

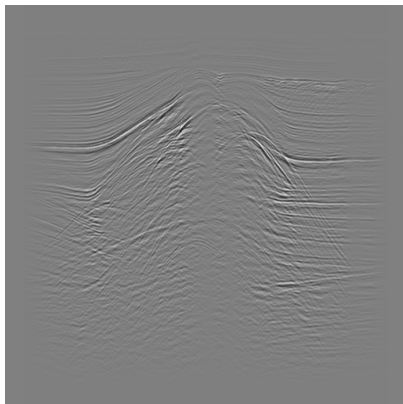
# Sparse nonlinear approximation of functions in two dimensions by sums of exponential functions.

Marcus Carlsson

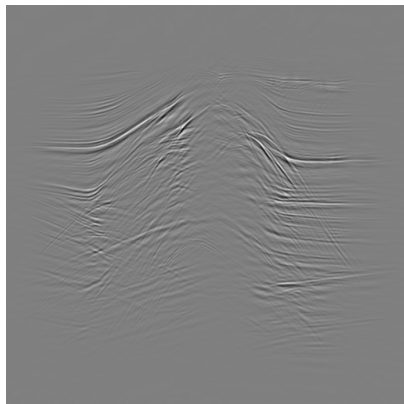
January 11, 2010

Approximation of seismic image:

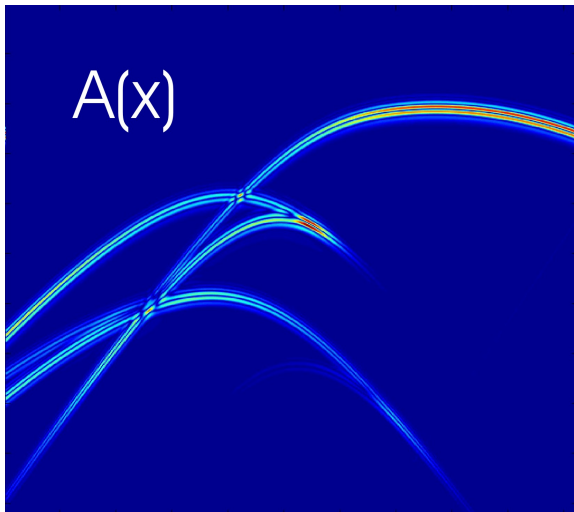
Original  $512 \times 512$ -image:



Reconstruction using 5130-wave packets.

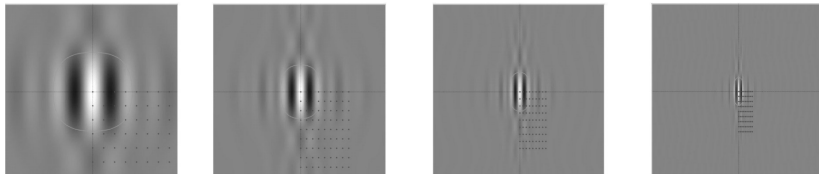


The aim is to sparsely decompose (seismic) images into sums of "wave packets". E.g. this one:

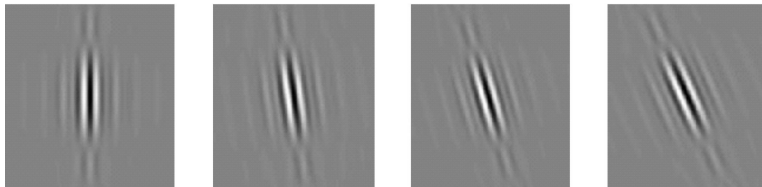


# Wave-packets:

different scales  $k$

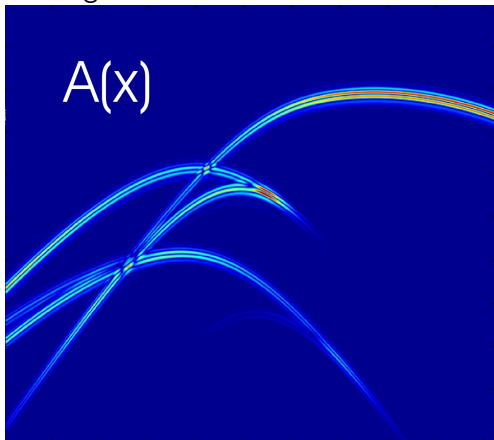


$k=4$ ; different rotations  $\nu$



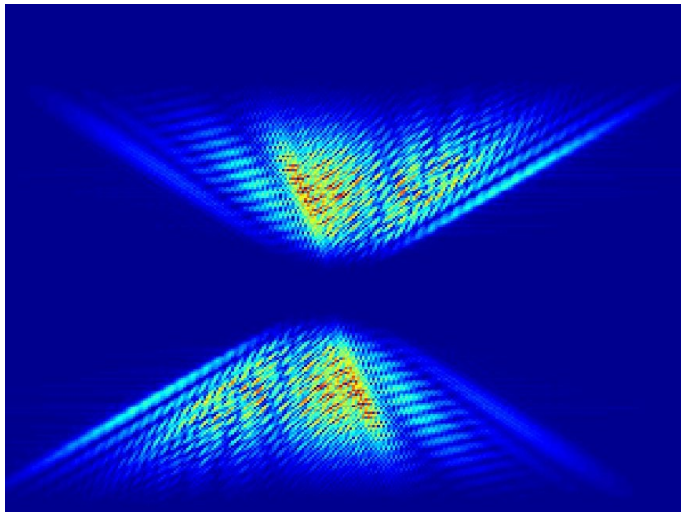
# The wave-packet decomposition algorithm; Review and flaws.

We begin with a function  $A$  to be decomposed:

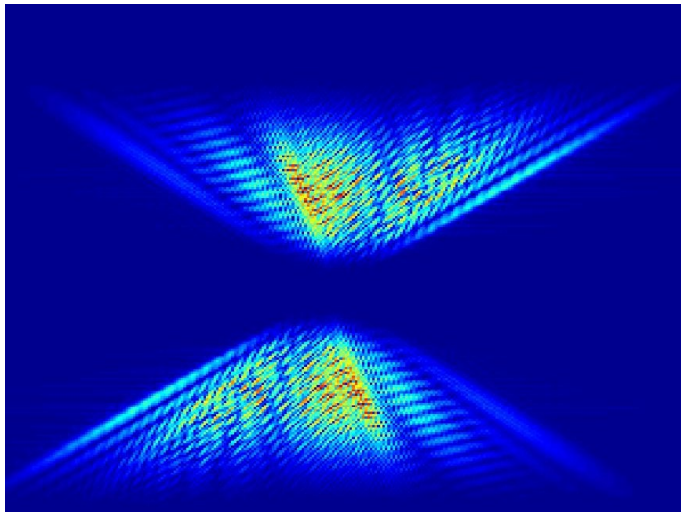


1 Fourier transform  $A(x)$  to get  $\hat{A}(\xi)$  :

$\hat{A}(\xi)$  :



- 1 Fourier transform  $A(x)$  to get  $\hat{A}(\xi)$  :
- 2 Dyadic parabolic partitioning of the " $\xi$ -domain";  
 $\hat{A}(\xi)$  :

