton iterate 683. SupNorm of Inv. Eq. 1.1102230246251565e-16
Time of last Newton iterate = 4.280968"
Newton iterate 684. SupNorm of Inv. Eq. 0.0000000000000000e+00
```

These computations have been done with a single computer with and Intel i7-4770K QuadCore processor @ 3.50GHz with 32Gb of RAM DDR3 @ 1867MHz.

The Haar results

By a result analogous to Triebel Theorem quoted above but for Haar wavelets we get the following picture analogous to the previous one. The same comments apply



A word on Daubechies wavelets

- Daubechies Wavelets do not have closed form. This means that a specific iterative algorithm (Vidakovich algorithm) is necessary to compute the wavelet coefficients. This algorithm involves multiplying two matrices T_0 and T_1 according to the binary expansion of the fractional part on the point θ where the matrix has to be evaluated. This is very specific and difficult programming, since a high level of optimization is essential.
- Fortunately the wavelets matrices are sparse and self similar. This means that they can be stored saving a lot of memory. Again a very delicate programming is necessary. It is compulsory to develop specific data types.

A word on Daubechies wavelets

- **Bad news:** The nice equalities that we have for Haar Wavelets do not hold. Preconditioning works but the preconditioned Jacobian matrix is not so nice as the Haar one. It has the same properties but band-wise. So, the solution of the system is not iterative and is not linear on time with N. However GMRES-Arnoldi seem to work efficiently on the preconditioned system.