Wavelets and Signal Processing

François Chaplais
Systems and Control Center
École des Mines de Paris
Summary

Wavelet allow the analysis and manipulation of singularities, and, more generally, sharp transients in signals.
Plan

• Perfect reconstruction sub band coding
• Fourier background
• From discrete to continuous time
• Multiresolution approximations
• Orthogonal wavelets
• Wavelet analysis
• Other wavelets
• Applications
Decomposition and Reconstruction of Discrete Signals
Imagine that we wish to apply a different processing to the low and high frequency part of a discrete signal $a$.
Sub Band Filtering

Imagine that we wish to apply a different processing to the low and high frequency part of a discrete signal \( a \)

\[
\begin{align*}
a & \leftarrow \text{high pass filter} \quad \overleftrightarrow{ah} \\
& \quad \overleftrightarrow{al} \\
& \quad \text{low pass filter}
\end{align*}
\]
Sub Band Filtering

Now we do our processing

\[ a \leftarrow \text{high pass filter} \quad \rightarrow \quad ah \]

\[ a \leftarrow \text{low pass filter} \quad \rightarrow \quad al \]
Sub Band Filtering

Now we do our processing

\[ a \quad \xrightarrow{\text{high pass filter}} \quad ah \quad \xrightarrow{\text{HF processing}} \quad ahf \]
\[ \quad \xleftarrow{\text{low pass filter}} \quad al \quad \xrightarrow{\text{LF processing}} \quad alf \]
Reconstruction

Now that we are done with our processing, we want to recover a single signal for further usage.

\[ a \]

- high pass filter \(\rightarrow ah\)
- low pass filter \(\rightarrow al\)
Reconstruction

Now that we are done with our processing, we want to recover a single signal for further usage.
Consistency

We have two requirements in the previous scheme


**Consistency**

We have two requirements in the previous scheme:

- High pass filter: \( a \rightarrow ah \rightarrow HF \text{ processing} \rightarrow ahf \}
- Low pass filter: \( a \rightarrow al \rightarrow LF \text{ processing} \rightarrow alf \}

Reconstruction: \( ahf \rightarrow b \)
We have two requirements in the previous scheme

- If no processing is performed, we want to recover the original signal
- If some processing is performed, we want to be able to interpret the two channels hf and bf as the transform of a modified signal, and use this to reconstruct a composite signal
Consistency

Point 1 can be satisfied using suitable filters in the following scheme
Consistency

Point 1 can be satisfied using suitable filters in the following scheme

- \( a \rightarrow ah \rightarrow \text{HF filter} \rightarrow \text{high pass reconstruction filter} \)
- \( a \rightarrow al \rightarrow \text{LF filter} \rightarrow \text{low pass reconstruction filter} \)

\[ 1 \rightarrow b \]

However

if some processing is done on \( ah \) and \( al \),
then the processed channels may not be interpreted as the decomposition of a signal.
Redundancy

This is because the LF-HF channels

\[ a \quad \overset{\text{high pass filter}}{\longrightarrow} \quad ah \quad \overset{\text{low pass filter}}{\longrightarrow} \quad al \]
Redundancy

This is because the LF-HF channels are a redundant representation of the signal $a$. In fact, the two channels stream twice as much data as $a$.

*Redundancy may be useful! (more on this later)*
Redundancy

This is because the LF-HF channels are a redundant representation of the signal \( a \). In fact, the two channels stream twice as much data as \( a \).
To avoid redundancy, we apply sub-sampling to the decomposition and then apply some over-sampling (by zero insertion) before the reconstruction to obtain a signal with the original data rate.
Non redundant processing

To avoid redundancy, we apply sub-sampling to the decomposition

\[ a \xrightarrow{\text{high pass filter}} ah \xrightarrow{\downarrow 2} \text{LF filter} \xrightarrow{\uparrow 2} \text{high pass reconstruction filter} \]

\[ a \xrightarrow{\text{low pass filter}} al \xrightarrow{\downarrow 2} \text{HF filter} \xrightarrow{\uparrow 2} \text{low pass reconstruction filter} \]

and then apply some over-sampling (by zero insertion) before the reconstruction to obtain a signal with the original data rate.

The sub-sampling is done by removing samples with odd indices; the zero insertion is applied having the indices multiplied by two and having samples with odd indices set to 0.
Perfect reconstruction filter banks

Let’s give names to the decomposition and reconstruction filters

If the previous scheme yields $a=b$, then the quadruple $(h,g,\tilde{h},\tilde{g})$ is called a **perfect reconstruction** filter bank.

Note: the bar on the decomposition filters means that they are the mirrors of $h$ and $g$, i.e. $\tilde{h}[i] = h[-i]$
Perfect reconstruction filter banks

Let’s give names to the decomposition and reconstruction filters

\[
\begin{align*}
\bar{g} & \rightarrow \downarrow 2 \rightarrow \uparrow 2 \rightarrow \tilde{g} \\
\bar{h} & \rightarrow \downarrow 2 \rightarrow \uparrow 2 \rightarrow \tilde{h}
\end{align*}
\]

If the previous scheme yields \( a=b \), then the quadruple \((h,g,\bar{h},\tilde{g})\) is called a perfect reconstruction filter bank.

Note: the bar on the decomposition filters means that they are the mirrors of \( h \) and \( g \), i.e. \( \bar{h}[i] = h[-i] \)

If \( h=\tilde{h} \) and \( g=\tilde{g} \), then the pair \((h,g)\) is called a pair of conjugate mirror filters.
At the decomposition we have

\[ al[n] = \sum_{k \in \mathbb{Z}} h[k-2n]a[k] \quad \text{and} \quad ag[n] = \sum_{k \in \mathbb{Z}} g[k-2n]a[k] \]

and at the reconstruction we have

\[ b[n] = \sum_{k \in \mathbb{Z}} \tilde{h}[n-2k]al[k] + \sum_{k \in \mathbb{Z}} \tilde{g}[n-2k]ah[k]. \]

We will find again these formulas when crossing scales in multiresolution approximations and wavelet transforms.
Cascade of filter banks

The previous scheme can be applied recursively without losing the perfect reconstruction property.
Cascade of filter banks

The previous scheme can be applied recursively without losing the perfect reconstruction property.
Cascade of filter banks

The previous scheme can be applied recursively without losing the perfect reconstruction property.
Cascade of filter banks

The previous scheme can be applied recursively without losing the perfect reconstruction property.
Redundant v.s. Non Redundant

We have two ways to compute a signal decomposition: a redundant and a non redundant one.

ababab

ababab
Redundant v.s. Non Redundant

We have *two* ways to compute a signal decomposition: a redundant and a non redundant one. This is because there are two ways to sub-sample a signal

```
ababab
```

```
ababab
```
We have *two* ways to compute a signal decomposition: a redundant and a non redundant one. This is because there are two ways to sub-sample a signal:

a) keeping the *even* indices

| a | a | a | a |

| ababab |
Redundant v.s. Non Redundant

We have two ways to compute a signal decomposition: a redundant and a non redundant one. This is because there are two ways to sub-sample a signal:

a) keeping the *even* indices

```
  a a a
```

b) keeping the *odd* indices

```
  b b b
```
Redundant v.s. Non Redundant

We have two ways to compute a signal decomposition: a redundant and a non redundant one. This is because there are two ways to sub-sample a signal:

a) keeping the even indices  
   a a a

b) keeping the odd indices  
   b b b

The redundant decomposition keeps both variants:  
abaaba
Let $h$ a discrete filter and $a_j$ the $j^{th}$ iteration (or scale) of a subsampled transform

$$a_j[n] = \sum_{k \in \mathbb{Z}} h[k - 2n]a_{j-1}[k]$$
Define the “à trous” filter $h_j$ with

$$h_j[n] = \begin{cases} h[p] & \text{if } n = p2^j \\ 0 & \text{else} \end{cases}$$
Extracting the Non Redundant Decomposition

Define the “à trous” filter \( h_j \) with

\[
h_j[n] = \begin{cases} 
  h[p] & \text{if } n = p2^j \\
  0 & \text{else}
\end{cases}
\]

and let \( \tilde{a}_j \) the \( j^{th} \) iteration of the “à trous” transform:

\[
\tilde{a}_j[n] = \sum_{k \in \mathbb{Z}} h_j[k - n] \tilde{a}_{j-1}[k]
\]

with \( \tilde{a}_0 = a_0 \).
Extracting the Non Redundant Decomposition

Then $a_j[n] = \tilde{a}_j[2^j n]$ which means that the non redundant transform is obtained from the redundant transform by extracting one sample out of $2^j$ samples.
Extracting the Non Redundant Decomposition

Then $a_j[n] = \tilde{a}_j[2^j n]$

which means that the non redundant transform is obtained from the redundant transform by extracting one sample out of $2^j$ samples.