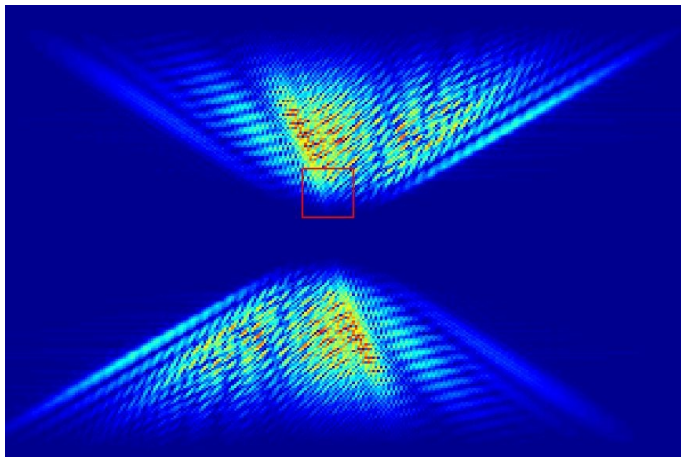


- 1 Fourier transform  $A(x)$  to get  $\hat{A}(\xi)$  :
- 2 Dyadic parabolic partitioning of the " $\xi$ -domain";

$B_{k,\nu}$  :

$k = 1$ ,

$\nu = i$

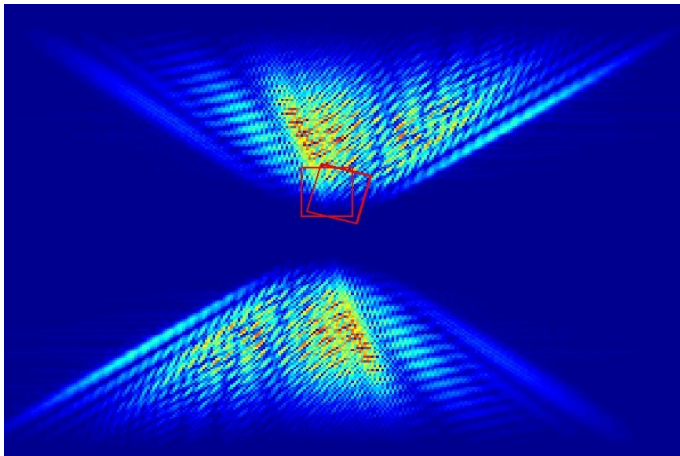


- 1 Fourier transform  $A(x)$  to get  $\hat{A}(\xi)$  :
- 2 Dyadic parabolic partitioning of the " $\xi$ -domain";

$B_{k,\nu}$  :

$k = 1,$

$\nu = \dots$

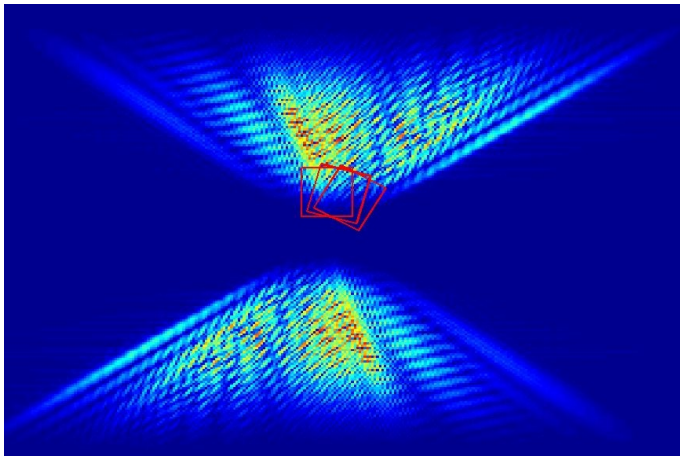


- 1 Fourier transform  $A(x)$  to get  $\hat{A}(\xi)$  :
- 2 Dyadic parabolic partitioning of the " $\xi$ -domain";

$B_{k,\nu}$  :

$k = 1,$

$\nu = \dots$

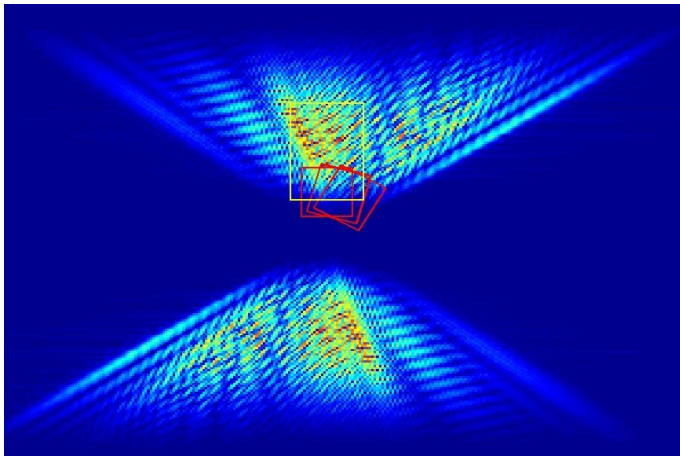


- 1 Fourier transform  $A(x)$  to get  $\hat{A}(\xi)$  :
- 2 Dyadic parabolic partitioning of the " $\xi$ -domain";

$B_{k,\nu}$  :

$k = 2$ ,

$\nu = i$

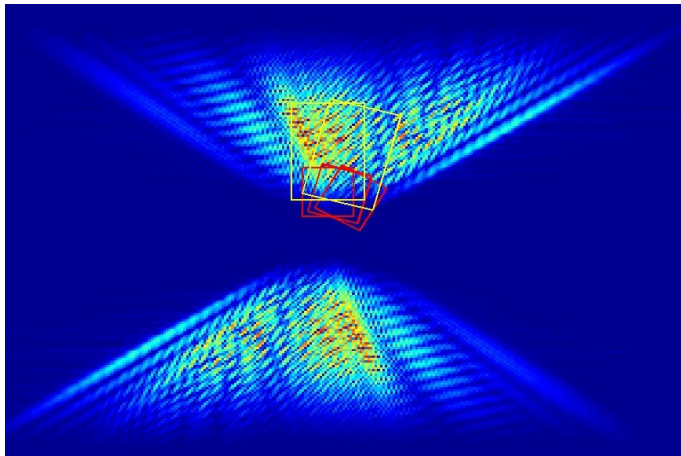


- 1 Fourier transform  $A(x)$  to get  $\hat{A}(\xi)$  :
- 2 Dyadic parabolic partitioning of the " $\xi$ -domain";

$B_{k,\nu}$  :

$k = 2,$

$\nu = \dots$

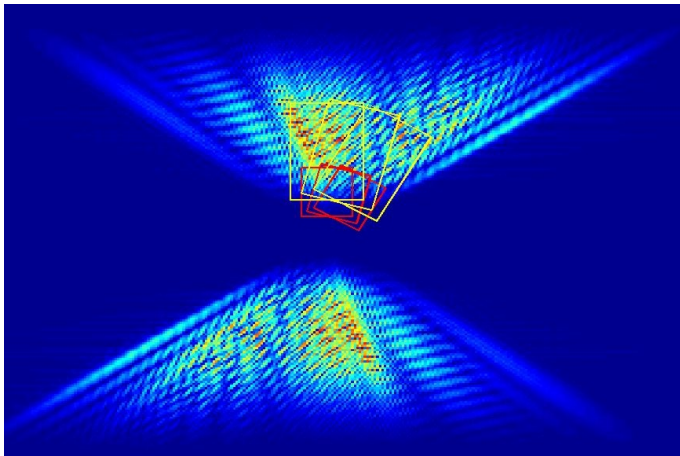


- 1 Fourier transform  $A(x)$  to get  $\hat{A}(\xi)$  :
- 2 Dyadic parabolic partitioning of the " $\xi$ -domain";

