Algorithms and Complexity

"... divide each difficulty at hand into as many pieces as possible and as could be required to better solve them."

René Descartes, *Discourse on the Method*

Classic results

Complexity of discrete bases computation

Complexity of wavelet series computation

Complexity of overcomplete expansions

Special topics

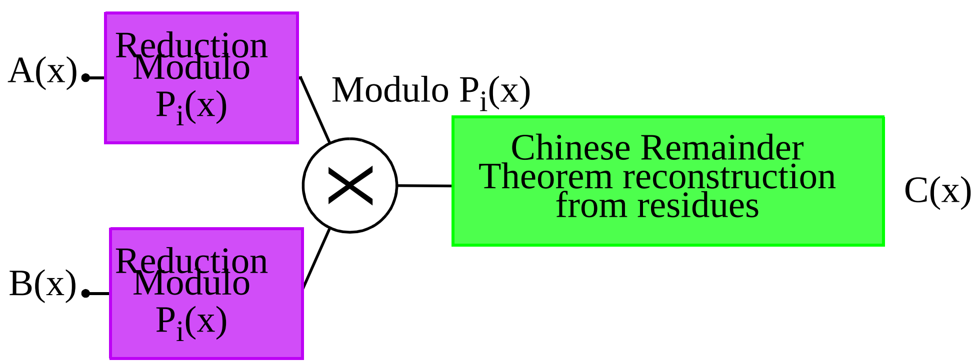
Classic results...
... fast convolution

$$a[n]^* b[n] \Leftrightarrow A(x)B(x)$$

Example: $\{a[0], a[1]\}^* \{b[0], b[1]\}$

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} b_0 & 0 & 0 \\ 0 & b_0 - b_1 & 0 \\ 0 & 0 & b_1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$$

Reduction, pointwise multiplication, interpolation



Classic results...
... Chinese Remainder Theorem

There is a one-to-one map between integers $\mathrm{mod}(N)$ **and their residues mod** p_i **when** $N = \prod p_i$ **and** $(p_i, p_j) = 1,\ i \neq j$

Reconstruction formula from residues $a_i = a \bmod(p_i)$**:**

$$a = \sum_i a_i \cdot \beta_i \cdot \frac{N}{p_i}$$

where β_i is the solution of

$$\beta_i \cdot \frac{N}{p_i} = 1 \qquad \mathrm{mod}(p_i)$$

which has always a solution since $\left(\frac{N}{p_i}, p_i\right) = 1$ (Bezout)

Classic results...
... Convolution Theorem

Theorem: [Convolution Theorem]

- circular convolution: $A(x)B(x)\bmod(x^N - 1)$
- factorization into roots of unity

$$x^N - 1 = \prod_i (x - W_N^i)$$

- reduction modulo $(x^N - 1)$ is equivalent to the evaluation of $A(x)$ and $B(x)$ at roots of unity
- in matrix notation, this leads to the usual Fourier transform factorization of circular convolution

$$c = B \cdot a = F^{-1} \cdot \Lambda \cdot F \cdot a$$

- therefore: efficient convolution requires efficient Fourier transforms

Classic results...
... fast Fourier transform

Fast Fourier transform

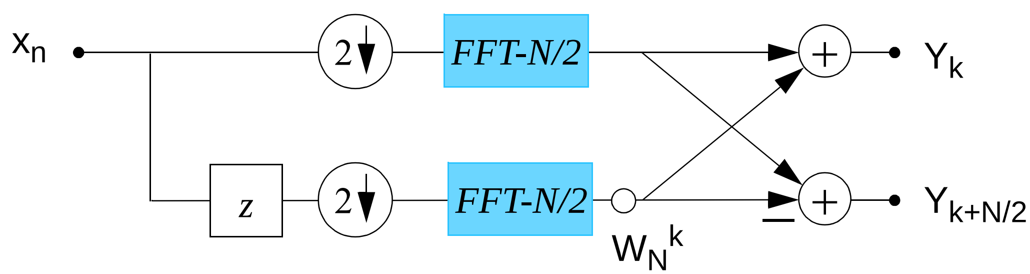
$$Y_k = \sum_{n=0}^{N-1} x_n \cdot W_N^{nk} \qquad k = 0 \ldots N-1$$

