

## Wavelets

- ▶ Compress audio signals and images
- ▶ Reveal structure in turbulence
  - but yet to give economical algorithm for turbulence
- ▶ **Local Finite Differences**
  - good for discontinuities
  - poor for waves, 8+ points per cycle
- ▶ **Global Spectral**
  - good for waves
  - poor for discontinuities,  $\tilde{f} \sim 1/k$  with no wave of period  $2\pi/k$  (NB  $k^{-5/3}$  spectrum of turbulence)

Wavelets: best of both: **local waves**

**Musical tune:** sequence of notes of different amplitude, frequency, duration

Fourier not see finite duration, FD need 8+ points per cycle

Musical score very economical → wavelets

## Continuous Wavelet Transform

**Mother wavelet**  $\psi(x)$ : translate through  $b$ , dilate by  $a$

$$\psi_{a,b} = a^{-1/2} \psi\left(\frac{x-b}{a}\right)$$

Wavelet components

$$f_{a,b} = \int \psi_{a,b}^*(x) f(x) dx$$

Invert

$$f(x) = \frac{1}{C_\psi} \int f_{a,b} \psi_{a,b} \frac{dadb}{a^2} \quad \text{where } C_\psi = \int \frac{|\tilde{\psi}(s)|^2}{|k|} dk$$

For PDEs:  $\frac{\partial}{\partial b} f_{a,b} = \left(\frac{\partial f}{\partial x}\right)_{a,b}$

## Possible wavelets

Morlet

$$(e^{ikx} - e^{-k^2/2})e^{-x^2/2}$$

Mexican hat (Marr)

$$\frac{d^2}{dx^2} (e^{-x^2/2}) = (x^2 - 1)e^{-x^2/2}$$

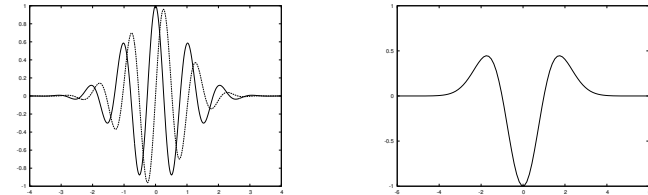


Figure : (a) Morlet wavelet with  $k = 6$ . (b) Marr Mexican hat wavelet.

Decay rapidly in  $x$  and Fourier  $k$

## Discrete Wavelet Transform

For unit interval  $[0, 1]$ , with periodic extension (on a circle)

$N = 2^n$  points,  $x_k = k/N$  for  $k = 0, 1, \dots, N-1$

Restrict to discrete set of translations and dilations

$$\psi_{i,j} = 2^{i/2} \psi(2^i x - j)$$

for  $i = 0, \dots, n-1$  and  $j = 0, \dots, 2^i - 1$

E.g. one wavelet  $\psi_{0,0} = \psi(x)$  on  $[0, 1]$

Two  $\psi_{1,0} = \sqrt{2}\psi(2x)$  on  $[0, \frac{1}{2}]$  and  $\psi_{1,1} = \sqrt{2}\psi(2x-1)$  on  $[\frac{1}{2}, 1]$

Down to finest level with  $2^{n-1}$  wavelets on  $2^{n-1}$  subintervals

Total number of wavelets =  $1 + 2 + \dots + 2^{n-1} = N$  on  $N$  data points

**A Multiscale representation**

## ... Discrete Wavelet Transform

Wavelet components (using periodic extension near boundary)

$$f_{i,j} = \frac{1}{N} \sum_k \psi_{i,j}(x_k) f(x_k)$$

If  $\psi(x)$  is nonzero only on unit interval,

then  $f_{0,0}$  is sum over  $N$  points,

$f_{1,0}$  and  $f_{1,1}$  are each sums over  $N/2$  points, etc

Hence cost of all components is  $O(N \ln_2 N)$

Advantage of special **orthogonal wavelets** (discrete)

$$\frac{1}{N} \sum_k \psi_{i,j}(x_k) \psi_{l,m}(x_k) = \delta_{il} \delta_{jm}$$

Then inverse discrete wavelet transform

$$f(x_k) = \sum_{i,j} f_{i,j} \psi_{i,j}(x_k)$$

Possible orthogonal wavelets: Haar, Sinc, Meyer, Battle-Lemarié,  
Daubechies, symlets, Coiflets

## ... Fast Wavelet Transform

The 7 components cannot represent 8 data points. Missing mean value, so introduce

$$f_{0,0}^\phi = \frac{1}{8}(f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7).$$

Then inversion works, e.g. (student exercise!)

$$f_0 = f_{0,0}^\phi \phi_{0,0}(0) + f_{0,0} \psi_{0,0}(0) + f_{1,0} \psi_{1,0}(0) + f_{2,0} \psi_{2,0}(0),$$

with  $\phi_{0,0}(0) = \psi_{0,0}(0) = 1$ ,  $\psi_{1,0}(0) = \sqrt{2}$  and  $\psi_{2,0}(0) = 2$ .