$B_{k,\nu}$  :

$k = 2,$

$\nu = \dots$

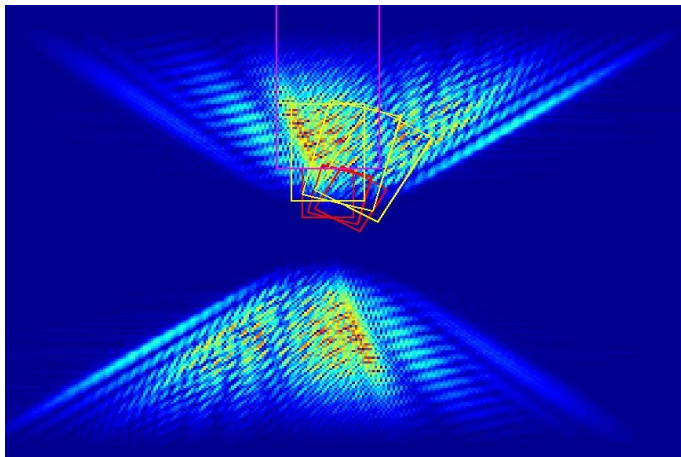


- 1 Fourier transform  $A(x)$  to get  $\hat{A}(\xi)$  :
- 2 Dyadic parabolic partitioning of the " $\xi$ -domain";

$B_{k,\nu}$  :

$k = 3$ ,

$\nu = i$



- 1 Fourier transform  $A(x)$  to get  $\hat{A}(\xi)$  :
- 2 Dyadic parabolic partitioning of the " $\xi$ -domain";
- 3 Construct a partition of unity  $\sum_{k,\nu} \phi_{k,\nu}^2 = 1$ , where  $\phi_{k,\nu}$  has support on the box  $B_{k,\nu}$ .

- 1 Fourier transform  $A(x)$  to get  $\hat{A}(\xi)$  :
- 2 Dyadic parabolic partitioning of the " $\xi$ -domain";
- 3 Construct a partition of unity  $\sum_{k,\nu} \phi_{k,\nu}^2 = 1$ , where  $\phi_{k,\nu}$  has support on the box  $B_{k,\nu}$ .
- 4 For each index  $k, \nu$  do:
  - 4.1 a Fourier series expansion of  $\phi_{k,\nu} \hat{A}$  on the box  $B_{k,\nu}$ ;

$$\phi_{k,\nu} \hat{A}(\xi) = \sum_{j \in \mathbb{Z}^2} a_{j,(k,\nu)} e^{i \langle \zeta_{j,(k,\nu)}, \xi \rangle}$$

- 4.2 Throw away small coefficients  $a_{j,(k,\nu)}$  to obtain

$$\phi_{k,\nu} \hat{A}(\xi) \approx \sum_{\text{Finite sum in } m} a_{j_m,(k,\nu)} e^{i \langle \zeta_{j_m,(k,\nu)}, \xi \rangle}$$

- 1 Fourier transform  $A(x)$  to get  $\hat{A}(\xi)$  :
- 2 Dyadic parabolic partitioning of the " $\xi$ -domain";
- 3 Construct a partition of unity  $\sum_{k,\nu} \phi_{k,\nu}^2 = 1$ , where  $\phi_{k,\nu}$  has support on the box  $B_{k,\nu}$ .
- 4 For each index  $k, \nu$  do:
  - 4.1 a Fourier series expansion of  $\phi_{k,\nu} \hat{A}$  on the box  $B_{k,\nu}$ ;

$$\phi_{k,\nu} \hat{A}(\xi) = \sum_{j \in \mathbb{Z}^2} a_{j,(k,\nu)} e^{i \langle \zeta_{j,(k,\nu)}, \xi \rangle}$$

- 4.2 Throw away small coefficients  $a_{j,(k,\nu)}$  to obtain

$$\phi_{k,\nu} \hat{A}(\xi) \approx \sum_{\text{Finite sum in } m} a_{j_m,(k,\nu)} e^{i \langle \zeta_{j_m,(k,\nu)}, \xi \rangle}$$

- 5 Sum up all approximations to obtain

$$\hat{A}(\xi) = \sum_{k,\nu} \phi_{k,\nu}^2 \hat{A}(\xi) \approx \sum_{\text{All } k,\nu} \sum_{\text{Finite in } m} a_{j_m,(k,\nu)} \phi_{k,\nu}(\xi) e^{i \langle \zeta_{j_m,(k,\nu)}, \xi \rangle}$$

# The wave-packet algorithm continued

4  $\phi_{k,\nu} \hat{A}(\xi) \approx \sum_{\text{Finite sum in } m} a_{jm,(k,\nu)} e^{i\langle \zeta_{jm,(k,\nu)}, \xi \rangle}$  (valid on  $B_{k,\nu}$ )

5  $\hat{A}(\xi) \approx \sum_{k,\nu} \sum_m a_{jm,(k,\nu)} \underbrace{\phi_{k,\nu}(\xi) e^{i\langle \zeta_{jm,(k,\nu)}, \xi \rangle}}_{\widehat{\psi_\gamma} - (\text{the wave-packets})}$  (valid in  $\mathbb{R}^2$ )

6  $\mathcal{F}^{-1}(\phi_{k,\nu}(\xi) e^{i\langle \zeta_{jm,(k,\nu)}, \xi \rangle}) = \psi_\gamma$  are the wave-packets, (where  $\gamma$  is some index specifying "generation"  $k$ , direction  $\nu$  and "Fourier exponent"  $\zeta_j$ .) With  $k$  and  $\nu$  fixed, different exponents  $\zeta_j$  yield translations (in  $x$ ) of a fixed wave-packet.

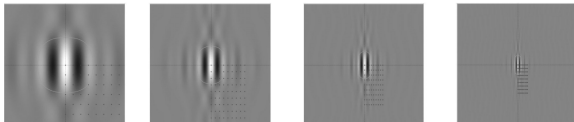
7 Applying  $\mathcal{F}^{-1}$  to 5 we obtain the wave-packet decomposition:

$$A(x) \approx \sum_{\text{Finite in } \gamma} a_\gamma \psi_\gamma(x)$$

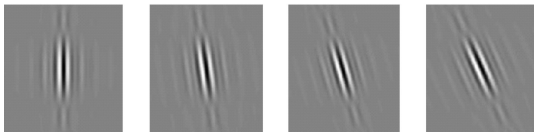
# Pictures and comments

★ For each generation  $k$  there is a "mother wave-packet". All other wave-packets are rotations and translations of this one.

different scales  $k$



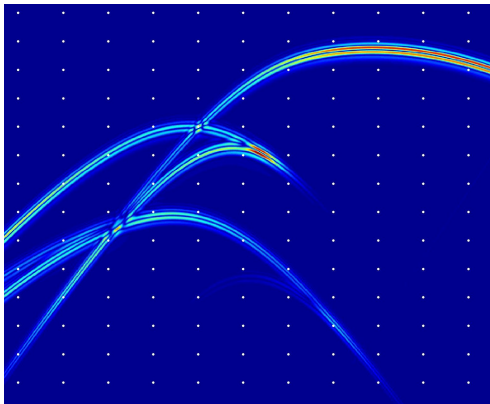
$k=4$ ; different rotations  $\nu$



# Pictures and comments

- ★ For each generation  $k$  there is a "mother wave-packet". All other wave-packets are rotations and translations of this one.
- ★ For each direction  $\nu$  the possible "centers" for the wave-packets lie on rotated grids.

$$k=1,$$
$$\nu = i$$

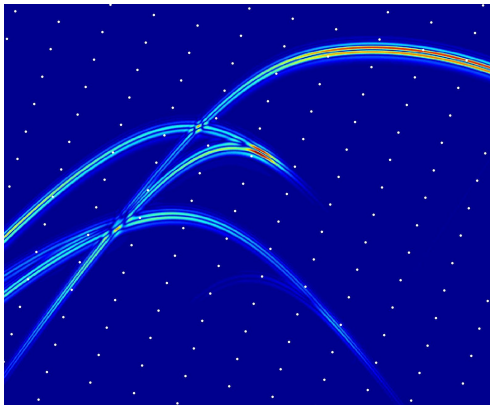


# Pictures and comments

- ★ For each generation  $k$  there is a "mother wave-packet". All other wave-packets are rotations and translations of this one.
- ★ For each direction  $\nu$  the possible "centers" for the wave-packets lie on rotated grids.