where W_N is the root of unity, and N is a power of 2

Idea: consider $\{x_{2n}, x_{2n+1}\}$ **and** $\{Y_k, Y_{k+N/2}\}$ **instead**

$$Y_k = \sum_{n=0}^{N/2-1} x_{2n} \cdot W_{N/2}^{nk} + \left(\sum_{n=0}^{N/2-1} x_{2n+1} \cdot W_{N/2}^{nk} \right) W_N^k$$

$$Y_{k+N/2} = \sum_{n=0}^{N/2-1} x_{2n} \cdot W_{N/2}^{nk} - \left(\sum_{n=0}^{N/2-1} x_{2n+1} \cdot W_{N/2}^{nk} \right) W_N^k$$



Classic results...
... fast Fourier transform

Iterative procedure

- $\log_2 N - 1$ times
- $N/2$ multiplications each time
- order $N \log_2 N$ complexity
- this is the famous Cooley-Tukey FFT, for $N = 2^m$

Other lengths

- prime factor: N is composite, coprime factors
- this leads to a true MD FFT

Rader's algorithm: when N is prime
it becomes a circular convolution N-1

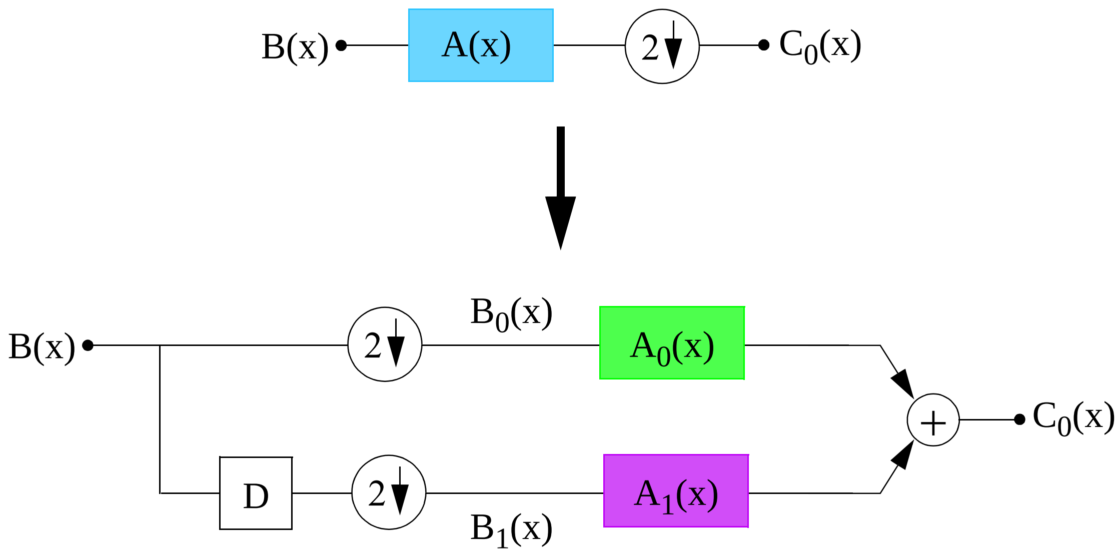
Other famous transform: DCT

- DCT-N $\Leftrightarrow$ permutation + FFT-N + post-rotations
- same complexity as a real-FFT + $3N/2$ multiplications

These algorithms come in handy for fast STFT and FWT...

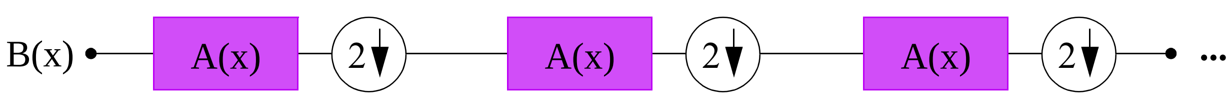
Classic results...
... complexity of multirate signal processing

Compute at lowest possible rate



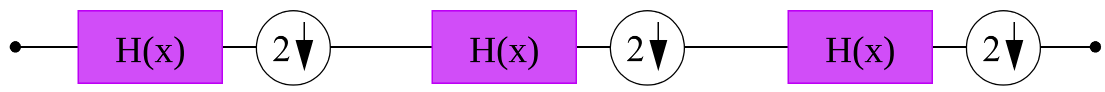
Complexity remains bounded when iterating, since