The other samples correspond to various choices of odd v.s. even sub-sampling at each scale.
The *discrete time* formulation is nice for algorithm, but if we want to study the fine properties of the (let’s call it *wavelet*) transform, it is better to switch to *continuous time analysis*.
The *discrete time* formulation is nice for algorithm, but if we want to study the fine properties of the (let’s call it *wavelet*) transform, it is better to switch to *continuous time* analysis.

To do so, we shall re-scale the sampling rate to let it tend to 0 as the number of scales $j$ grows.
Fourier Background
When piling up linear filters, the most convenient tool to analyze the behavior of the various transforms is the *Fourier Transform*. 
Linear Time Invariant (LTI) Filters

They are linear operators $L$ which commute with translations.
Linear Time Invariant (LTI) Filters

They are linear operators $L$ which commute with translations.

They are characterized by their impulse response $h$, which is the output for a Dirac input.
**Linear Time Invariant (LTI) Filters**

They are linear operators $L$ which commute with translations.

They are characterized by their *impulse response* $h$, which is the output for a Dirac input.

The output is obtained by *convolution* between the input and the impulse response:

$$L f(t) = \int_{-\infty}^{+\infty} f(u) h(t - u) \, du = \int_{-\infty}^{+\infty} h(u) f(t - u) \, du = h \star f(t).$$
**Fourier Transform**

The Fourier transform of $f$ is

$$
\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t) \exp^{-i\omega t} \, dt
$$
Fourier Transform

The Fourier transform of $f$ is

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t) \exp^{-i\omega t} \, dt$$

It is an isometry from $L^2$ into $L^2$. Its inverse is

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) \exp^{i\omega t} \, d\omega$$
Fourier Transform

The Fourier transform of $f$ is

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t) \exp^{-i\omega t} \, dt$$

It is an isometry from $L^2$ into $L^2$. Its inverse is

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) \exp^{i\omega t} \, d\omega$$

If an LTI filter $L$ has an impulse response $h$, then the Fourier transform of $h$ is called the transfer of $L$. 
LTI Filters in the Fourier Domain

To us, the most important feature of the Fourier transform is that it transforms a convolution into an ordinary product:

\[ \hat{f}_1 \ast \hat{f}_2(\omega) = \hat{f}_1(\omega) \hat{f}_2(\omega) \]

Hence, if \( L \) is an LTI filter with impulse response \( h \), we have

\[ \hat{L}f = \hat{h}\hat{f} \]
LTi Filters in the Fourier Domain

To us, the most important feature of the Fourier transform is that it transforms a convolution into an ordinary product:

$$f_1 \ast f_2(\omega) = \hat{f}_1(\omega) \hat{f}_2(\omega)$$

Hence, if \( L \) is an LTI filter with impulse response \( h \), we have

$$\hat{Lf} = \hat{h}\hat{f}$$

i.e., the Fourier transform of the output is the product of the Fourier transform of the input with the transfer of \( L \).
Fourier Transforms of Sampled Data and Filters

As we wish to switch from discrete time to continuous time, it is sensible to interpret discrete data and filters as samples of continuous objects.

Let us start with filters.
Fourier Transforms of Sampled Data and Filters

As we wish to switch from discrete time to continuous time, it is sensible to interpret discrete data and filters as *samples* of continuous objects.

Let us start with filters.

A sampled filter $L_d$ of a continuous filter $L$ with sample interval $T$ is computed as follows:

$$L_d(t) = \sum_{k=-\infty}^{+\infty} h(kT) f(t - kT)$$
Fourier Transforms of sampled Data and Filters

Its impulse response is

\[ h_d(t) = \sum_{k=-\infty}^{+\infty} h(kT)\delta(t - kT) \]

which yields a transfer function

\[ \hat{h}_d(\omega) = \sum_{k=-\infty}^{+\infty} h(kT)e^{-ik\omega T} \]
Fourier Transforms of sampled Data and Filters

Its impulse response is

\[ h_d(t) = \sum_{k=-\infty}^{+\infty} h(kT) \delta(t - kT) \]

which yields a transfer function

\[ \hat{h}_d(\omega) = \sum_{k=-\infty}^{+\infty} h(kT)e^{-ik\omega T} \]

Since signals and filters play a symmetric role in convolutions, similar formulas hold for signals.
Fourier Analysis of the Decomposition/Reconstruction

Let us come back to our scheme
Fourier Analysis of the Decomposition/Reconstruction

Let us come back to our scheme

\( h, g, \tilde{h}, \tilde{g} \) are standard LTI filters, so their Fourier transforms are well known.
Fourier Analysis of the Decomposition/Reconstruction

Let us come back to our scheme

\[ a \xrightarrow{\bar{g}} \downarrow 2 \xrightarrow{\bar{h}} \downarrow 2 \xrightarrow{\uparrow 2} \tilde{g} \xrightarrow{\tilde{h}} \tilde{g} \xrightarrow{+} b \]

\(h, g, \tilde{h}, \tilde{g}\) are standard LTI filters, so their Fourier transforms are well known.

By contrast, the sub-sampling and over-sampling operators are not LTI, mostly because they incorporate a change in the sampling rate.
Fourier Analysis of Sub and Over-Sampling

Define $y$ by $y[n] = x[2n]$. Then, in the Fourier domain

$$\hat{y}(2\omega) = \frac{1}{2} (\hat{x}(\omega) + \hat{x}(\omega + \pi))$$

Conversely, define

$$y[n] = \begin{cases} 
  x[p] & \text{if } n = 2p \\
  0 & \text{if } n = 2p + 1 
\end{cases}$$

Then $\hat{y}(\omega) = \hat{x}(2\omega)$. 
Fourier Analysis of Sub and Over-Sampling

Define $y$ by $y[n] = x[2n]$. Then, in the Fourier domain

$$\hat{y}(2\omega) = \frac{1}{2} (\hat{x}(\omega) + \hat{x}(\omega + \pi))$$

Conversely, define

$$y[n] = \begin{cases} 
  x[p] & \text{if } n = 2p \\
  0 & \text{if } n = 2p + 1
\end{cases}$$

Then $\hat{y}(\omega) = \hat{x}(2\omega)$.

Note: these transforms are not time invariant!
$g$ can be eliminated from the condition on perfect reconstruction.
Conjugate mirror filters satisfy
\[ |\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2. \]
Fourier Condition for Conjugate Mirror Filters

g can be eliminated from the condition on perfect reconstruction. Conjugate mirror filters satisfy

\[ |\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2. \]

Moreover, if there exists \( l \) such that

\[ \hat{g}(\omega) = \exp^{-i(2l+1)\omega} \hat{h}^*(\omega + \pi) \]

then \((h,g)\) are conjugate mirror filters. This condition is necessary if the filters have a finite length.
Fourier Condition for Conjugate Mirror Filters

$g$ can be eliminated from the condition on perfect reconstruction.
Conjugate mirror filters satisfy

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

Moreover, if there exists $l$ such that

$$\hat{g}(\omega) = \exp^{-i(2l+1)\omega} \hat{h}^*(\omega + \pi)$$

then $(h,g)$ are conjugate mirror filters.
This condition is necessary if the filters have a finite length.