$k=1,$

$\nu = \dots$

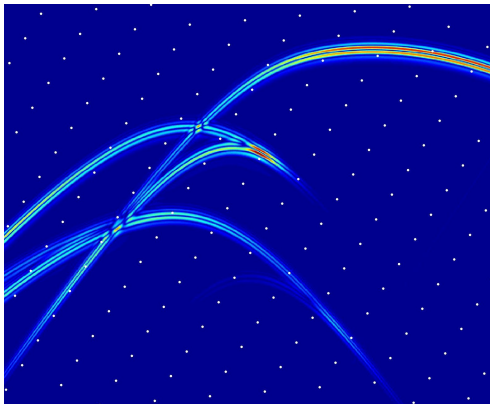


# Pictures and comments

- ★ For each generation  $k$  there is a "mother wave-packet". All other wave-packets are rotations and translations of this one.
- ★ For each direction  $\nu$  the possible "centers" for the wave-packets lie on rotated grids.

$k=1,$

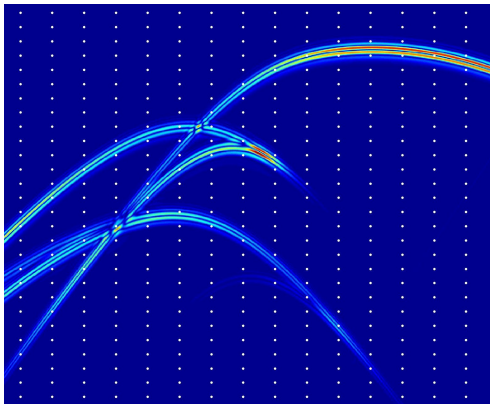
$\nu = \dots$



# Pictures and comments

- ★ For each generation  $k$  there is a "mother wave-packet". All other wave-packets are rotations and translations of this one.
- ★ For each direction  $\nu$  the possible "centers" for the wave-packets lie on rotated grids.

$$k=2,$$
$$\nu = i$$

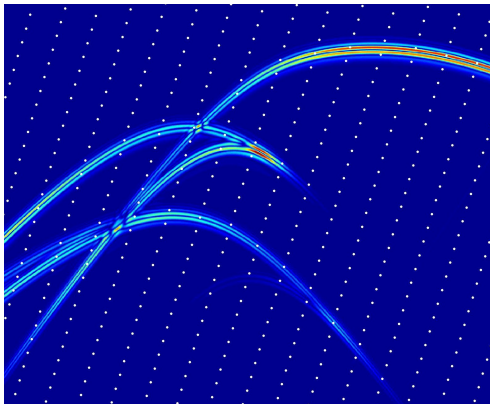


# Pictures and comments

- ★ For each generation  $k$  there is a "mother wave-packet". All other wave-packets are rotations and translations of this one.
- ★ For each direction  $\nu$  the possible "centers" for the wave-packets lie on rotated grids.

$k=2$ ,

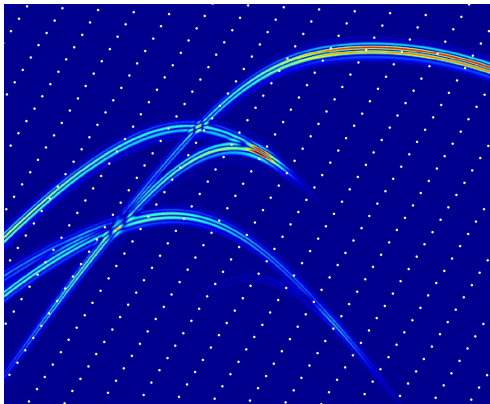
$\nu = \dots$



# Pictures and comments

- ★ For each generation  $k$  there is a "mother wave-packet". All other wave-packets are rotations and translations of this one.
- ★ For each direction  $\nu$  the possible "centers" for the wave-packets lie on rotated grids.

$k=2$ ,  
 $\nu = \dots$

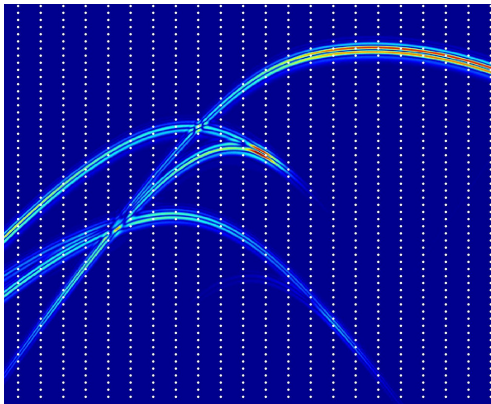


# Pictures and comments

★ For each generation  $k$  there is a "mother wave-packet". All other wave-packets are rotations and translations of this one.

★ For each direction  $\nu$  the possible "centers" for the wave-packets lie on rotated grids.

$k=3,$   
 $\nu = i$

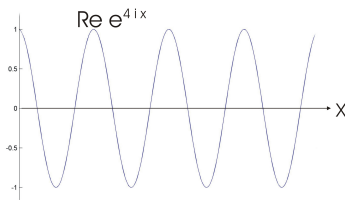
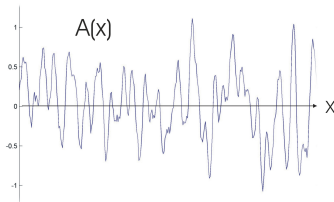


# Method 1. Thresholding the Fourier coefficients.

Given function  $A : [0, 2\pi] \rightarrow \mathbb{R}$  we wish to approximate it by a sum of exponential functions. Do:

- 1 Write  $A(x) = \sum_{k=-\infty}^{\infty} a_k e^{ikx}$
- 2 Let  $\epsilon > 0$  be threshold level.  
Throw away all  $a_k$ 's such that  $|a_k| < \epsilon$ . Let  $\{a_{k_j}\}_{j=1}^n$  be the remaining ones.
- 3 Voilà; a "sparse" approximation remains:

$$A(x) \approx \sum_{j=1}^n a_{k_j} e^{ik_j x}$$

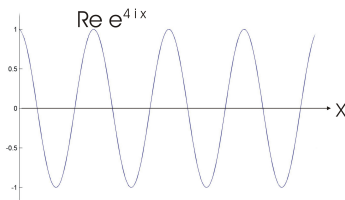
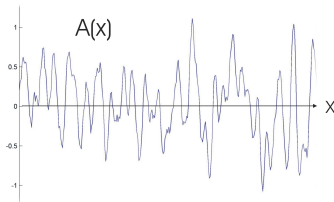