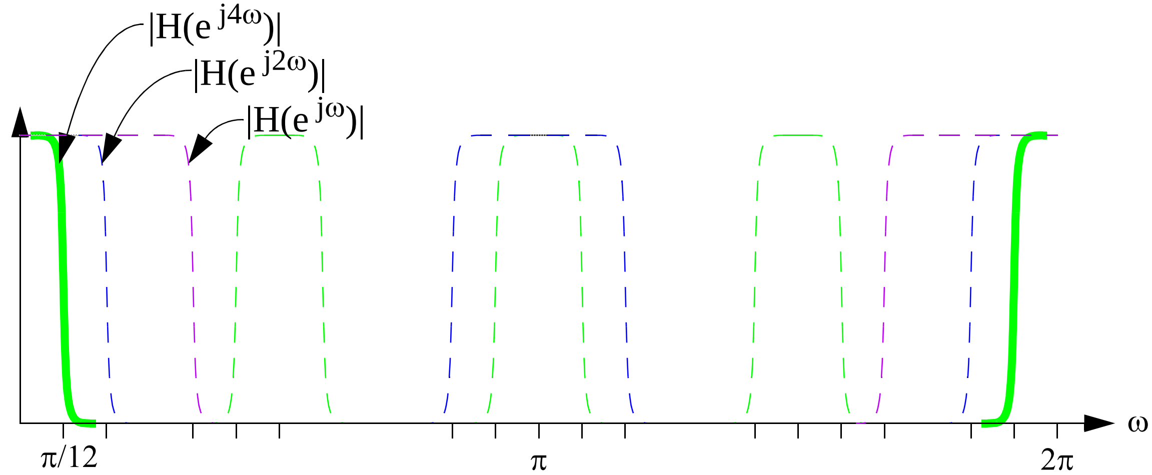
$$L + L/2 + L/4 + L/8 + \ldots < 2L$$



Classic results...
... complexity of multirate signal processing

Multirate implementation of narrow-band filters

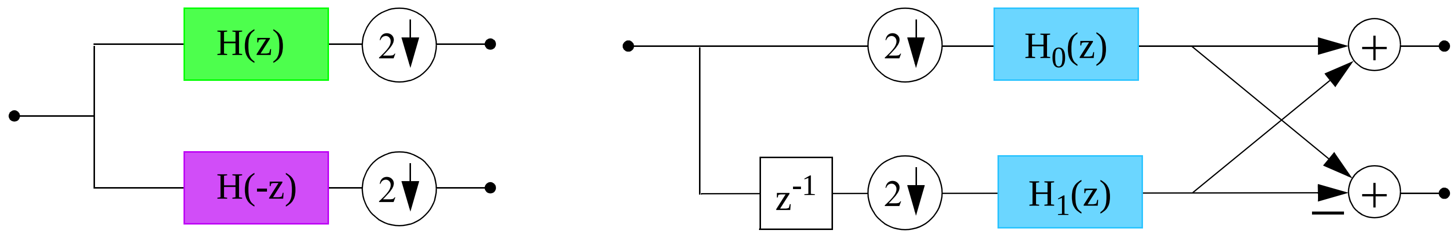




Iteration of 1/3**-band filter leads to** 1/12**-band filter, with transition band** 1/4 **of original, and bounded complex**

Complexity of discrete bases computation

QMF filter



since the polyphase matrix allows the factorization

$$\begin{bmatrix} H_0 & H_1 \\ H_0 & -H_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} H_0 & 0 \\ 0 & H_1 \end{bmatrix}$$

where H_i are the polyphase components of $H(z)$

Complexity of discrete bases computation...
... lattice forms

Linear phase lattice: section with symmetric matrices

$$\begin{bmatrix} 1 & \alpha \\ \alpha & 1 \end{bmatrix} = c \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \beta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- 1 or 2 multiplications per section

Orthogonal lattice: sections with rotation matrices

$$\begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \alpha + \beta & 0 & 0 \\ 0 & \alpha - \beta & 0 \\ 0 & 0 & -\beta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix}$$