In general, we take $l = 0$, i.e.

$$g[n] = (-1)^n h[1 - n]$$
From Discrete Time to Continuous Time
Cascade of Low Pass Filters

Since the sub-sampled transform is not time invariant, we use the redundant “algorithme à trous” instead. We have

\[ \hat{a}_j(\omega) = \left( \prod_{p=0}^{j-1} \hat{h}^* (2^p \omega) \right) \hat{a}(\omega) \]
Cascade of Low Pass Filters

Since the sub-sampled transform is not time invariant, we use the redundant "algorithme à trous" instead. We have

\[ \hat{a}_j(\omega) = \left( \prod_{p=0}^{j-1} \hat{h}^*(2^p \omega) \right) \hat{a}(\omega) \]

Let us do a time re-scaling, so that the sampling rate of the coarsest filter is 1.
Cascade of Low Pass Filters

Since the sub-sampled transform is not time invariant, we use the redundant “algorithme à trous” instead. We have

\[ \hat{a}_j(\omega) = \left( \prod_{p=0}^{j-1} \hat{h}^*(2^p \omega) \right) \hat{a}(\omega) \]

Let us do a time re-scaling, so that the sampling rate of the coarsest filter is 1.

We have \( \hat{\tilde{a}}_j(\omega) = \left( \prod_{p=1}^{j} \hat{h}^*(2^{-p} \omega) \right) \hat{\tilde{a}}_0(\omega) \)

after time re-scaling.
Transfer at the Infinity

Let the number $j$ of scales go to the infinity. The transfer becomes

$$H^*(\omega) = \prod_{p=1}^{+\infty} \hat{h}^*(2^{-p}\omega).$$
**Transfer at the Infinity**

Let the number $j$ of scales go to the infinity. The transfer becomes

$$H^*(\omega) = \prod_{p=1}^{+\infty} \hat{h}^*(2^{-p}\omega).$$

Assume that $\phi$, the inverse Fourier transform of $H$, exists. Then the corresponding transform is

$$a_\infty(u) = a * \bar{\phi} = \int a(t)\phi(t - u)dt = \langle a, \phi_u \rangle$$
Transfer at the Infinity

Let the number $j$ of scales go to the infinity. The transfer becomes

$$H^*(\omega) = \prod_{p=1}^{+\infty} \hat{h}^*(2^{-p}\omega).$$

Assume that $\phi$, the inverse Fourier transform of $H$, exists. Then the corresponding transform is

$$a_\infty(u) = a * \tilde{\phi} = \int a(t)\phi(t-u)dt = \langle a, \phi_u \rangle$$

where $\phi_u$ is the translate of $\phi$ by an amount of $u$. 
Scaling Equation

Because of the particular form of the transfer $H$, its impulse response necessarily satisfy the following scaling equation:

$$\phi \left( \frac{t}{2} \right) = 2 \sum_{k \in \mathbb{Z}} h[k] \phi(t - k)$$
Because of the particular form of the transfer $H$, its impulse response necessarily satisfy the following scaling equation:

$$
\phi \left( \frac{t}{2} \right) = 2 \sum_{k \in \mathbb{Z}} h[k] \phi(t - k)
$$

This equation is an essential component of multiresolution approximations.
Multiresolution Approximations
**Definition**

A sequence \( \{V_j\}_{j \in \mathbb{Z}} \) of closed subspaces of \( L^2(\mathbb{R}) \) is a multiresolution approximation if the following six properties hold

\[
\forall (j, k) \in \mathbb{Z}^2, \quad f(t) \in V_j \iff f(t - 2^j k) \in V_j, \quad (1)
\]

\[
\forall j \in \mathbb{Z}, \quad V_{j+1} \subset V_j, \quad (2)
\]

\[
\forall j \in \mathbb{Z}, \quad f(t) \in V_j \iff f\left(\frac{t}{2}\right) \in V_{j+1}, \quad (3)
\]

\[
\lim_{j \to +\infty} V_j = \bigcap_{j=-\infty}^{+\infty} V_j = \{0\}, \quad (4)
\]

\[
\lim_{j \to -\infty} V_j = \text{Closure} \left( \bigcup_{j=-\infty}^{+\infty} V_j \right) = L^2(\mathbb{R}). \quad (5)
\]

There exists \( \theta \) such that \( \{\theta(t - n)\}_{n \in \mathbb{Z}} \) is a Riesz basis of \( V_0 \). (6)
Definition (continued)

• \( \theta \) is called a pre-scale function

• if the Riesz basis \( \{\theta(t - n)\}_{n \in \mathbb{Z}} \) is orthogonal, then it is a Hilbert basis and \( \theta \) is called a scaling function

• we shall denote scaling functions by \( \phi \)

• we shall assume now that we have an orthogonal basis
Example: the Haar Basis

The first example of multiresolution approximation has been given by Haar.

In this example, the scaling function is the characteristic function of \([0,1]\); \(V_0\) is the space of functions which are constant on the integer intervals.
Example: the Haar Basis

The first example of multiresolution approximation has been given by Haar.

In this example, the scaling function is the characteristic function of \([0,1]\); \(V_0\) is the space of functions which are constant on the integer intervals.

The approximation of \(f\) in the space \(V_j\) is obtained by averaging \(f\) on the intervals \([k2^j,(k+1)2^j]\).
Multiresolution in the Haar Basis

Original signal
Multiresolution in the Haar Basis

Approximation at scale 1
Multiresolution in the Haar Basis

Approximation at scale 2
Multiresolution in the Haar Basis

Approximation at scale 3
Multiresolution in the Haar Basis

Approximation at scale 4
Multiresolution in the Haar Basis

Approximation at scale 5
Multiresolution in the Haar Basis

Approximation at scale 6
Scale Limitation

At coarse scales, the dilated scale functions are too large to render even the slowest of the variations of the signal.
Scale Limitation

Here is the multiresolution approximation at scale 5 of the same signal with the Daubechies 2 scaling function.
Scale Limitation

It is clearly determined by the shape of the scaling function, and not of the signal

Daubechies2 scaling function
Smoother Scaling Functions

Perfect reconstruction filters (related to biorthogonal scaling functions) provide a smoother low resolution approximations. Here is the previous signal at 5 scales with these functions.
Another example: Splines

A theorem states that the convolution of a pre-scale function with $1_{[0,1]}$ is another pre-scale function.
Another example: Splines

A theorem states that the convolution of a pre-scale function with \(1_{[0,1]}\) is another pre-scale function.

Since iterated convolutions of \(1_{[0,1]}\) with itself generates the so-called \(B(ox)\)-splines, and that the latter generate by translations the space of spline functions on an integer grid, we see that spline functions constitute multiresolution approximations.

The B-splines are not orthogonal, however.
Scaling Equation

Properties (2) and (3) imply that

\[ \frac{1}{\sqrt{2}} \phi \left( \frac{t}{2} \right) = \sum_{n=-\infty}^{+\infty} h[n] \phi(t - n), \]
Properties (2) and (3) imply that

\[
\frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{+\infty} h[n] \phi(t - n),
\]

with

\[
h[n] = \frac{1}{\sqrt{2}} \langle \phi\left(\frac{t}{2}\right), \phi(t - n) \rangle.
\]
In the Fourier domain, the scaling equation translates as
\[ \sqrt{2}\hat{\phi}(2\omega) = \hat{h}(\omega)\hat{\phi}(\omega) \]
In the Fourier domain, the scaling equation translates as

\[ \sqrt{2} \hat{\phi}(2\omega) = \hat{h}(\omega) \hat{\phi}(\omega) \]

which implies (provided convergence occurs)

\[ \hat{\phi}(\omega) = \prod_{p=1}^{+\infty} \frac{\hat{h}(2^{-p}\omega)}{\sqrt{2}} \hat{\phi}(0) \]

By convention, we have \( \hat{\phi}(0) = 1 \)
Fourier Transform of the Scaling Function

In the Fourier domain, the scaling equation translates as

$$\sqrt{2} \hat{\phi}(2\omega) = \hat{h}(\omega)\hat{\phi}(\omega)$$

which implies (provided convergence occurs)

$$\hat{\phi}(\omega) = \prod_{p=1}^{+\infty} \frac{\hat{h}(2^{-p}\omega)}{\sqrt{2}} \hat{\phi}(0)$$

By convention, we have $$\hat{\phi}(0) = 1$$ so the knowledge of $$h$$ is equivalent to the knowledge of $$\phi$$.
Characterization of Multiresolution Approximations

Necessary condition (Mallat, Meyer): Let $\phi \in L^2(\mathbb{R})$ an integrable scaling function. Then the Fourier series of $h[n] = \langle \frac{1}{\sqrt{2}} \phi(\frac{t}{2}), \phi(t - n) \rangle$ satisfies the conjugate mirror filters condition:

$$\forall \omega \in \mathbb{R}, \quad |\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2,$$

and the average condition

$$\hat{h}(0) = \sqrt{2}.$$
Characterization of Multiresolution Approximations

Sufficient condition (Mallat, Meyer): Conversely, if $\hat{h}(\omega)$ is $2\pi$ periodic and continuously differentiable in a neighborhood of $\omega = 0$, if it satisfies the previous conjugate mirror and average conditions, and if

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

then

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

is the Fourier transform of a scaling function $\phi \in L^2(\mathbb{R})$. 
From these conditions, it follows that the design of a scaling function is basically equivalent to finding conjugate mirror filters. As we shall see, others properties of the scaling function can be translated to conditions on the low pass filter.
Numerical Convergence of the Transfer

The product of scaled filter transfers converges fast to the transfer of a low pass filter

filter = Daubechies 4
Numerical Convergence of the Transfer

The product of scaled filter transfers converges fast to the transfer of a low pass filter

filter = Daubechies 4
Numerical Convergence of the Transfer

The product of scaled filter transfers converges fast to the transfer of a low pass filter.

filter = Daubechies 4
Numerical Convergence of the Transfer

The product of scaled filter transfers converges fast to the transfer of a low pass filter

filter = Daubechies 4
Numerical Convergence of the Transfer

The product of scaled filter transfers converges fast to the transfer of a low pass filter.

filter = Daubechies 4
Numerical Convergence of the Transfer

The product of scaled filter transfers converges fast to the transfer of a low pass filter.

The periodized instances of the transfer are pushed towards the infinity as the number of scales grows.

(filter = Daubechies 4)
The projection of a signal $f$ on a resolution $V_j$ is given by

$$\Pi_j f = \sum_{k \in \mathbb{Z}} a_j[k] \phi_{j,k}$$

with

$$\phi_{j,k}(t) = 2^{-j/2} \phi(2^{-j} t - k)$$

and

$$a_j[k] = \langle f, \phi_{j,k} \rangle$$
Crossing Scales

The projection of a signal $f$ on a resolution $V_j$ is given by

$$\Pi_j f = \sum_{k \in \mathbb{Z}} a_j[k] \phi_j,k$$

with

$$\phi_j,k(t) = 2^{-j/2} \phi(2^{-j} t - k)$$

and $a_j[k] = \langle f, \phi_j,k \rangle$

The scaling equation directly implies

$$a_{j+1}[p] = \sum_{n=-\infty}^{+\infty} h[n - 2p] a_j[n]$$
Crossing Scales

The projection of a signal \( f \) on a resolution \( V_j \) is given by
\[
\Pi_j f = \sum_{k \in \mathbb{Z}} a_j[k] \phi_{j,k}
\]
with
\[
\phi_{j,k}(t) = 2^{-j/2} \phi(2^{-j} t - k)
\]
and \( a_j[k] = \langle f, \phi_{j,k} \rangle \)

The scaling equation directly implies
\[
a_{j+1}[p] = \sum_{n=-\infty}^{+\infty} h[n - 2p] a_j[n]
\]
which is just the low pass decomposition formula in our sub band coding
Property (3) (scaling), (1) (shift invariance) and (6) (basis of $V_0$) imply that 
\[
\{ \phi_{j,k} \}_{k \in \mathbb{Z}} \text{ with } \phi_{j,k}(t) = 2^{-j/2} \phi(2^{-j}t - k)
\]
is an orthogonal basis of $V_j$. 
The $V_j$ are approximation spaces because, as $j$ decreases (i.e. the scale gets smaller), smooth functions are efficiently approximated.
**Theorem (Strang,Fix)** Let \( \Pi_j \) the orthogonal projection from \( L^2(\mathbb{R}) \) into \( V_j \). We assume that the scaling function \( \phi \) is compactly supported. The following three propositions are equivalent:

\[
\forall f \in H^N(\mathbb{R}) 2^{-jN} \| \Pi_j f - f \|_{L^2(\mathbb{R})} \to 0 \text{ when } j \to -\infty
\] (1)

\[
\left( \forall f \in H^{N+1}(\mathbb{R}) \right) \left( \forall j \leq 0 \right) \| \Pi_j f - f \|_{L^2(\mathbb{R})} \leq C 2^{j(N+1)} \| f^{(N+1)} \|_{L^2(\mathbb{R})}
\] (2)

\[
(\forall p \in \mathbb{N}) \ p \leq N \Rightarrow \sum_{n \in \mathbb{Z}} \phi(t - n) \int_{\mathbb{R}} \phi(s - n) s^p ds = t^p
\] (3)
Approximations?

The approximation condition is actually related to the low pass filter \( h \).

**Theorem (Fix, Strang)** Suppose that the scaling function \( \phi \) is compactly supported. Then the previous condition is equivalent to:
\[
\hat{h} \text{ and its } N \text{ first derivatives vanish at } \omega = \pi.
\]

This can be extended to scaling functions with suitable decay.
Approximations?

If a function presents some singularities, the latter are not captured by the resolution approximations. Instead, they belong to the orthogonal supplement of the approximation space. This is the wavelet domain.

This property depends on the Strang and Fix condition and also on the smoothness of the scaling function.
Orthogonal Wavelets
From property (2), we know that the resolution spaces are nested.

The information that is lost when switching from the resolution $V_{j-1}$ to the coarser resolution $V_j$ is contained in the detail space $W_j$: 
From property (2), we know that the resolution spaces are nested. The information that is lost when switching from the resolution $V_{j-1}$ to the coarser resolution $V_j$ is contained in the detail space $W_j$: 

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

which is an orthogonal supplement.
Wavelets and Detail Spaces

Not surprisingly, bases of the detail spaces can be obtained by shifting and dilating a single function: there exists a function $\psi$ such that

$$
\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}} \psi \left( \frac{t - 2^j k}{2^j} \right)
$$
Wavelets and Detail Spaces

Not surprisingly, bases of the detail spaces can be obtained by shifting and dilating a single function: there exists a function \( \psi \) such that

\[
\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}} \psi \left( \frac{t - 2^j k}{2^j} \right)
\]

is a basis of \( W_j \).

\( \psi \) is called a wavelet.
Computing a Wavelet

Theorem (Mallat, Meyer): Let $\phi$ a scaling function and $h$ its low pass conjugate filter. Define $\psi$ by its Fourier transform

$$\hat{\psi}(\omega) = \frac{1}{\sqrt{2}} \hat{g} \left( \frac{\omega}{2} \right) \hat{\phi} \left( \frac{\omega}{2} \right),$$

with

$$\hat{g}(\omega) = \exp^{-i\omega} \hat{h}^*(\omega + \pi).$$

Define

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

For every scale $j$, $\{\psi_{j,n}\}_{n \in \mathbb{Z}}$ is an orthogonal basis of $W_j$. Across scales, $\{\psi_{j,n}\}_{(j,n) \in \mathbb{Z}^2}$ is an orthogonal basis of $L^2(\mathbb{R})$. 
Computing a Wavelet

In the time domain, we have

\[ g[n] = (-1)^n h[1 - n] \]

From previous theorems, we know that \((h, g)\) is a pair of conjugate mirror filters.
Vanishing Moments

The $n^{th}$ moment of a function $f$ is the integral

$$\int_{\mathbb{R}} f(t)t^{n-1}dt$$

The function $f$ is said to have $p$ vanishing moments if the previous integral is 0 for $n = 1\ldots p$. Wavelets have at least one vanishing moment, i.e. they have a zero average.
Vanishing Moments

Actually, the vanishing moments of the wavelet are related to the approximation capacity of the scaling function.
Vanishing Moments

**Theorem** Let $\psi$ a wavelet which spans a basis of $L^2(\mathbb{R})$. If $\hat{\psi}(\omega)$ is $p$ continuously differentiable at $\omega = 0$, the following three properties are equivalent:

i) The wavelet $\psi$ has $p$ vanishing moments.

ii) $\hat{\psi}(\omega)$ and its $p - 1$ first derivatives are zero at $\omega = 0$.

iii) $\hat{h}(\omega)$ and its $p - 1$ first derivatives are zero at $\omega = \pi$.

The last condition is equivalent to the Strang and Fix approximation condition on the scaling function $\phi$. 
Having a basis of $L^2(\mathbb{R})$ is nice, but there are many useful functions which do not belong to $L^2(\mathbb{R})$. For instance, polynomials do not belong to $L^2(\mathbb{R})$. Indeed, if the wavelet has $p$ vanishing moments, we can easily see that polynomials of degree $< p$ have zero coordinates in the wavelet basis.
Beyond $L^2$

More generally, the solutions of linear, autonomous differential equations do not belong to $L^2(\mathbb{R})$. Indeed, they are a combination of polynomials, exponentials and sinusoids. For instance, the sinusoid does not belong to $L^2(\mathbb{R})$ since its Fourier transform is a Dirac.
Beyond $L^2$

Fortunately, grouping the projections on coarse detail spaces together, we can obtain a decomposition of signals into a resolution approximation and finer details which holds for a much larger class of functions.
Beyond $L^2$

Let $\phi$ a scaling function, $\psi$ its wavelet, and $f \in L^2(\mathbb{R})$. Then

$$f = \sum_{n \in \mathbb{Z}} \langle f, \phi_{J,n} \rangle \phi_{J,n} + \sum_{j \leq J} \sum_{n \in \mathbb{Z}} \langle f, \psi_{j,n} \rangle \psi_{j,n}$$

This decomposition holds for a broad class of functions. For instance, it is valid for polynomials. It is also valid for Lipschitz functions (assuming, for instance, that $\psi$ is compactly supported).
Fast Decomposition and Reconstruction (Mallat)

Remember that

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

Since we have orthogonal bases for all of these spaces, we can rewrite this equation in terms of coordinates.
Fast Decomposition and Reconstruction (Mallat)

Define \( a_j[n] = \langle f, \phi_{j,n} \rangle \) and \( d_j[n] = \langle f, \psi_{j,n} \rangle \).
From the scaling equation and the definition of \( \psi \) from \( \phi \), we have

\[
a_j[p] = \sum_{n=-\infty}^{+\infty} h[n - 2p] a_{j-1}[n]
\]

\[
d_j[p] = \sum_{n=-\infty}^{+\infty} g[n - 2p] a_{j-1}[n].
\]

This is exactly the decomposition in the sub-band coding scheme!
Fast Decomposition and Reconstruction (Mallat)

Since we know that \((h, g)\) is a pair of conjugate mirror filters, we have the following reconstruction formula:

\[
a_{j-1}[p] = \sum_{n=-\infty}^{+\infty} h[p-2n] a_j[n] + \sum_{n=-\infty}^{+\infty} g[p-2n] d_j[n]
\]
The Time

Warp
Sample? What Sample?

Until now, we have called *samples* the values of a discrete time signal.
Sample? What Sample?

Until now, we have called samples the values of a discrete time signal.
This implicitly states that the discrete time values are the instantaneous capture of the value of continuous time signal.
Sample? What Sample?

Until now, we have called *samples* the values of a discrete time signal. This implicitly states that the discrete time values are the instantaneous capture of the value of continuous time signal.

Actually, the continuous to discrete conversion often involve fine physical processes which are neglected in the model.
Cheating (on) Time

From now on, we shall assimilate the discrete time material we are working with to the scaling coefficients of a continuous time signal $f$, i.e.
Cheating (on) Time

From now on, we shall assimilate the discrete time material we are working with to the scaling coefficients of a continuous time signal $f$, i.e.

$$a_j[n] = \langle f, \phi_j, n \rangle$$
From now on, we shall assimilate the discrete time material we are working with to the scaling coefficients of a continuous time signal \( f \), i.e.

\[
a_j[n] = \langle f, \phi_j, n \rangle
\]

And remember: the transfer of a not so long cascade of discrete filter banks approximates the scalar product with a continuous time scaling function or wavelet.
From now on, we shall assimilate the discrete time material we are working with to the scaling coefficients of a continuous time signal $f$, i.e.

$$a_j[n] = \langle f, \phi_j, n \rangle$$

And remember: the transfer of a not so long cascade of discrete filter banks approximates the scalar product with a continuous time scaling function or wavelet.

Actually, the plot of the Daubechies scaling function was computed using these very same cascades of discrete conjugate mirror filter banks!
Cheating (on) Time

From now on, we shall assimilate the discrete time material we are working with to the scaling coefficients of a continuous time signal $f$, i.e.

$$a_j[n] = \langle f, \phi_j, n \rangle$$

And remember: the transfer of a not so long cascade of discrete filter banks approximates the scalar product with a continuous time scaling function or wavelet.

Actually, the plot of the Daubechies scaling function was computed using these very same cascades of discrete conjugate mirror filter banks!

The rationale behind this scheme is that the cascade algorithm approximates the scalar product of the scaling function with a Dirac.
Scale Bounds

In practice we have two bounds on the time scales we use:
Scale Bounds

In practice we have two bounds on the time scales we use:

- the lower bound is given by the basic time rate of our signal
In practice we have two bounds on the time scales we use:

- the lower bound is given by the basic time rate of our signal

- the upper bound is given by the scale of the coarsest features in the signal (see previous examples)
Time and Filter
Finite Support

• We can reasonably assume that the original signal has a finite support
• The transforms have also finite support if the filters have a Finite Impulse Response (FIR)
FIR Filters

Theorem The scaling function \( \phi \) is compactly supported if and only if \( h \) has a finite impulse response, and they have the same support.

If the support of \( h \) (and of \( \phi \)) is \([N_1, N_2]\), then the support of \( \psi \) is \( \left[ \frac{N_1 - N_2 + 1}{2}, \frac{N_2 - N_1 + 1}{2} \right] \).

So, once again, questions of interest in the continuous domain can be solved by careful design of the discrete filter \( h \).
Getting Wavelets
Filters v.s. Functions

In most cases, properties of the scaling functions and wavelets are equivalent to properties of the low pass filter $h$:

The characterization of multiresolution approximations is done through conditions on $h$. 
Filters v.s. Functions

In most cases, properties of the scaling functions and wavelets are equivalent to properties of the low pass filter $h$:

The Discrete Wavelet Transform (DWT) and its inverse are computed by filter banks which are immediately defined by $h$
Filters v.s. Functions

In most cases, properties of the scaling functions and wavelets are equivalent to properties of the low pass filter $h$:

Approximation properties of the scaling function and vanishing moments of the wavelet are equivalent to a condition on $h$.
Filters v.s. Functions

In most cases, properties of the scaling functions and wavelets are equivalent to properties of the low pass filter $h$:

Compact support of the scaling function and wavelet is equivalent to $h$ being a FIR filter
Filters v.s. Functions

In most cases, properties of the scaling functions and wavelets are equivalent to properties of the low pass filter $h$:

Symmetry (or lack of) is equivalent to a condition on $h$
Consequence

In numerical packages, do not look for (dyadic) wavelets or scaling functions.
Consequence

In numerical packages, do not look for (dyadic) wavelets or scaling functions.

What you will get is the filter $h$, under the form of its impulse response if it is finite.

The filter $g$ follows immediately.
In numerical packages, do not look for (dyadic) wavelets or scaling functions. What you will get is the filter $h$, under the form of its impulse response if it is finite. The filter $g$ follows immediately.

Moreover, if you really need the values of the wavelet or scaling function, the scaling function can be computed by using a cascade algorithm solely based on $h$, and the wavelet is given by

$$
\frac{1}{\sqrt{2}} \psi \left( \frac{t}{2} \right) = \sum_{n=-\infty}^{+\infty} g[n] \phi(t - n)
$$
Consequence

In numerical packages, do not look for (dyadic) wavelets or scaling functions.

What you will get is the filter \( h \), under the form of its impulse response if it is finite.

The filter \( g \) follows immediately.

But most of the time you will only need the filters!
Cascade Algorithm

Here is how the Daubechies 2 scaling function is computed by iterations of low resolution approximations using a cascade of filter banks on a discrete impulse.
Cascade Algorithm

Impulse ("discrete Dirac")
Cascade Algorithm

Low resolution approximation at scale 1
Cascade Algorithm

Low resolution approximation at scale 2
Cascade Algorithm

Low resolution approximation at scale 3
Cascade Algorithm

Low resolution approximation at scale 4
Cascade Algorithm

Low resolution approximation at scale 5
Cascade Algorithm

Low resolution approximation at scale 6
Cascade Algorithm

Low resolution approximation at scale 7
Observe the spikes. It has been proven that the Daubechies 2 scaling function is not differentiable at dyadic points.
Wavelet classes

Typically, when looking for a filter $h$ in a numerical package, you will have to provide two kinds of information:

- the name of the wavelet family
- the number of vanishing moments of the wavelet

Here are a selected inventory of (dyadic) wavelet families
Wavelet classes

Meyer Wavelets

Meyer wavelets are compactly supported in the Fourier domain and are hence

• infinitely differentiable
• do not have a bounded support.

They are implemented in the Fourier domain.
Wavelet classes

Battle-Lemarié Wavelets

B-Splines are a non orthogonal basis of spline resolution spaces. This basis is orthogonalized into the Battle-Lemarié wavelets.

The scaling functions are not compactly supported, but they have an exponential decay. They are symmetric.
Wavelet classes

Battle-Lemarie cubic spline wavelet
Wavelet classes

Daubechies Wavelets

Daubechies wavelets are orthogonal compactly supported wavelets.

For a given number $p$ of vanishing moments, $h$ has minimum length support $[0,2p-1]$
Wavelet classes

Symlets

The only symmetric, compactly supported, scaling function is given by the Haar basis.

Symlets have been designed by I. Daubechies to be orthogonal, compactly supported, and "as symmetric as possible".
Wavelet classes

Coiflets

Coiflets have been designed by I. Daubechies for R.R. Coifman. In addition to the wavelets, the scaling functions also have vanishing moments:

$$\int_{\mathbb{R}} t^k \phi(t) \, dt = 0 \quad \text{for } 1 \leq k < p$$

The scaling function provides good approximations from function samples.
By assimilating the values discrete signal to the scaling coefficients of a continuous functions, we reap the benefits of having two points of view:
Summary

By assimilating the values of a discrete signal to the scaling coefficients of a continuous function, we reap the benefits of having two points of view:

- Moving across scales, computing approximations and details, involves only a finite number of operations.
Summary

By assimilating the values discrete signal to the scaling coefficients of a continuous functions, we reap the benefits of having two points of view:

• moving across scales, computing approximations and details, involves only a finite number of operations

• these computations can be interpreted in the light of continuous analysis
Warning

The wavelet processing scheme presented is not *shift invariant*. This may produce undesirable effects, as the transform of a shifted signal may be different from the transform of the unshifted one.

A (discrete) shift invariant transform is available by using the “*algorithme à trous*”.
Shift Invariant Example

Here are the low resolution Haar approximations of a signal using a shift invariant transform. (This is the same signal as the one used before)
Shift Invariant Example

Original signal
Shift Invariant Example

Scale 1
Shift Invariant Example

Scale 2
Shift Invariant Example

Scale 3
Shift Invariant Example

Scale 4
Shift Invariant Example

Scale 5
Shift Invariant Example

The blocky appearance has disappeared
A common feature of the previous transforms is that they have a discrete set of scales (in geometric progression).

Intermediate scales in the wavelet transform can be reached when processing discrete signals by interpolation on the "natural" scales (which are called octaves). Such an interpolation is not innocuous.
Actually, wavelet transforms with a finer granularity in time and scale are related to another class of wavelet transforms: the *continuous* wavelet transforms.
Actually, wavelet transforms with a finer granularity in time and scale are related to another class of wavelet transforms: the *continuous* wavelet transforms.

(and this is another story)
The spread of a function in the time/frequency domain is characterized by its Heisenberg boxes.
Time/Frequency Resolution

The spread of a function in the time/frequency domain is characterized by its Heisenberg boxes.

An important feature of wavelets is that, as the scale $j$ gets finer, the time spread of the wavelet

$$\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}} \psi \left( \frac{t - 2^j k}{2^j} \right)$$

shrinks down to 0.
Time/Frequency Tiling

Here is roughly how the time/frequency spread of a wavelet basis covers the "time/frequency domain".
Wavelet Coefficients
At Fine Scales

Assume that the scale of a wavelet is “fine” and, to simplify, that 0 belongs to the support of the wavelet. If $f$ is a smooth function then

$$f(t) = \sum_{k=0}^{p-1} \frac{t^k}{k!} f^{(k)}(0) + \frac{1}{(p-1)!} \int_0^t (t - s)^{p-1} f^{(p)}(s) \, ds$$

If the wavelet $\psi$ has $p$ vanishing moments then

$$|\langle f, \psi_{j,0} \rangle| \leq K |\text{supp}(\psi)|^{p+1} \|\psi\|_{\infty} \|f^{(p)}\|_{\infty} 2^j (p+\frac{1}{2})$$

since

$$\int_{\mathbb{R}} t^k \psi(t) \, dt = 0 \text{ for } 0 \leq k \leq p - 1$$
Wavelet Coefficients At Fine Scales

Actually, the behavior of the wavelet coefficients of $\psi_{j,0}$ as $j$ tends to $-\infty$ depends only on the regularity of $f$ in the neighborhood of $0$. This is why the wavelet transform allows analysis and manipulation of the pointwise regularity of signals.
Lipschitz Regularity

A function \( f \) is \( \alpha \)-Lipschitz, \( \alpha > 0 \), at point \( v \), if there exist \( K > 0 \) and a polynomial \( p_v \) with degree \( m = \lfloor \alpha \rfloor \) such that

\[
\forall t \in \mathbb{R}, \quad |f(t) - p_v(t)| \leq K |t - v|^{\alpha}.
\]
Lipschitz Regularity and Wavelet Transform

We assume that the wavelet has $n$ vanishing moments, and that its scaling function is $n$ times continuously differentiable.
Lipschitz Regularity and Wavelet Transform

Necessary condition (Jaffard): If $f \in L^2(\mathbb{R})$ is $\alpha$ Lipschitz at $v$ with $\alpha \leq n$, then there exists $A$ such that, for all $j \leq 0$,

$$\sup_{k \in \mathbb{Z}} | \langle f, \psi_{j,k} \rangle | \leq A 2^j(\alpha + \frac{1}{2}) \left( 1 + \left| \frac{2^j k - v}{2^j} \right|^{\alpha} \right).$$

If $v$ belongs to the support of $\psi_{j,k}$, then

$$| \langle f, \psi_{j,k} \rangle | \leq A 2^j(\alpha + \frac{1}{2})(1 + L^\alpha),$$

where $L$ is the length of the support of $\psi$. 
Lipschitz Regularity and Wavelet Transform

Sufficient condition (Jaffard): Let $S_{v,j,\epsilon}$ the set of indices $k$ such that the support of $\psi_{j,k}$ intersects $(v - \epsilon, v + \epsilon)$. If $\alpha < n$ is not an integer, and if there exists $A$ such that, for all $j \leq 0$,

$$\sup_{k \in S_{v,j,\epsilon}} |\langle f, \psi_{j,k} \rangle| \leq A 2^j(\alpha + \frac{1}{2})$$

and if $f$ is locally integrable, then there exists a polynomial $p_v$ with degree $E(\alpha)$ and $K$ such that

$$\forall t \in (v - \epsilon, v + \epsilon) \ |f(t) - p_v(t)| \leq K |t - v|^\alpha$$

In particular, if $f$ is bounded, then $f$ is $\alpha$-Lipschitz at $v$. 
Lipschitz Regularity and Wavelet Transform

Interpretation

If we restrict the Lipschitz condition to a neighborhood of \( \nu \) with an non integer \( \alpha \), with an compactly supported wavelet, then the Lipschitz condition is equivalent to

\[
\log_2 |\langle f, \psi_{j,k} \rangle| \approx j \left( \alpha + \frac{1}{2} \right) \text{ as } j \to -\infty
\]

when \( \nu \) belongs to the support of \( \psi \).
Lipschitz Regularity and Wavelet Transform

Application

If a point $v$ is singular, i.e., it has a lower Lipschitz regularity than its neighbors, then the wavelet coefficients of wavelets whose support converge to $v$ converge slower than their neighbors.
This means that the wavelet coefficients of singular points dominate the others at the fine scales.
Other Wavelet Bases
Biorthogonal Wavelets

Biorthogonal wavelets are constructed from a pair of multiresolution approximations such that the pre-scale functions satisfy

\[ \int_{\mathbb{R}} \phi(t - j) \tilde{\phi}(t - k) dt = \delta_{j,k} \text{ for all } j, k \in \mathbb{Z} \]
Biorthogonal wavelets are constructed from a pair of multiresolution approximations such that the pre-scale functions satisfy

$$\int_{\mathbb{R}} \phi(t - j) \tilde{\phi}(t - k) dt = \delta_{j,k} \text{ for all } j, k \in \mathbb{Z}$$

The corresponding filters are a quadruple of perfect reconstruction filters.
Biorthogonal wavelets are constructed from a pair of multiresolution approximations such that the pre-scale functions satisfy

$$\int_{\mathbb{R}} \phi(t - j) \tilde{\phi}(t - k) \, dt = \delta_{j,k} \text{ for all } j, k \in \mathbb{Z}$$

The corresponding filters are a quadruple of perfect reconstruction filters.

The Strang and Fix condition on $\phi$ is equivalent to a condition on $\tilde{h}$. 

**Biorthogonal Wavelets**
Handling Finite Data

Until now, we have handled signals with support $= \mathbb{R}$.

In practice, only a finite amount of data is available.
How do we handle it?
Zero Padding

The simplest thing to do is to extend the signal by zero outside its domain. If you use FIR filters, you will have only a finite number of computations to do.
The simplest thing to do is to extend the signal by zero outside its domain. If you use FIR filters, you will have only a finite number of computations to do. (since the signal is finite and discrete, only a finite number of scales are relevant).
Zero Padding

The simplest thing to do is to extend the signal by zero outside its domain. If you use FIR filters, you will have only a finite number of computations to do. (since the signal is finite and discrete, only a finite number of scales are relevant).

Unless your signal is smoothly vanishes at the bounds, you will have singularities there, and hence large wavelet coefficients. This is a feature present in several other methods.
Example

Here is the original signal

![Graph of the original signal](image-url)
Example

Here is how it looks in Simulink
We use a redundant, shift invariant transform to better localize the singularities
Here are the wavelet coefficients at scale 1
Example

Wavelet coefficients at scale 2
Example

Wavelet coefficients at scale 3
Example

Wavelet coefficients at scale 4
Example

Wavelet coefficients at scale 5
Example

Wavelet coefficients at scale 6
Example

And finally, the scale coefficients at scale 6
Periodization

Another trick is to periodize the signal to have its support equal to $\mathbb{R}$.
Periodization

Another trick is to periodize the signal to have its support equal to $\mathbb{R}$. If the values of the signal are not equal at the left and right bound, then the periodized signal is discontinuous there.
Periodization

Another trick is to periodize the signal to have its support equal to $\mathbb{R}$.

If the values of the signal are not equal at the left and right bound, then the periodized signal is discontinuous there.

The wavelet transform of periodized signals is computed by replacing convolutions by circular convolutions:

$$f \ast h[n] = \sum_{p=0}^{N-1} f[p] h[n - p] = \sum_{p=0}^{N-1} f[n - p] h[p]$$
Periodization

Another trick is to periodize the signal to have its support equal to $\mathbb{R}$.

If the values of the signal are not equal at the left and right bound, then the periodized signal is discontinuous there.

The formula holds if both signals have the same support. If not, some zero padding may be necessary.
Folded Wavelets

To avoid creating discontinuities by periodization, the signal is folded before periodization:
Folded Wavelets

To avoid creating discontinuities by periodization, the signal is folded before periodization:

It turns out that folding the signal is equivalent to folding the wavelet. The periodic signal is continuous, but generally has a discontinuous derivative.
Wavelets on the Interval

All the previous methods create large wavelet coefficients at the boundaries. A way to avoid this is to create a basis on $L^2[0, 1]$.

Restricting the scaling functions to [0,1] creates a basis, but it is not orthogonal and does not produce wavelets.
Wavelets on the Interval

Cohen, Daubechies and Vial have constructed orthogonal wavelets on $[0,1]$ with vanishing moments.
The transforms involve specific filters at the neighborhood of the boundaries.
Wavelets on the Interval

The catch
Because we use dyadic sub-sampling and over-sampling, at some point the number of samples must a multiple of $2^j$, where $j$ is the number of scales of the transform.

Using redundant transforms (based, for instance, on interpolation) removes this dyadic requirement. The redundancy may be restricted to the edges.
Wavelets on Images

The simplest way to build 2-dimensional wavelets is to start from 1-D wavelets and proceed as follows:
The scaling functions and its translates are defined by
\[ \phi_{m,n}(x, y) = \phi(x - m) \psi(y - n) \]
and there are three wavelets defined by
\[ \psi^1(x, y) = \phi(x) \psi(y) , \quad \psi^2(x, y) = \psi(x) \phi(y) , \quad \psi^3(x, y) = \psi(x) \psi(y) \]
2-D Example
(from UviWave)

Here is a decomposition on a classic image.

Original Image
Here is a decomposition on a classic image.
2-D Example
(from UviWave)

Here is a decomposition on a classic image.
2-D Example
(from UviWave)

Here is a decomposition on a classic image.
Here is a decomposition on a classic image.
When the signal (such a digital image) has integer values, it is efficient to use fixed point arithmetic in the wavelet transform.

An example of integer wavelet transform is based on polynomial interpolation.
When the signal (such a digital image) has integer values, it is efficient to use fixed point arithmetic in the wavelet transform.

On an integer grid, polynomial interpolation is a finite dimensional linear problem with integers coefficients.
**Integer Wavelets**

When the signal (such a digital image) has integer values, it is efficient to use fixed point arithmetic in the wavelet transform.

Its solution is a linear combination of the signal with a finite number or rational numbers. It is suited to fixed point arithmetic.
**Integer Wavelets**

When the signal (such a digital image) has integer values, it is efficient to use fixed point arithmetic in the wavelet transform.

The properties of these filters can be further improved by using integer (or rational) wavelet lifting.
Applications
Nonlinear Approximation

As we have said, the wavelet coefficients at singularities and at the fine scales dominate the others.
As we have said, the wavelet coefficients at singularities and at the fine scales dominate the others.

If we sort the wavelet coefficients by decreasing magnitude, then the wavelet coefficients corresponding to singularities appear first.
Nonlinear Approximation

As we have said, the wavelet coefficients at singularities and at the fine scales dominate the others.

If we sort the wavelet coefficients by decreasing magnitude, then the wavelet coefficients corresponding to singularities appear first.

After a while, the only wavelet coefficients that appear correspond to regular points.
Consider a signal $f$ decomposed in an orthogonal basis $\{g_m\}$:

$$f = \sum_m \langle f, g_m \rangle g_m$$
Nonlinear Approximation

Consider a signal $f$ decomposed in an orthogonal basis $\{g_m\}$:

$$f = \sum_m \langle f, g_m \rangle g_m$$

Denote by $\epsilon[M]$ the minimum of $\|f - f_M\|^2$

where $f_M$ is obtained by selecting a family of $M$ basis vectors in the decomposition.
Nonlinear Approximation

Consider a signal $f$ decomposed in an orthogonal basis $\{g_m\}$:

$$f = \sum_m \langle f, g_m \rangle g_m$$

Denote by $\epsilon[M]$ the minimum of $\|f - f_M\|^2$

where $f_M$ is obtained by selecting a family of $M$ basis vectors in the decomposition.

$\epsilon[M]$ is minimized by sorting the coefficients magnitudes in decreasing order, and choosing the $M$ first ones.
Nonlinear Approximation in Wavelet Bases

We consider signals on \([0,1]\), and aggregate the scaling functions with the wavelets. Bringing the two previous slides together yields

**Theorem:** If \( f \in L^2[0,1] \) is uniformly \( \alpha \)-Lipschitz on \([0,1]\) – \( S \) where \( S \) is a finite set of points in \([0,1]\) and \( \alpha \) est not an integer with \( \alpha < q \), then

\[
\epsilon_{\text{min}}[M] = O(M^{-2\alpha})
\]

We have assumed that the wavelets are \( q \) times continuously differentiable and have \( q \) vanishing moments.
Nonlinear Approximation in Wavelet Bases

We consider signals on $[0,1]$, and aggregate the scaling functions with the wavelets. Bringing the two previous slides together yields In other words, the decay of the optimal error $\epsilon_M$ is independent of the singularities. By contrast, if the wavelet coefficients are sorted by scale, the decay of the error is determined by the worst order of singularity.
Nonlinear Approximation in Wavelet Bases

This shows that setting the smallest wavelet coefficients to zero is an efficient way to approximate signals with a finite number of singularities.
Example

Here is a signal with a slight numerical noise

Original signal and denoised signal
Example

Denoised signal and approximated one. Only 0.4% of all coefficients are kept.
Example

Approximation error
Images do not feature “a finite number of singularities”. A more suitable concept is \textit{bounded variation}:
Nonlinear Approximation for Images

Images do not feature “a finite number of singularities”. A more suitable concept is bounded variation:

The total variation of an image $f$ on $[0,1] \times [0,1]$ is

$$\| f \|_V = \int_0^1 \int_0^1 |\nabla f(x, y)| \, dx \, dy$$

where the gradient may be a distribution. The image has a bounded variation if the total variation is bounded.
Nonlinear Approximation for Images

Sorting the wavelet coefficients by scale gives the following error decay:

\[ \| f - P_{V_j} f \|_{L^2} \leq B \| f \|_V \| f \|_\infty 2^{j/2} \]
Nonlinear Approximation for Images

Sorting the wavelet coefficients by scale gives the following error decay:

\[
\| f - P_{V_j} f \|_{L^2} \leq B \| f \|_V \| f \|_\infty 2^{j/2}
\]

Sorting the wavelet coefficients by decreasing magnitude gives the following error decay:

\[
\epsilon_{\text{min}}[M] \leq B_2 \| f \|_V^2 M^{-1}
\]
Nonlinear Approximation for Images

Sorting the wavelet coefficients by scale gives the following error decay:

$$\| f - P_{V^J} f \|_{L^2} \leq B \| f \|_V \| f \|_\infty 2^{j/2}$$

Sorting the wavelet coefficients by decreasing magnitude gives the following error decay:

$$\epsilon_{\min}[M] \leq B_2 \| f \|_V^2 M^{-1}$$

If, for comparison purposes, we take $M = 2^{-j}$, $j < 0$, the error becomes

$$\epsilon_{\min}[2^{-j}] \leq B_2 \| f \|_V^2 2^j$$
Example
(From Stéphane Mallat)

Original Lena image of $N^2 = 256^2$ pixels.
Linear approximation from the $M = N^{2/16}$ Symmlet 4 wavelet coefficients at the scales $2^j > 2^4$.
The relative norm of the error is $0.036$. 
Example

(From Stéphane Mallat)

Non-linear approximation of a Lena image of $N^2=256^2$ pixels, with $M = N^2/16$ wavelet coefficients. The relative norm of the error is 0.011.
Example
(From Stéphane Mallat)

Location of the basis coefficient with largest magnitude
The previous nonlinear approximation of the Lena image has, if we simply discard the zero coefficients, a 0.5 bit/pixel compression ratio.
The previous nonlinear approximation of the Lena image has, if we simply discard the zero coefficients, a 0.5 bit/pixel compression ratio.

However, careful quantization and coding of the chosen coefficients may greatly improve the image quality for a similar bit/pixel ratio.
Compression

The actual compression mixes wavelet considerations with more general compression algorithms.
Compression

The actual compression mixes wavelet considerations with more general compression algorithms.

In this presentation we will stick to two basic remarks on wavelet nonlinear approximation.
Compression

First, if the smallest non discarded basis coefficient has modulus, say, $2^j$, we can drop the bits from 1 to $j-1$ without notably affecting the image rendering.
Compression

First, if the smallest non discarded basis coefficient has modulus, say, $2^j$, we can drop the bits from 1 to $j-1$ without notably affecting the image rendering.

Secondly, even after this, there may be long or short binary representations of the remaining basis coefficient. Considering this improves the bit allocation.
Finally, compression algorithm from information theory further improve the bit rate of the compressed image.
Compression examples

(From Stéphane Mallat)

Here are compressed images at 0.5 bit/pixel

Lena

Images sizes are 256x256
**Compression examples**

*(From Stéphane Mallat)*

Here are compressed images at 0.5 bit/pixel

**GoldHill**

Images sizes are 256x256
Compression examples
(From Stéphane Mallat)

Here are compressed images at 0.5 bit/pixel

Bateaux

Images sizes are 256x256
Compression examples
(From Stéphane Mallat)

Here are compressed images at 0.5 bit/pixel

Mandrill

Images sizes are 256x256
Compression examples
(From Stéphane Mallat)

Here are the locations of the remaining pixels

Lena

There are more pixels than in the nonlinear approximation (see Lena), but the compression improves the bit/pixel ratio.
Compression examples

(From Stéphane Mallat)

Here are the locations of the remaining pixels

GoldHill

There are more pixels than in the nonlinear approximation (see Lena), but the compression improves the bit/pixel ratio.
Compression examples
(From Stéphane Mallat)

Here are the locations of the remaining pixels

There are more pixels than in the nonlinear approximation (see Lena), but the compression improves the bit/pixel ratio.
Compression examples
(From Stéphane Mallat)

Here are the locations of the remaining pixels

Mandrill

There are more pixels than in the nonlinear approximation (see Lena), but the compression improves the bit/pixel ratio.
Denoising

We shall not present the rigorous theory beneath denoising with wavelets.
We shall not present the rigorous theory beneath denoising with wavelets.

However, simple heuristics roughly explain what happens and provide some insight on the tuning of the requested parameters.
**Denoising**

Consider a piecewise regular signal. Here is the same signal contaminated by a white noise.
Close to the singularities, the amplitude of the noise is negligible. Outside, the low pass filter will remove the noise.
The strategy is the following:

- a threshold is chosen that separates the (smaller) noise wavelet coefficients from the (larger) wavelet coefficients corresponding to the singularities
- below this threshold, the wavelet coefficients are set to zero (this is called hard thresholding)
Denoising

The result is the following:

outside the neighborhood of the singularities, the wavelet coefficients are not due to these singularities; they are created by the noise. A suitable threshold removes them, and the denoising is left to the low resolution approximation.
Denoising

The result is the following:

close to the singularities, the wavelet coefficients are left unchanged: the noise is not removed, but the shape of the sharp transients (like slope) is preserved
Shift Invariant Denoising Example
(adapted from Mallat)

Synchronized clean and denoised signals
Shift Invariant Denoising Example

(adapted from Mallat)

Estimations error
Shift Invariant Denoising Example
(adapted from Mallat)

Noisy signal and denoised signal (with processing delay)
Denoising With (Almost No Delay)

Wavelet transforms adapted to the half-axis (for data) provides denoising with no (or very small) delay.

Here the data and noise are the same as in the previous slide, but the algorithm provides an estimate with no (or a small) delay. Here there is a delay of 10 samples, while the signal’s duration is of 2880 samples.
Conclusion

Wavelets are efficient in processing signals with sharp transients.