# Method 1. Thresholding the Fourier coefficients.

Given function  $A : [0, 2\pi] \rightarrow \mathbb{R}$  we wish to approximate it by a sum of exponential functions. Do:

- 1 Write  $A(x) = \sum_{k=-\infty}^{\infty} a_k e^{ikx}$
- 2 Let  $\epsilon > 0$  be threshold level.  
Throw away all  $a_k$ 's such that  $|a_k| < \epsilon$ . Let  $\{a_{k_j}\}_{j=1}^n$  be the remaining ones.
- 3 Voilà; a "sparse" approximation remains:

$$A(x) \approx \sum_{j=1}^n a_{k_j} e^{ik_j x}$$

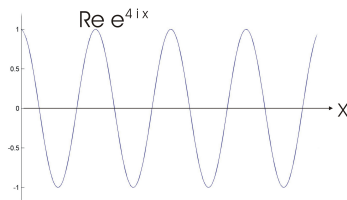
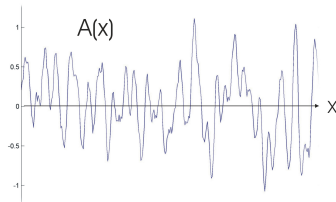


# Method 1. Thresholding the Fourier coefficients.

Given function  $A : [0, 2\pi] \rightarrow \mathbb{R}$  we wish to approximate it by a sum of exponential functions. Do:

- 1 Write  $A(x) = \sum_{k=-\infty}^{\infty} a_k e^{ikx}$
- 2 Let  $\epsilon > 0$  be threshold level.  
Throw away all  $a_k$ 's such that  $|a_k| < \epsilon$ . Let  $\{a_{k_j}\}_{j=1}^n$  be the remaining ones.
- 3 Voilà; a "sparse" approximation remains:

$$A(x) \approx \sum_{j=1}^n a_{k_j} e^{ik_j x}$$



# Thresholding the Fourier coefficients. Example

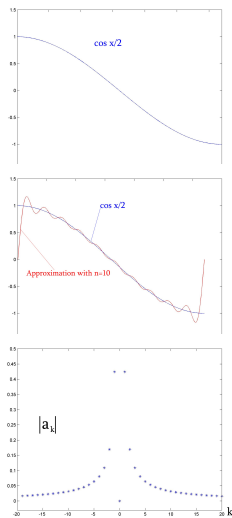
Lets take  $A(x) = \cos(x/2)$ . The fourier transform is

$$\cos(x/2) = \sum_{k=-\infty}^{\infty} \underbrace{\frac{-ik}{\pi(k-1/2)(k+1/2)}}_{a_k} e^{ikx}.$$

In this case,  $|a_k| \sim 1/k$  which decays very slowly. Hence either our approximation

$$\cos(x/2) \approx \sum_{k=-n}^n a_k e^{ikx}$$

will be a bad approximation or not so sparse.



# Need for improvement

Better idea: use  $e^{i\zeta_k x}$  with arbitrary  $\zeta_k \in \mathbb{R}$ , as opposed to  $e^{ikx}$  with  $k \in \mathbb{N}$ . For example, we would get

$$A(x) = \cos(x/2) = \frac{1}{2}e^{i\frac{1}{2}x} + \frac{1}{2}e^{i(-\frac{1}{2})x},$$

to be compared with  $\cos(x/2) \approx \sum_{k=-n}^n 2i \frac{k}{(k-1/2)(k+1/2)} e^{ikx}$ .

*But how to find suitable "nodes"  $\zeta_k \in \mathbb{R}$  to approximate a given function  $A$ ?*

# Need for improvement

Better idea: use  $e^{i\zeta_k x}$  with arbitrary  $\zeta_k \in \mathbb{R}$ , as opposed to  $e^{ikx}$  with  $k \in \mathbb{N}$ . For example, we would get

$$A(x) = \cos(x/2) = \frac{1}{2}e^{i\frac{1}{2}x} + \frac{1}{2}e^{i(-\frac{1}{2})x},$$

to be compared with  $\cos(x/2) \approx \sum_{k=-n}^n 2i \frac{k}{(k-1/2)(k+1/2)} e^{ikx}$ .

*But how to find suitable "nodes"  $\zeta_k \in \mathbb{R}$  to approximate a given function  $A$ ?*

# Need for improvement

Better idea: use  $e^{i\zeta_k x}$  with arbitrary  $\zeta_k \in \mathbb{R}$ , as opposed to  $e^{ikx}$  with  $k \in \mathbb{N}$ . For example, we would get

$$A(x) = \cos(x/2) = \frac{1}{2}e^{i\frac{1}{2}x} + \frac{1}{2}e^{i(-\frac{1}{2})x},$$

to be compared with  $\cos(x/2) \approx \sum_{k=-n}^n 2i \frac{k}{(k-1/2)(k+1/2)} e^{ikx}$ .

*But how to find suitable "nodes"  $\zeta_k \in \mathbb{R}$  to approximate a given function  $A$ ?*

# Overview of other approximation methods

- 1 Matching pursuit
- 2  $\ell^1$ -optimization

# Overview of other approximation methods

- 1 Matching pursuit
- 2  $\ell^1$ -optimization

# Overview of other approximation methods

- 1 Matching pursuit
- 2  $l^1$ -optimization

Common feature of 1 & 2: Both need large dictionaries  $\mathcal{D}$ .

E.g.  $\mathcal{D} = \{e^{i\frac{k}{1000}x}\}_{k=-\infty}^{\infty}$ .

# Overview of other approximation methods

- 1 Matching pursuit
- 2  $l^1$ -optimization
- 3 Toeplitz matrix approach. *By Beylkin and Monzón based on old results by Carathéodory.*

# Overview of other approximation methods

- 1 Matching pursuit
- 2  $l^1$ -optimization
- 3 Toeplitz matrix approach. *By Beylkin and Monzón based on old results by Carathéodory.*
- 4 Hankel matrix approach. (The AAK-algorithm). *By Beylkin and Monzón based on results by Adamyan, Arov and Krein.*

# Overview of other approximation methods

- 1 Matching pursuit
  - 2  $l^1$ -optimization
  - 3 Toeplitz matrix approach. *By Beylkin and Monzón based on old results by Carathéodory.*
  - 4 Hankel matrix approach. (The AAK-algorithm). *By Beylkin and Monzón based on results by Adamyan, Arov and Krein.*
- 3 yields approximations

$$A(x) \approx \sum_{k=1}^n a_k e^{i\zeta_k}$$

where  $\zeta_k \in \mathbb{R}$  is "chosen" by  $A$ .

4 is similar but produces  $\zeta_k \in \mathbb{C}$

# The 1 – d AAK-theorem.

$\Gamma_A : L^2(\mathbb{R}^+) \rightarrow L^2(\mathbb{R}^+)$  is given by

$$\Gamma_A(F) = \int_{\mathbb{R}^+} A(x+y)F(y)dy$$

$u_n/\sigma_n$  are singular vectors/values to  $\Gamma_A$ .  $\check{u}_n(\zeta) = \int_{\mathbb{R}^+} u_n(x)e^{ix\zeta}$ .