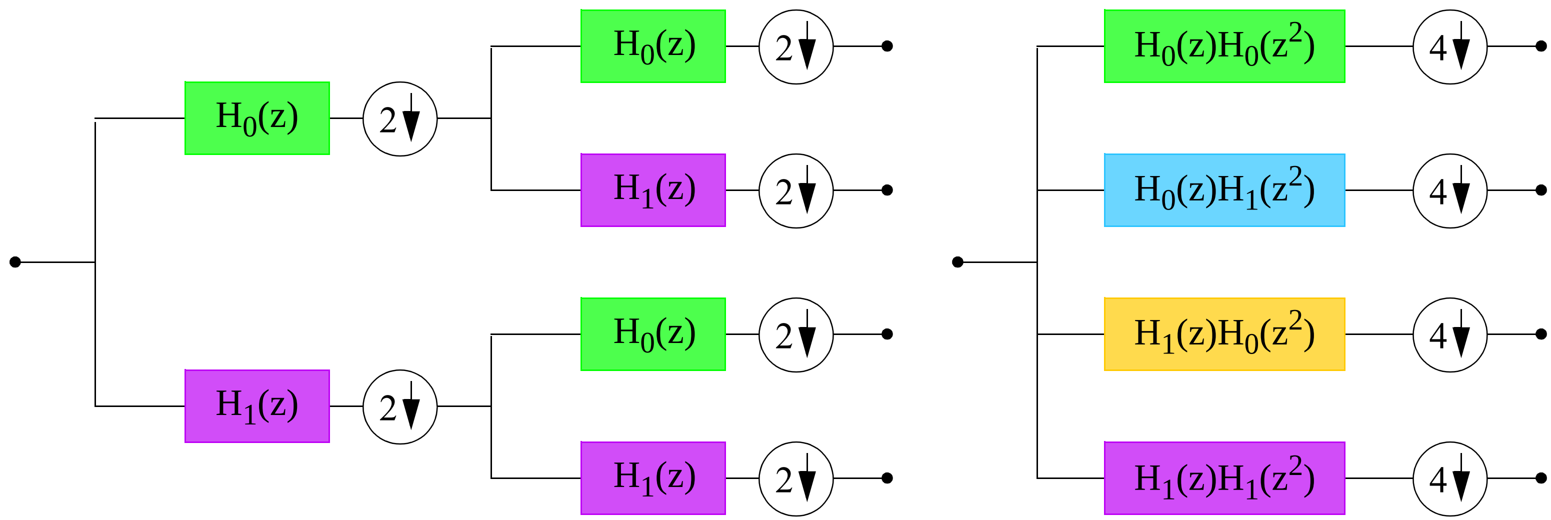
- 2 or 3 multiplications per section

Complexity of discrete bases computation...
... filter bank trees

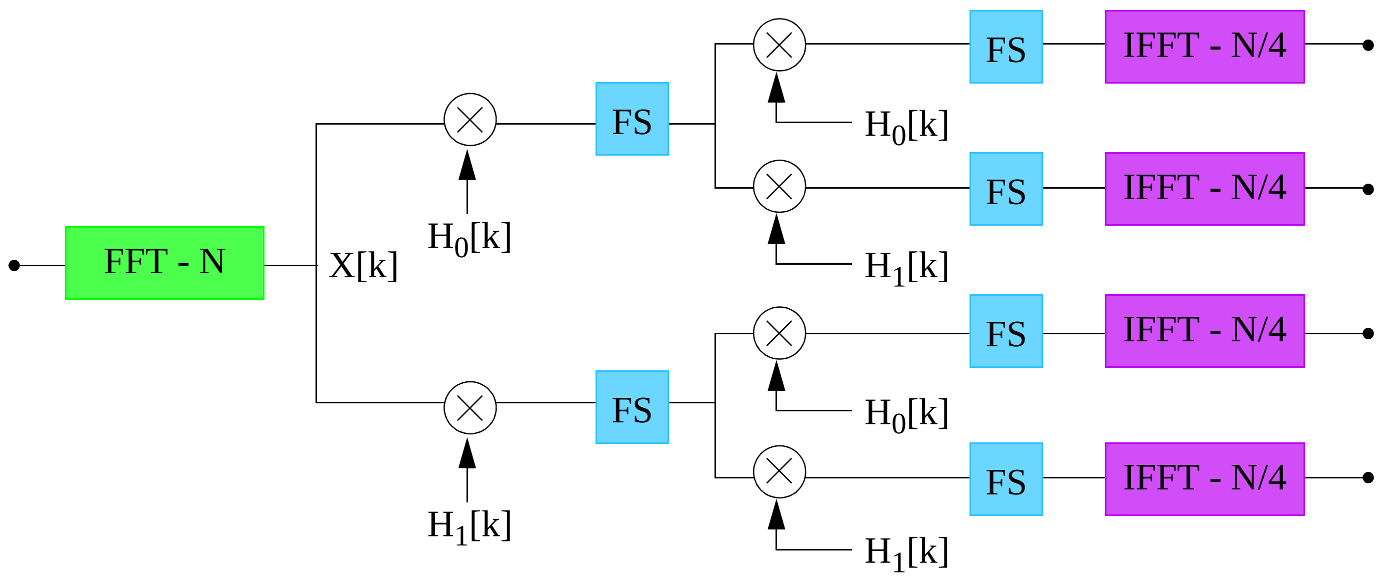
Full tree: depth times complexity of first stage

Octave-band tree: 2 times complexity of first stage

Parallelized version



Complexity of discrete bases computation...
... Fourier-domain computation



Downsampling in Fourier transform domain

Consider

$$Y_k = \sum_{n=0}^{N-1} x_n \cdot W_N^{nk} \qquad k = 0 \ldots N-1$$

as well as

$$Z_k = \sum_{n=0}^{N/2-1} x_{2n} \cdot W_{N/2}^{nk} \qquad k = 0 \ldots \frac{N}{2} - 1$$

then one can show (discrete version of subsampling)

$$Z_k = \frac{1}{2}(Y_k + Y_{k+N/2})$$

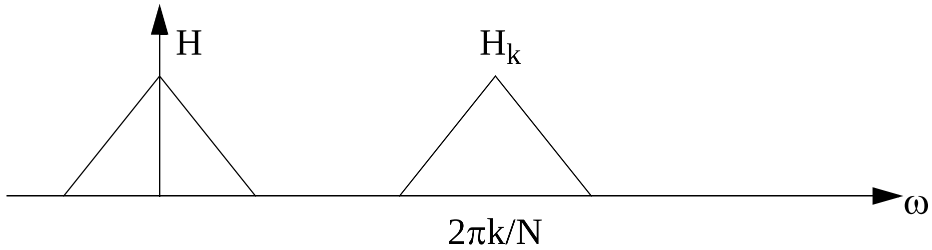
This can be used to do multirate operations in the Fourier domain, as in the previous algorithm

Complexity: Modulated filter banks

All filters derived from one prototype by modulation

$$H_k(z) = H(W_N^k z)$$

$$h_k[n] = \{h[0], W_N^{-k}h[1], W_N^{-2k}h[2], W_N^{-3k}h[3], \ldots\}$$



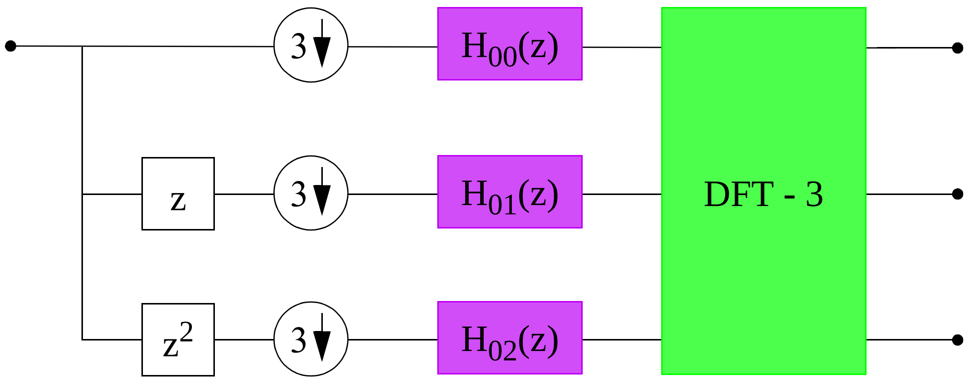
Polyphase matrix has a particular form

Example:

$$\begin{bmatrix} H_0 & H_1 & H_2 \\ H_0 & W^{-1}H_1 & W^{-2}H_2 \\ H_0 & W^{-2}H_1 & W^{-1}H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & W^{-1} & W^{-2} \\ 1 & W^{-2} & W^{-1} \end{bmatrix} \cdot \begin{bmatrix} H_0 & 0 & 0 \\ 0 & H_1 & 0 \\ 0 & 0 & H_2 \end{bmatrix}$$

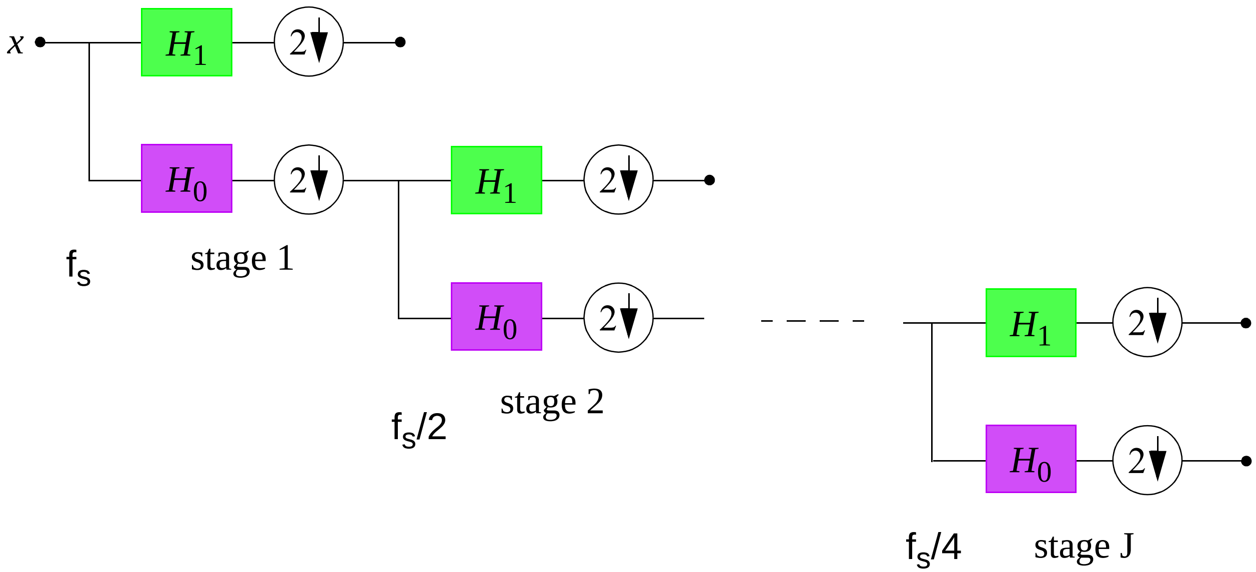
Complexity of discrete bases computation...
... modulated filter banks

Implementation: polyphase filters + FFT



- this leads to efficient STFT
- other modulated banks (such as cosine) have similar algorithms

Complexity of wavelet series computation



Total complexity: $C + C/2 + C/4 + C/8 + \ldots < 2C$

Iterated filters

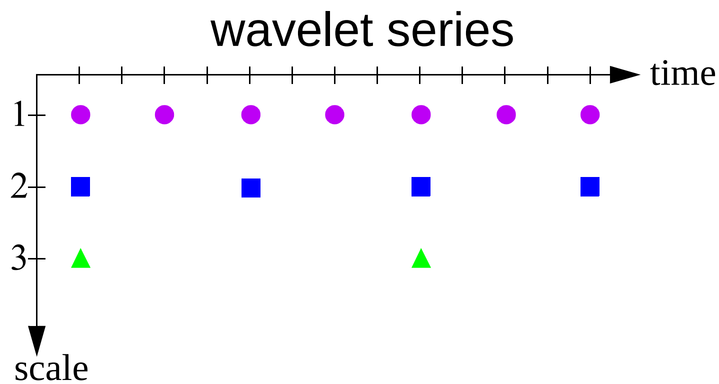
$$G^{(i)}(z) = \prod_{k=0}^{i-1} G\left(z^{2^k}\right) = G(z)G^{(i-1)}(z^2) = G\left(z^{2^{i-1}}\right)G^{(i-1)}(z)$$

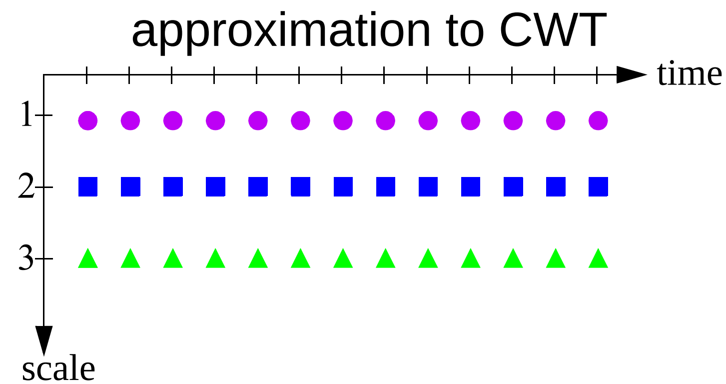
Complexity: $2^i L^2$ **or** $2^i L \log L$ **using FT**

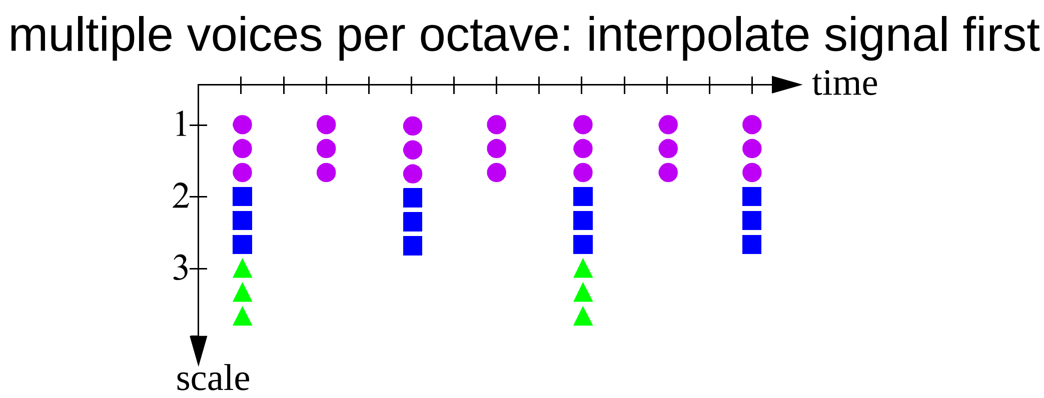
Complexity of overcomplete expansions

STFT: use modulated filter banks

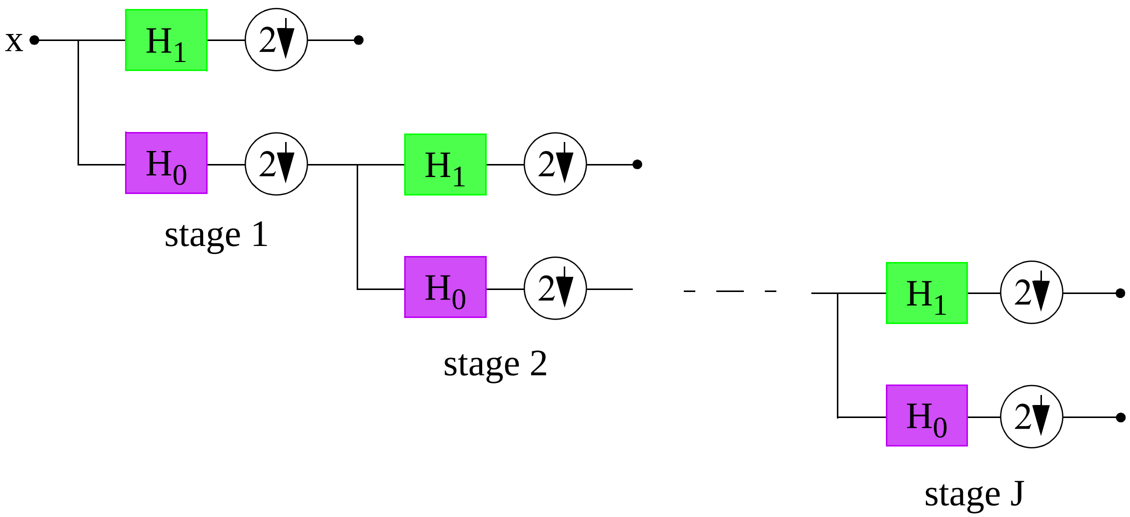
CWT: use ‘‘algorithme à trous’’



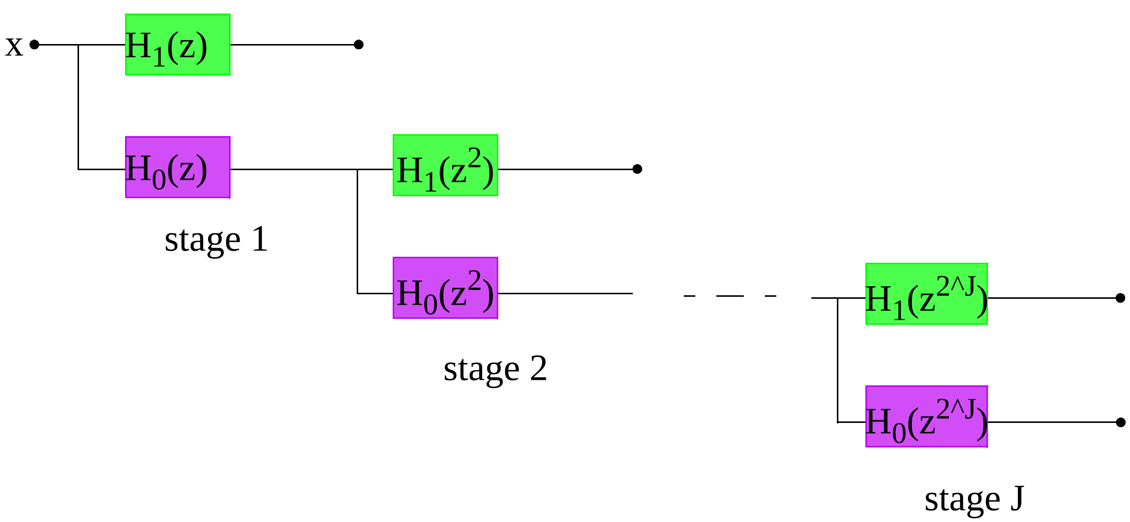




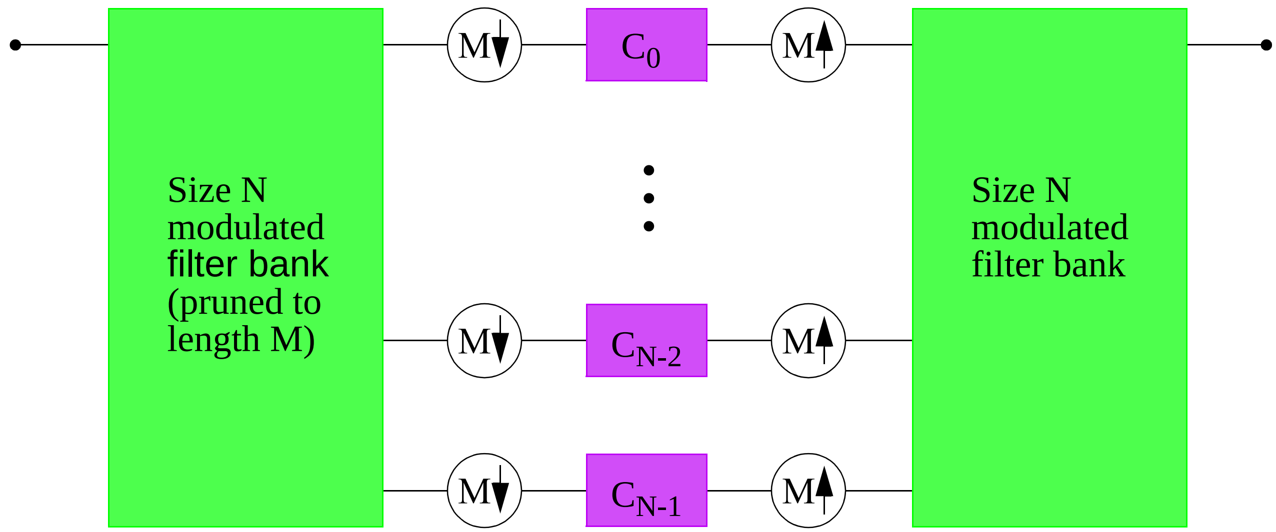
Algorithme à trous



Move downsamplers to the end and skip downsampling



Special topics in complexity...
... overlap save/add algorithm as filter bank



Special topics in complexity...
... generalizations

Channel filters

Filter banks based on fast convolutions