## Fast Wavelet Transform – $O(N)$

Start with simple case of Haar wavelet

$$\psi(x) = \begin{cases} 1 & \text{if } 0 \leq x < \frac{1}{2}, \\ -1 & \text{if } \frac{1}{2} \leq x < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Non-zero on single interval, so obviously orthogonal

Consider simple case of  $N = 8 = 2^3$  points. Wavelet components

$$f_{0,0} = \frac{1}{8}(f_0 + f_1 + f_2 + f_3 - f_4 - f_5 - f_6 - f_7),$$

$$f_{1,0} = \frac{\sqrt{2}}{8}(f_0 + f_1 - f_2 - f_3), \quad f_{1,1} = \frac{\sqrt{2}}{8}(f_4 + f_5 - f_6 - f_7),$$

$$f_{2,0} = \frac{1}{4}(f_0 - f_1), \quad f_{2,1} = \frac{1}{4}(f_2 - f_3),$$

$$f_{2,2} = \frac{1}{4}(f_4 - f_5), \quad f_{2,3} = \frac{1}{4}(f_6 - f_7).$$

Problem 1: mean value.      Problem 2: duplication

## ... Fast Wavelet Transform

Need **scaling function**  $\phi(x)$ , which for Haar is

$$\phi(x) = \begin{cases} 1 & \text{if } 0 \leq x < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Same dilations and translations of this basic scaling function

$$\phi_{i,j}(x) = 2^{i/2} \phi(2^i x - j)$$

for  $i = 0, \dots, n-1$  and  $j = 0, \dots, 2^i - 1$

Similar components

$$f_{i,j}^\phi = \frac{1}{N} \sum_k \phi_{i,j}(x_k) f(x_k).$$

## ... Fast Wavelet Transform

The Fast transform: Start at finest level

$$f_{2,0} = \frac{1}{4}(f_0 - f_1) \quad f_{2,0}^\phi = \frac{1}{4}(f_0 + f_1)$$

and similarly other  $f_{2,j}$   $f_{2,j}^\phi$

Next level up

$$f_{1,0} = \frac{1}{\sqrt{2}}(f_{2,0}^\phi - f_{2,1}^\phi) \quad f_{1,0}^\phi = \frac{1}{\sqrt{2}}(f_{2,0}^\phi + f_{2,1}^\phi),$$

Similarly next and coarsest level

$$f_{0,0} = \frac{1}{\sqrt{2}}(f_{1,0}^\phi - f_{1,1}^\phi) \quad f_{0,0}^\phi = \frac{1}{\sqrt{2}}(f_{1,0}^\phi + f_{1,1}^\phi).$$

Cost:  $4N$  operations

## ... Fast Wavelet Transform

The Inverse Transform: have  $f_{0,0}^\phi$  and all wavelets  $f_{i,j}$   
Start at coarsest level

$$f_{1,0}^\phi = \frac{1}{\sqrt{2}}(f_{0,0}^\phi + f_{0,0}) \quad f_{1,1}^\phi = \frac{1}{\sqrt{2}}(f_{0,0}^\phi - f_{0,0})$$

Similarily generate all  $f_{i,j}^\phi$  from coarser level  
Finally recover the data

$$f_0 = \frac{1}{2}(f_{2,0}^\phi + f_{2,0}) \quad f_1 = \frac{1}{2}(f_{2,0}^\phi - f_{2,0})$$

and similarly all other  $f_k$

NB: The Fast Transform and its Inverse do not use the values of the function, just the **filter coefficients**  $\pm \frac{1}{\sqrt{2}}$   
Gives generalisation from Haar to other orthogonal wavelets

## ... Fast Wavelet Transform

At any  $l$ th stage, the partial sum

$$\sum_{j=0}^{2^l-1} f_{l,j}^\phi \phi_{l,j}(x)$$

represents all the courser scale variations of the function which have not been described by wavelets at the scale of  $l$  and finer, as in

$$\sum_{j,i \geq l} f_{i,j} \psi_{i,j}(x).$$

Fast Wavelet Transform is a **bank of frequency filters** in signal processing

– a high-pass to the wavelet components and a low-pass to the remaining scaling components

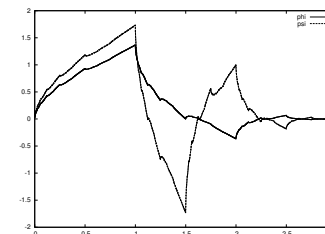
## Daubechies Wavelets

A scaling function must be a linear combination of finer scale scaling functions. The Daubechies D-2 has just four, so that

$$\phi(x) = \sqrt{2}(h_0\phi(2x) + h_1\phi(2x-1) + h_2\phi(2x-2) + h_3\phi(2x-3))$$

Constraints of orthogonality, normalisation and some vanishing moments require

$$h_0 = \frac{1 + \sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3 + \sqrt{3}}{4\sqrt{2}}, \quad h_2 = \frac{3 - \sqrt{3}}{4\sqrt{2}}, \quad h_3 = \frac{1 - \sqrt{3}}{4\sqrt{2}}$$



Distinctly irregular  
Not good for PDEs  
But just use filter coefficients

## ..Daubechies Wavelets

The Fast D-2 Wavelet Transform

$$f_{i,j}^{\phi} = \sum_k h_k f_{i+1,2i+k}^{\phi} \quad f_{i,j} = \sum_k g_k f_{i+1,2i+k}^{\phi},$$

where

$$g_0 = -h_3 \quad g_1 = h_2 \quad g_2 = -h_1 \quad g_3 = h_0$$

There is good Wavelet Toolbox in [MATLAB](#)