## Theorem

(AAK) Assume that  $\sigma_{n-1} > \sigma_n > \sigma_{n+1}$ . Then  $\check{u}_n$  has exactly  $n$  zeroes  $\zeta_1, \dots, \zeta_n \in \mathbb{C}^+$ , (counted with multiplicity). Moreover, there are coefficients  $c_1, \dots, c_n \in \mathbb{C}$  such that

$$\|\Gamma_{A - \sum_{k=1}^n c_k e^{i\zeta_k x}}\| = \sigma_n$$

# The 1 – d AAK-theorem.

$\Gamma_A : L^2(\mathbb{R}^+) \rightarrow L^2(\mathbb{R}^+)$  is given by

$$\Gamma_A(F) = \int_{\mathbb{R}^+} A(x+y)F(y)dy$$

$u_n/\sigma_n$  are singular vectors/values to  $\Gamma_A$ .  $\check{u}_n(\zeta) = \int_{\mathbb{R}^+} u_n(x)e^{ix\zeta}$ .

## Theorem

(AAK) Assume that  $\sigma_{n-1} > \sigma_n > \sigma_{n+1}$ . Then  $\check{u}_n$  has exactly  $n$  zeroes  $\zeta_1, \dots, \zeta_n \in \mathbb{C}^+$ , (counted with multiplicity). Moreover, there are coefficients  $c_1, \dots, c_n \in \mathbb{C}$  such that

$$\|\Gamma_{A - \sum_{k=1}^n c_k e^{i\zeta_k x}}\| = \sigma_n$$

(Kronecker)  $\text{Rank}(\Gamma_A) = 1$  iff  $a = e^{\zeta x}$  for some  $\zeta \in \mathbb{C}^+$

# The 1 - d AAK-theorem.

$\Gamma_A : L^2(\mathbb{R}^+) \rightarrow L^2(\mathbb{R}^+)$  is given by

$$\Gamma_A(F) = \int_{\mathbb{R}^+} A(x+y)F(y)dy$$

$u_n/\sigma_n$  are singular vectors/values to  $\Gamma_A$ .  $\check{u}_n(\zeta) = \int_{\mathbb{R}^+} u_n(x)e^{ix\zeta}$ .

## Theorem

(AAK) Assume that  $\sigma_{n-1} > \sigma_n > \sigma_{n+1}$ . Then  $\check{u}_n$  has exactly  $n$  zeroes  $\zeta_1, \dots, \zeta_n \in \mathbb{C}^+$ , (counted with multiplicity). Moreover, there are coefficients  $c_1, \dots, c_n \in \mathbb{C}$  such that

$$\|\Gamma_{A - \sum_{k=1}^n c_k e^{i\zeta_k x}}\| = \sigma_n$$

(Kronecker)  $\text{Rank}(\Gamma_A) = 1$  iff  $a = e^{\zeta x}$  for some  $\zeta \in \mathbb{C}^+$

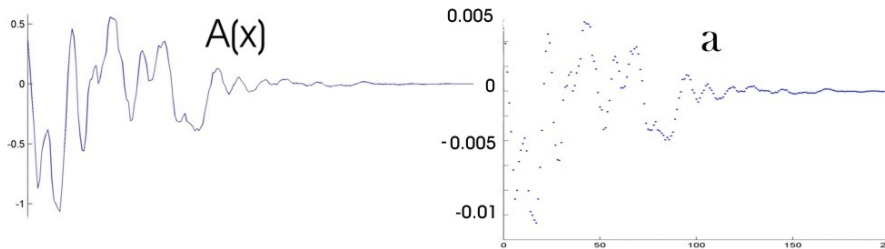
## Note

$$\sigma_n = \inf\{\|\Gamma_a - K\| : \text{Rank}(K) = n\}.$$

# Outline of the algorithm

- 1 Sample the function  $A$  with interval  $1/N$ , ( $N \in \mathbb{N}$ ), to get vector  $\mathcal{S}_N A = a$ :

$$\mathcal{S}_N A = (a_0, a_1, \dots, a_M, 0, 0, 0, \dots) = \left( \frac{1}{N} A\left(\frac{k}{N}\right) \right)_{k=0}^{\infty}.$$



# Outline of the algorithm

- 1 Sample the function  $A$  with interval  $1/N$ , ( $N \in \mathbb{N}$ ), to get vector  $\mathcal{S}_N A = a$ :

$$\mathcal{S}_N A = (a_0, a_1, \dots, a_M, 0, 0, 0, \dots) = \left( \frac{1}{N} A \left( \frac{k}{N} \right) \right)_{k=0}^{\infty}.$$

- 2 Form the finite Hankel matrix

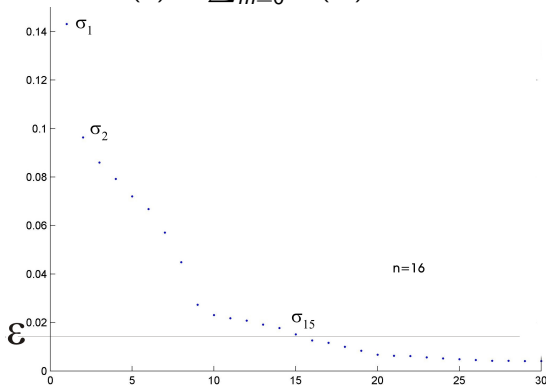
$$\Gamma_a = \begin{pmatrix} a_0 & a_1 & a_2 & \dots & a_M \\ a_1 & a_2 & \cdot & \dots & 0 \\ a_2 & \cdot & \cdot & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_M & 0 & 0 & \dots & 0 \end{pmatrix}$$

- 3 Let  $\sigma_0 \geq \sigma_1 \geq \dots \geq \sigma_M$  be its singular values and  $u_0, \dots, u_M$  the singular vectors.

1. Sample  $A \mapsto a \longrightarrow$ , 2. Hankel matrix  $\Gamma_a, \longrightarrow$
3. Sing. value's  $\sigma_k$  & sing. vector's  $u_k$ )

4 Take  $\sigma_n$  to be the first value under the desired accuracy  $\epsilon$ .

Put  $\check{u}_n(z) = \sum_{m=0}^M u_n(m) z^m$ .



## Theorem

(AAK) Assume that  $\sigma_{n-1} > \sigma_n > \sigma_{n+1}$ . Then  $\check{u}_n(e^{\frac{i}{N}\zeta})$  has exactly  $n$  zeroes  $\zeta_1, \dots, \zeta_n \in \mathbb{C}^+$  (counted with multiplicity). Moreover

- (1. Sample  $A \mapsto a \longrightarrow$ , 2. Hankel matrix  $\Gamma_a, \longrightarrow$   
 3. Sing. value's  $\sigma_k$  & sing. vector's  $u_k$ )

$$4 \quad \sigma_n < \epsilon < \sigma_{n-1} \\
 \check{u}_n(z) = \sum_{m=0}^M u_n(m) z^m.$$

## Theorem

(AAK) Assume that  $\sigma_{n-1} > \sigma_n > \sigma_{n+1}$ . Then  $\check{u}_n(e^{\frac{i}{N}\zeta})$  has exactly  $n$  zeroes  $\zeta_1, \dots, \zeta_n \in \mathbb{C}^+$ , (counted with multiplicity). Moreover, there are coefficients  $c_1, \dots, c_n \in \mathbb{C}$  such that, with

$$A_{ap}(x) = \sum_{k=1}^n c_k e^{i\zeta_k x},$$

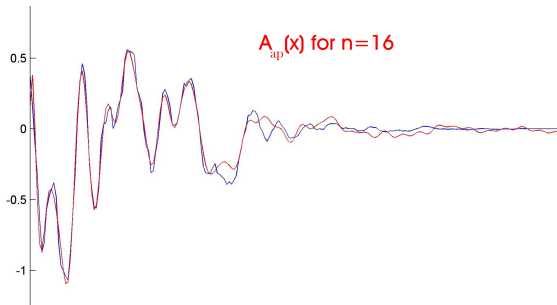
we have

$$\|\Gamma_{\mathcal{S}_N(A-A_{ap})}\| = \sigma_n$$

5 Compute the  $\zeta_1, \dots, \zeta_n$ 's and  $c_1, \dots, c_n$ 's; Then

$$A_{ap}(x) = \sum_{k=1}^n c_k e^{i\zeta_k x}$$

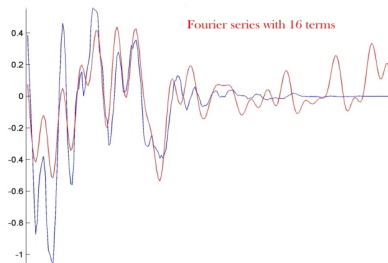
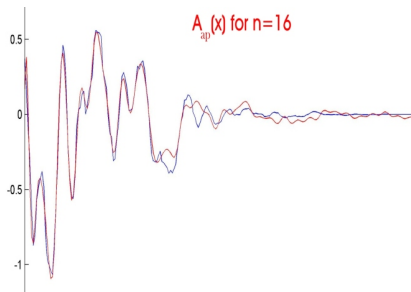
satisfies  $\|\Gamma_{\mathcal{S}_N}(A - A_{ap})\| = \sigma_n \leq \epsilon$ .



5 Compute the  $\zeta_1, \dots, \zeta_n$ 's and  $c_1, \dots, c_n$ 's; Then

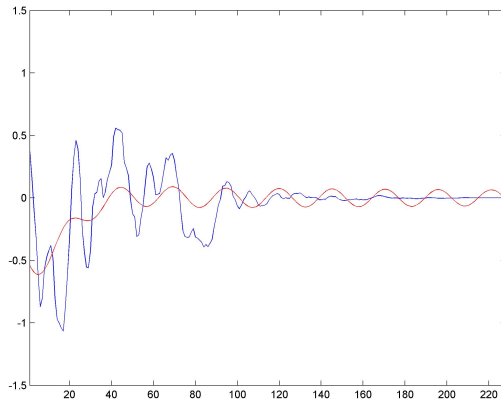
$$A_{ap}(x) = \sum_{k=1}^n c_k e^{i\zeta_k x}$$

satisfies  $\|\Gamma_{\mathcal{S}_N}(A - A_{ap})\| = \sigma_n \leq \epsilon$ .



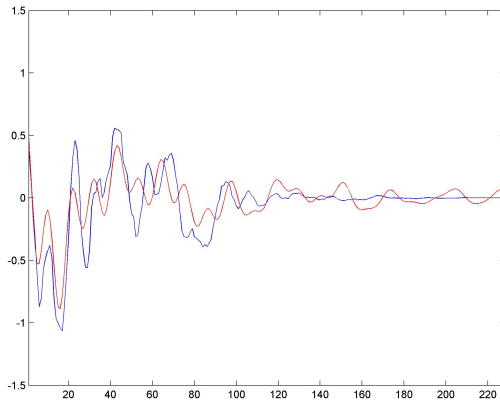
# Examples

$n=3$ :



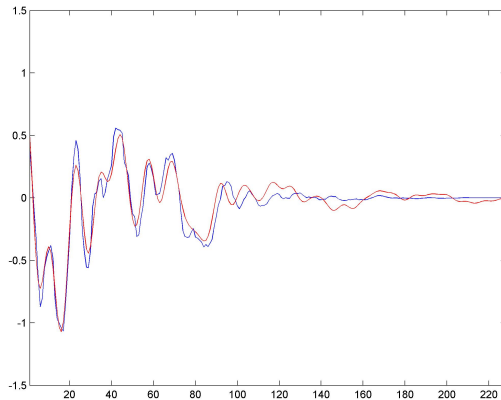
# Examples

$n=7$ :



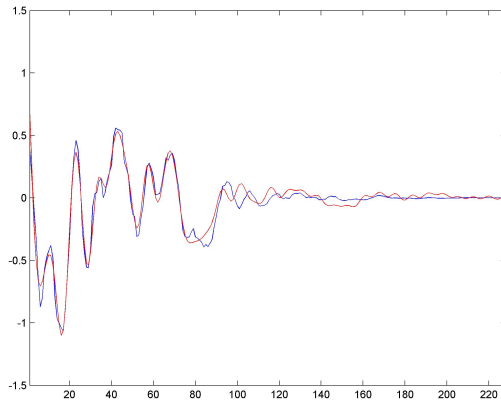
# Examples

$n=11$ :



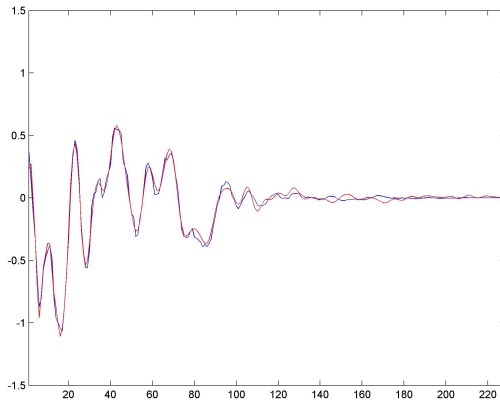
# Examples

$n=15$ :



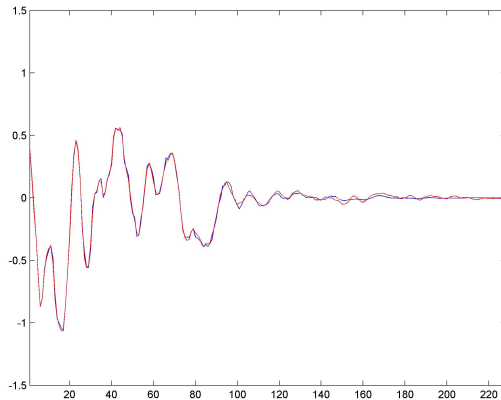
# Examples

$n=19$ :



# Examples

$n=29$ :



## 2-d Hankel operators

In 1 - d,

$$(\Gamma_a f)(k) = \sum_{j \geq 0} a_{k+j} f_j.$$

In 2 - d we thus set

$$(\Gamma_a f)(k_1, k_2) = \sum_{j_1, j_2 \geq 0} a_{(k_1+j_1, k_2+j_2)} f_{(j_1, j_2)}.$$

## 2-d Hankel operators

In 1 –  $d$ ,

$$(\Gamma_a f)(k) = \sum_{j \geq 0} a_{k+j} f_j.$$

In 2 –  $d$  we thus set

$$(\Gamma_a f)(k_1, k_2) = \sum_{j_1, j_2 \geq 0} a_{(k_1+j_1, k_2+j_2)} f_{(j_1, j_2)}.$$

Bad news:

- 1 AAK-theorem for 2 –  $d$  is unknown.
- 2 The Fourier transform of singular vectors  $u_n$  to a 2 –  $d$  Hankel operator is

$$\check{u}_n(z_1, z_2) = \sum u_n(k_1, k_2) z_1^{k_1} z_2^{k_2}$$

## 2-d Hankel operators

In 1 – d,

$$(\Gamma_a f)(k) = \sum_{j \geq 0} a_{k+j} f_j.$$

In 2 – d we thus set

$$(\Gamma_a f)(k_1, k_2) = \sum_{j_1, j_2 \geq 0} a_{(k_1+j_1, k_2+j_2)} f_{(j_1, j_2)}.$$

Good news:

For  $A \in C([0, 1]^2)$  put

$$\mathcal{S}_N(A) = A\left(\frac{k_1}{N}, \frac{k_2}{N}\right), \quad 0 \leq k_1, k_2 \leq N.$$

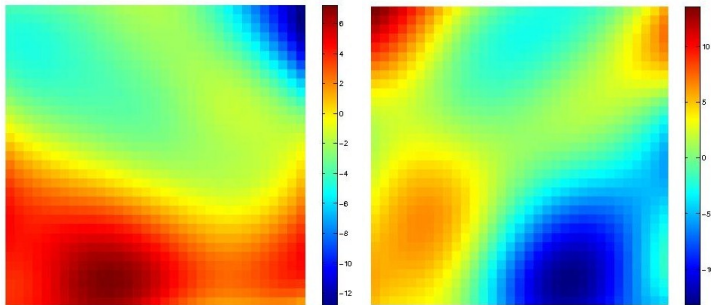
Then

$$\text{Rank} \Gamma_{\mathcal{S}_N(A)} = 1 \Leftrightarrow A(x_1, x_2) = c e^{i(\zeta_1 x_1 + \zeta_2 x_2)}, \quad c, \zeta_1, \zeta_2 \in \mathbb{C}.$$

We wish to approximate the function  $A$  :

Re  $A$ :

Im  $A$ :

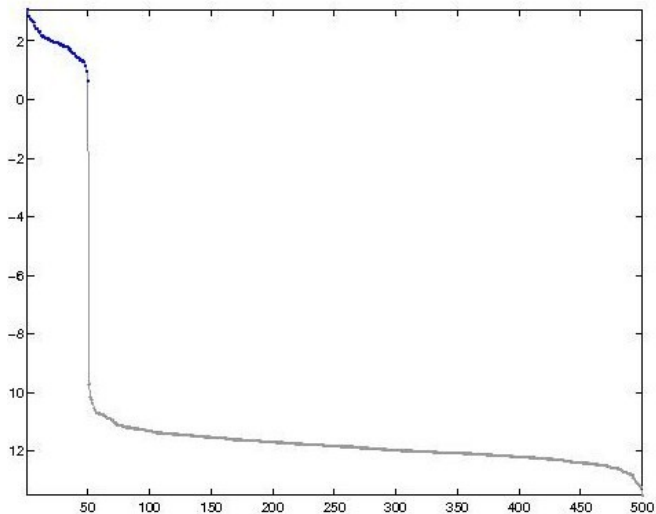


Using a fixed singular vector  $u_n$  to  $\Gamma_{S_n(A)}$ , the algorithm yields way too many points  $\{(\zeta_1^k, \zeta_2^k)\}_k$  such that  $e^{i(\zeta_1^k x_1 + \zeta_2^k x_2)}$  might be useful for approximating  $A$ . We solve

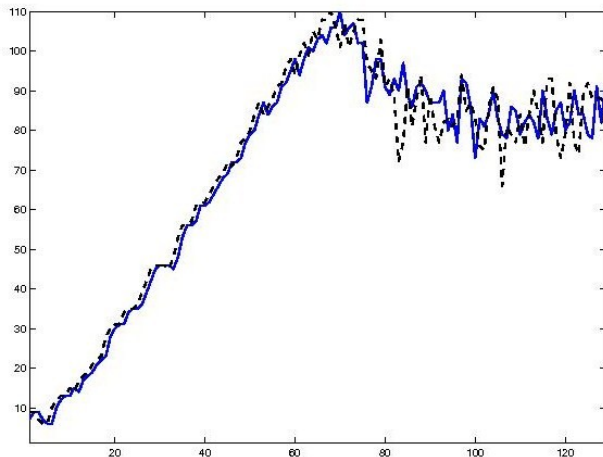
$$A \approx \sum_k a_k e^{i(\zeta_1^k x_1 + \zeta_2^k x_2)}$$

using the least squares method.

For  $n = 36$ , here is the result:

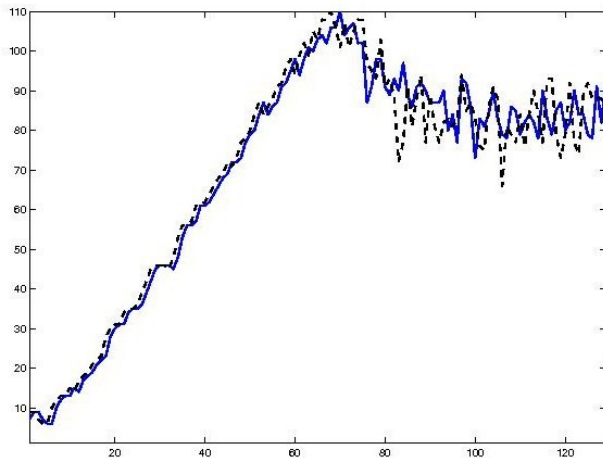


$N$  (the number of significant nodes) **versus**  $n$  (the number of the singular vector  $u_n$ ).

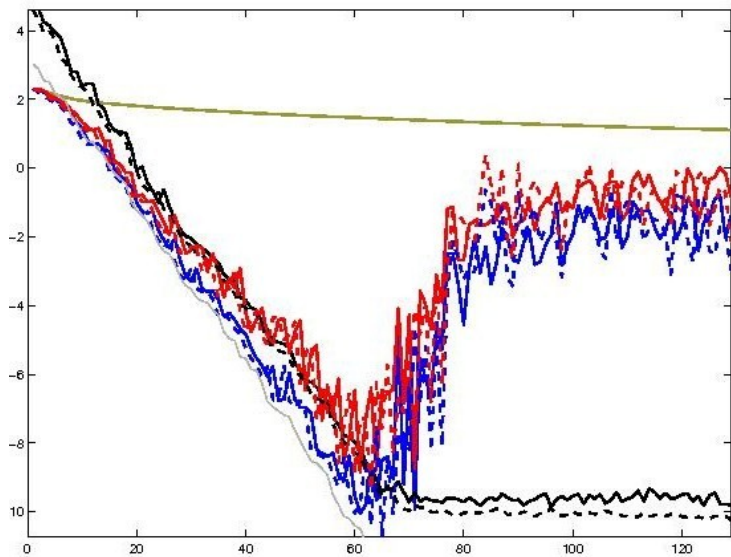


What about the approximation error as function of  $n$ ?

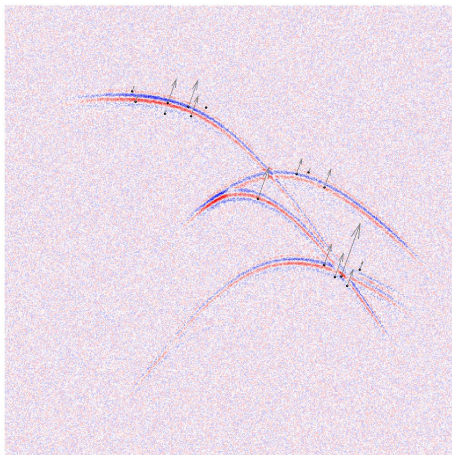
$N$  (the number of significant nodes) versus  $n$  (the number of the singular vector  $u_n$ ).



What about the approximation error as function of  $n$ ?



Stability with respect to noise:



Stability with respect to noise:

