

# Compressed Wavefield Extrapolation with Curvelets



Tim T.Y. Lin and Felix J. Herrmann  
University of British Columbia

SEG 2007

San Antonio, Sept 25

# Introduction

---

- Concerned with *explicit* forms of wavefield propagator **w** of the linearized forward model

$$\text{P}^- = \sum_{x_3 > 0} \text{W}^- \text{R}^+ \text{W}^+ \text{S}^+ \Delta x_3$$

- Would like to find explicit **w** suitable for wave-equation migration:
  - simultaneously operates on sets of traces
  - fully incorporates velocity information of medium
  - no parabolic approximations

# Introduction

- Goal: employ the complete 1-Way Helmholtz operator for **W**

Grimbergen, J., F. Dessing, and C. Wapenaar, 1998, Modal expansion of one-way operator on laterally varying media: Geophysics, **63**, 995–1005.

$$\mathbf{W}^{\pm} = e^{\mp j \Delta x \mathbf{H}_1} \quad \mathbf{H}_2 = \mathbf{H}_1 \mathbf{H}_1$$

- Problem: computation & storage complexity
  - creating and storing  $\mathbf{H}_2$  is trivial
  - however  $\mathbf{H}_1$  is *not* trivial to compute and store

$$\mathbf{H}_2 = \begin{bmatrix} \diagdown \end{bmatrix}$$

$$\mathbf{H}_1 = \begin{bmatrix} | & | & | & | & | \end{bmatrix}$$

# Introduction

- In this case **w** is computed by eigenvalue decomposition

$$\mathbf{H}_2 = \mathbf{L}\mathbf{\Lambda}\mathbf{L}^T = \begin{bmatrix} | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \end{bmatrix} \begin{bmatrix} \diagdown \\ \diagup \end{bmatrix} \begin{bmatrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{bmatrix}$$

$\mathbf{L} \quad \mathbf{\Lambda} \quad \mathbf{L}^T$

$$\mathbf{W}^\pm = \begin{bmatrix} | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \end{bmatrix} \begin{bmatrix} \diagdown \\ \diagup \end{bmatrix} \begin{bmatrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{bmatrix}$$

$\mathbf{L} \quad e^{-j\sqrt{\mathbf{\Lambda}}\Delta x_3} \quad \mathbf{L}^T$

- requires, per frequency:
  - 1 eigenvalue problem ( $O(n^4)$ )
  - 2 full matrix-vector for eigenspace transform ( $O(n^2)$ )

# Introduction

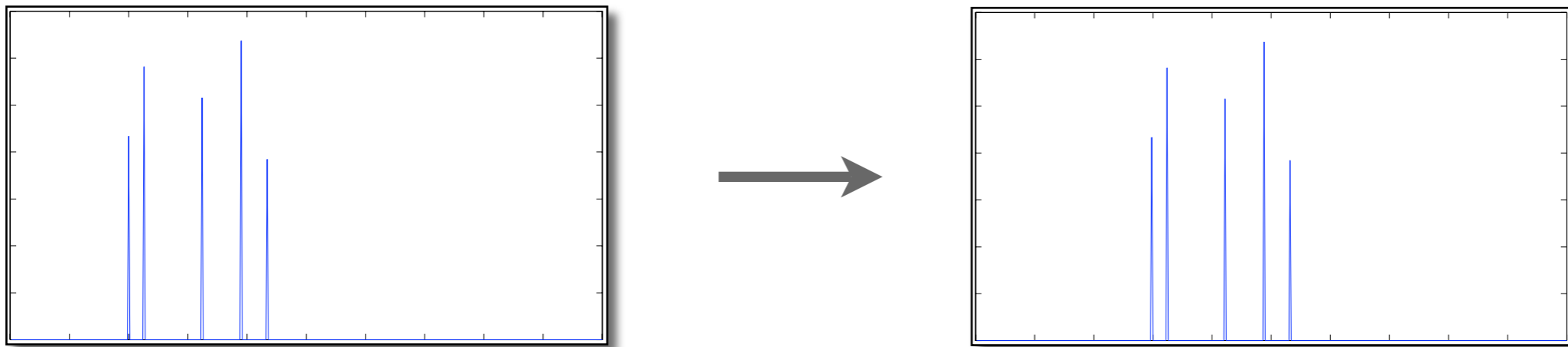
---

- Band-diagonalization techniques like parabolic approximation trades for speed with **approximations**
- *Is there another way?*

# Our approach

---

- Consider a related, but simpler problem: shifting (or translating) signal



- operator is  $\mathbf{S} = e^{-j \frac{\Delta x}{2\pi}} \mathbf{D}$
- $\mathbf{D}$  is differential operator

$$\mathbf{D} = \begin{bmatrix} \text{---} \end{bmatrix}$$

# Our approach

- Computation requires similar approach to  $\mathbf{w}^\pm$

$$\mathbf{D} = \mathbf{L}\mathbf{\Lambda}\mathbf{L}^T = \begin{bmatrix} | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \end{bmatrix} \begin{bmatrix} \diagdown \\ \diagup \end{bmatrix} \begin{bmatrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{bmatrix}$$

$\mathbf{L} \qquad \mathbf{\Lambda} \qquad \mathbf{L}^T$

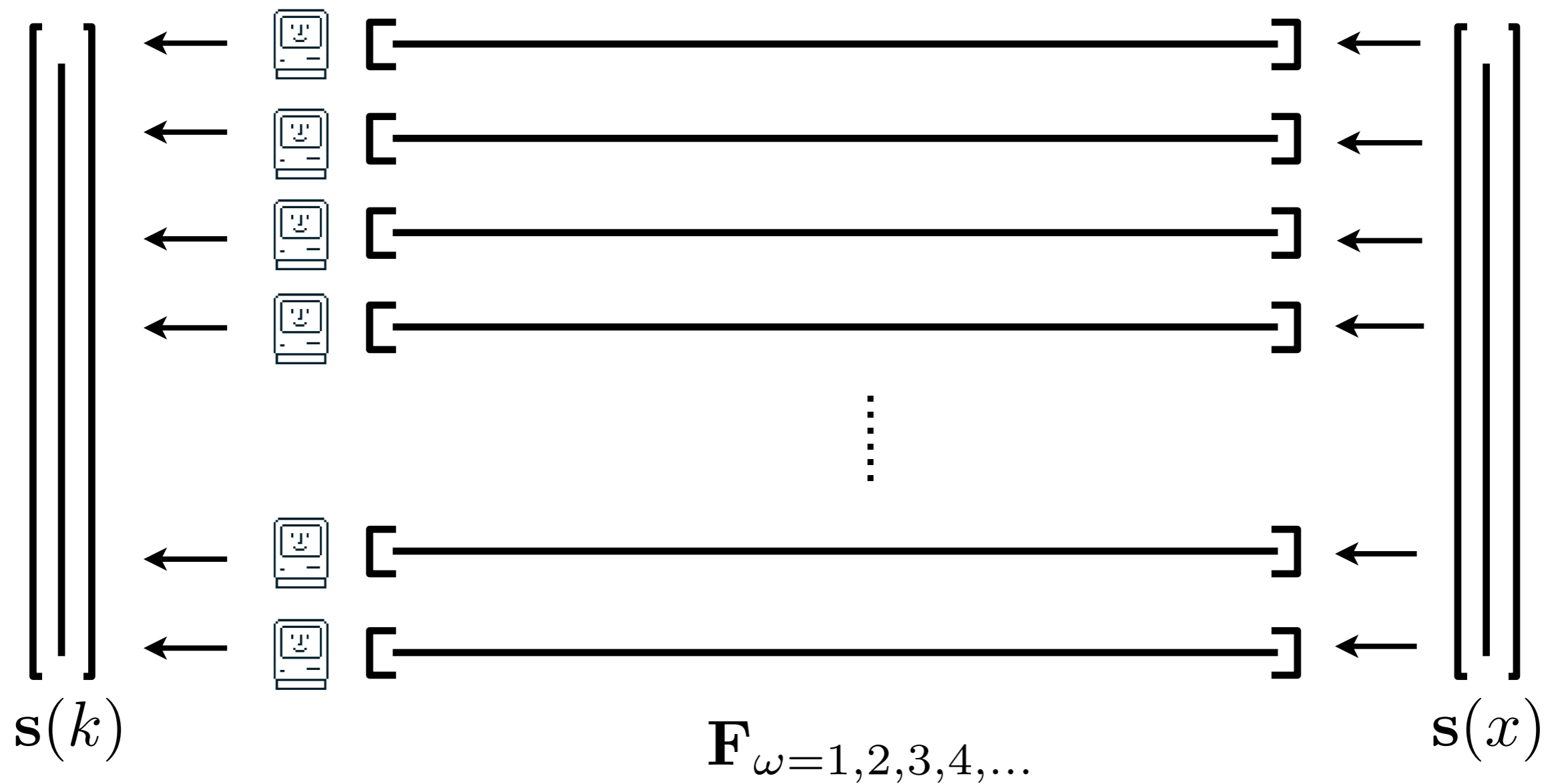
$$\mathbf{S} = \begin{bmatrix} | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \end{bmatrix} \begin{bmatrix} \diagdown \\ \diagup \end{bmatrix} \begin{bmatrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{bmatrix}$$

$\mathbf{L} \qquad e^{-j \frac{\Delta x}{2\pi}} \mathbf{\Lambda} \qquad \mathbf{L}^T$

- However, for  $\mathbf{D}$ ,  $\mathbf{L} = \text{DFT}$ , so computation trivial with FFT

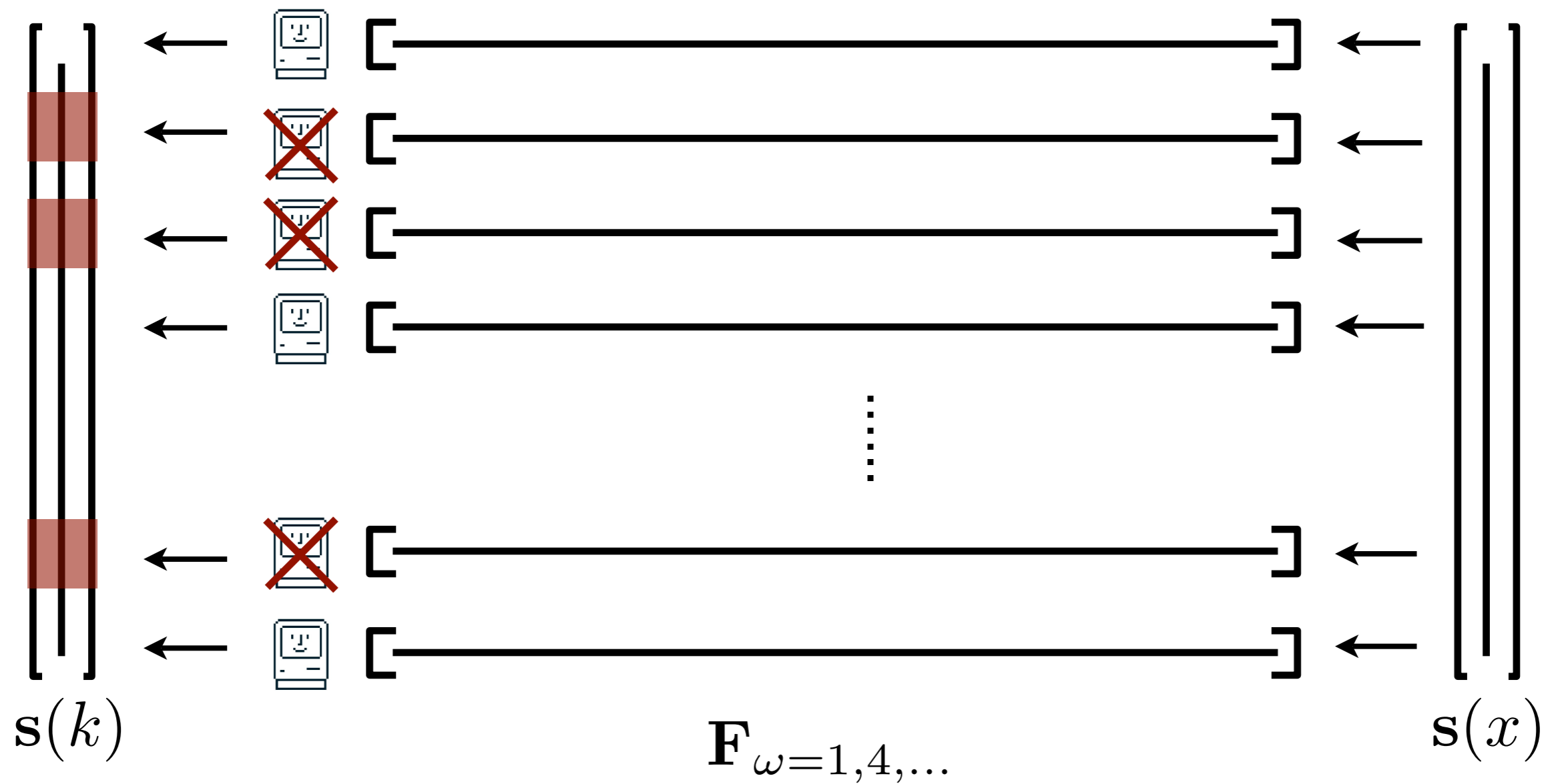
# Our approach

- Suppose FFT does not exist yet



# Our approach

- suppose some nodes didn't finish their jobs



# Our approach

---

- mathematically, the system is incomplete

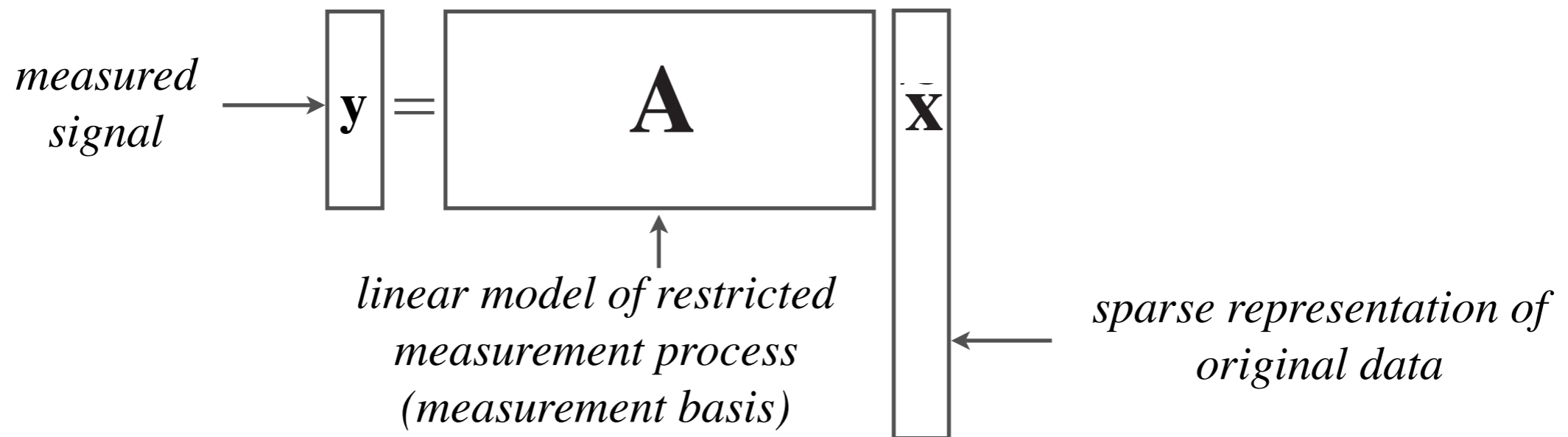
$$\begin{array}{c} \boxed{\phantom{s(k)}} \\ s(k) \end{array} = \boxed{\mathbf{F}_{\omega=1,4,\dots}} \begin{array}{c} \boxed{\phantom{s(x)}} \\ s(x) \end{array}$$

- evidently some information of original  $s(x)$  is invariably lost. *Or is it?*

# Compressed Sensing

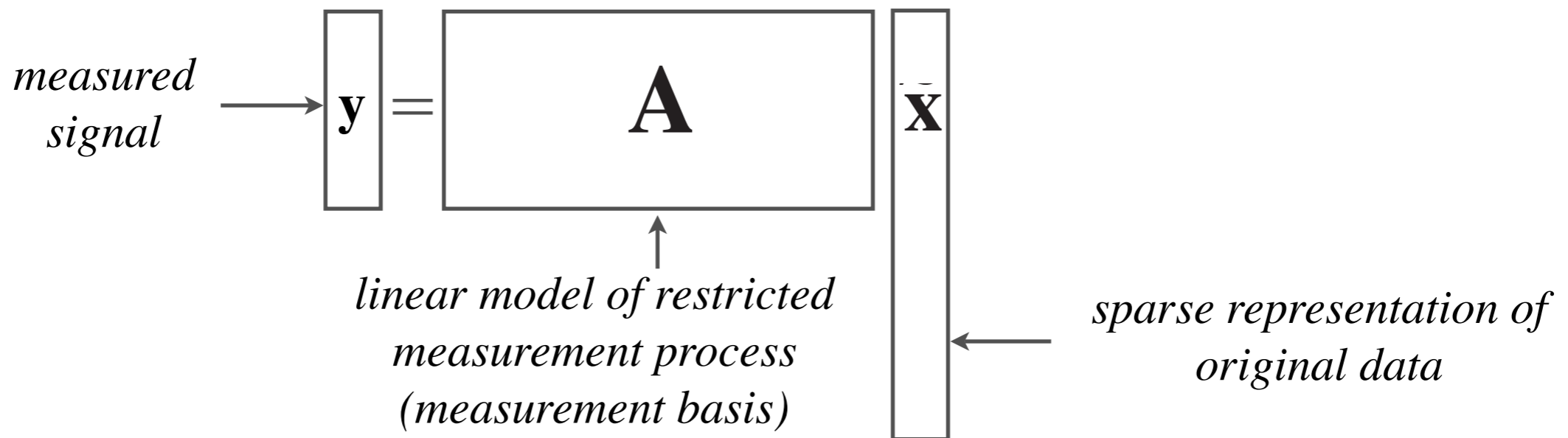
---

- states that given system of the form



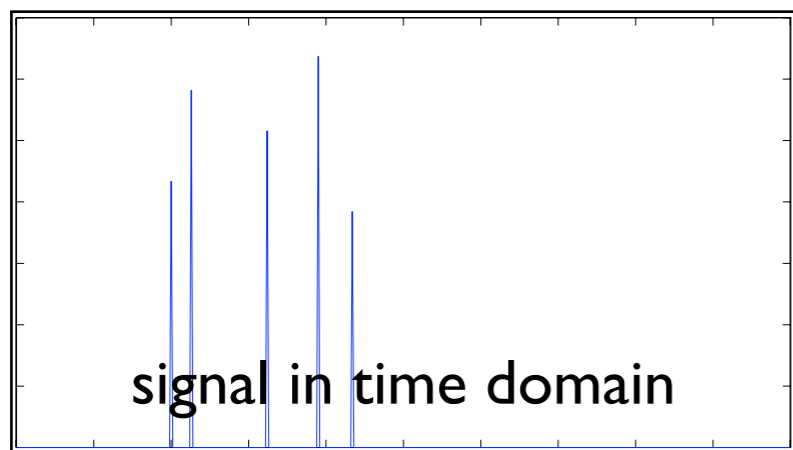
# Compressed Sensing

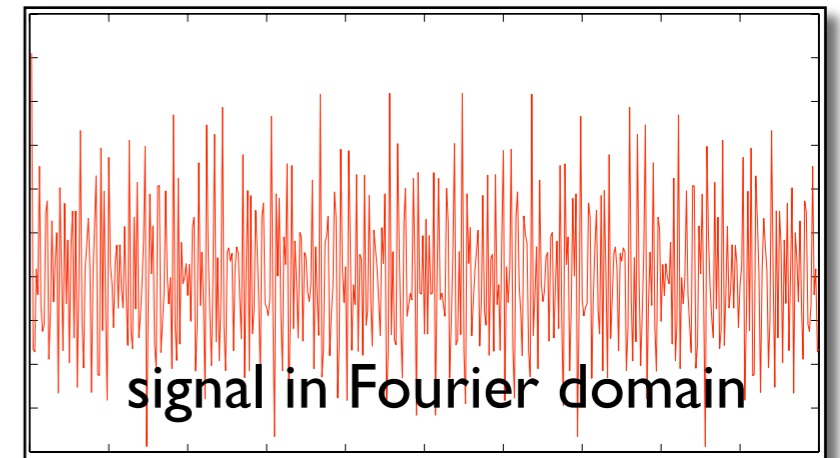
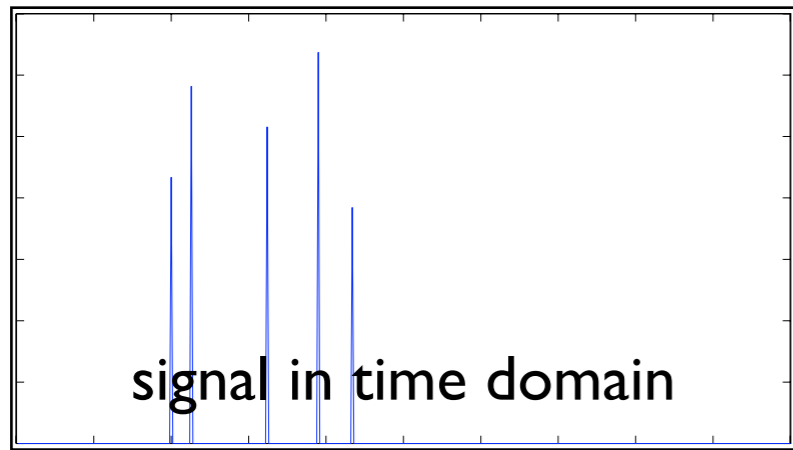
- states that given system of the form

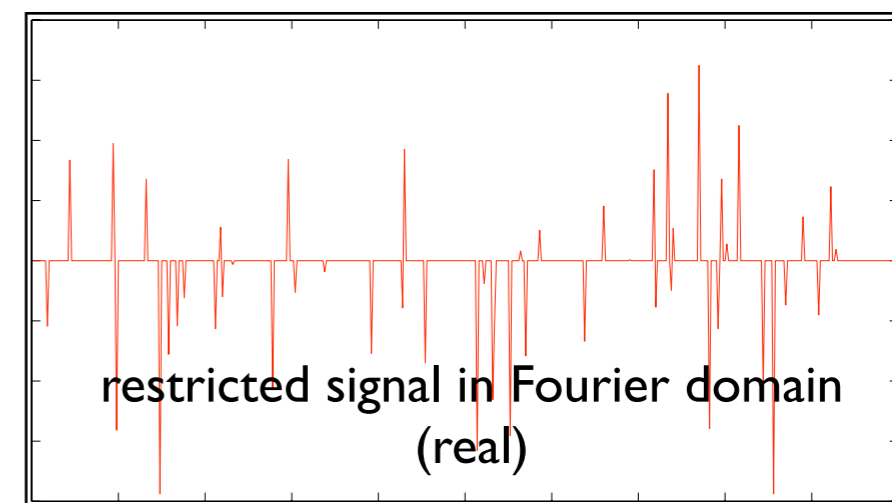
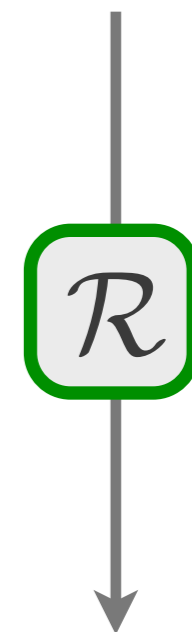
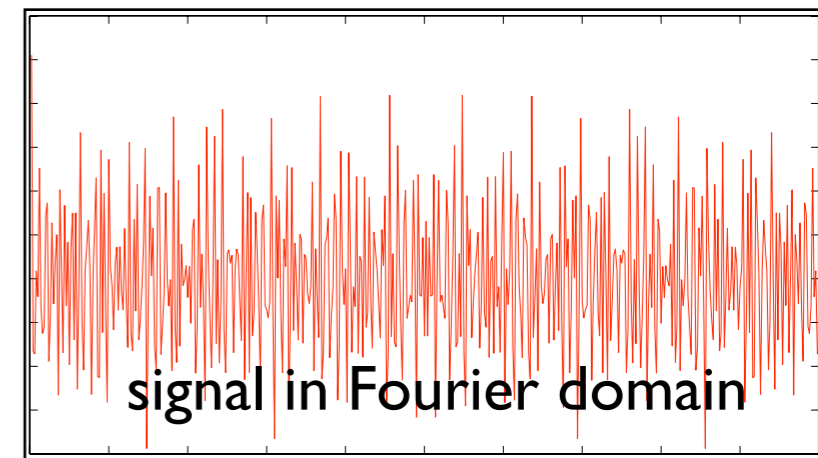
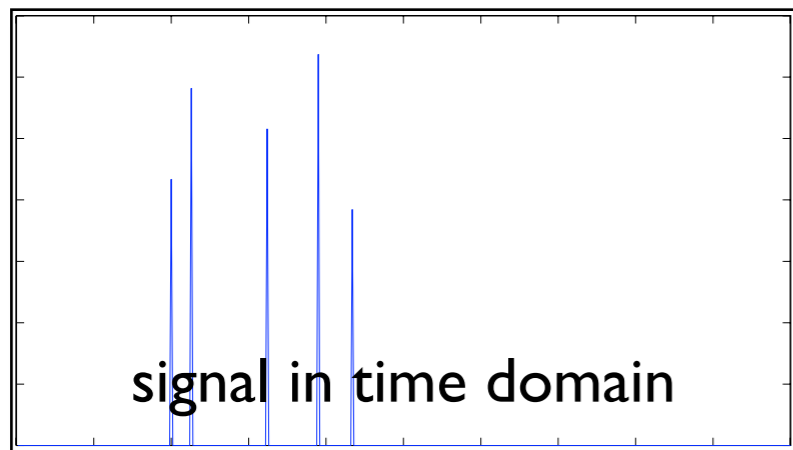


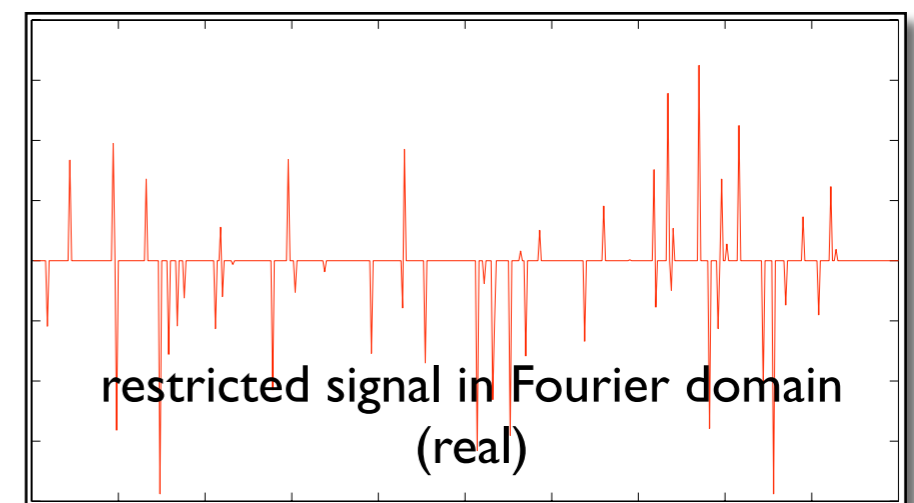
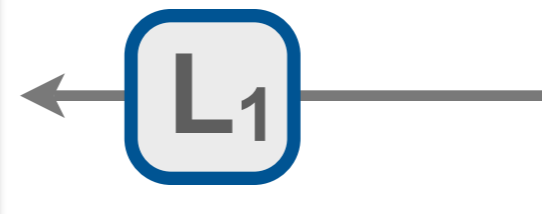
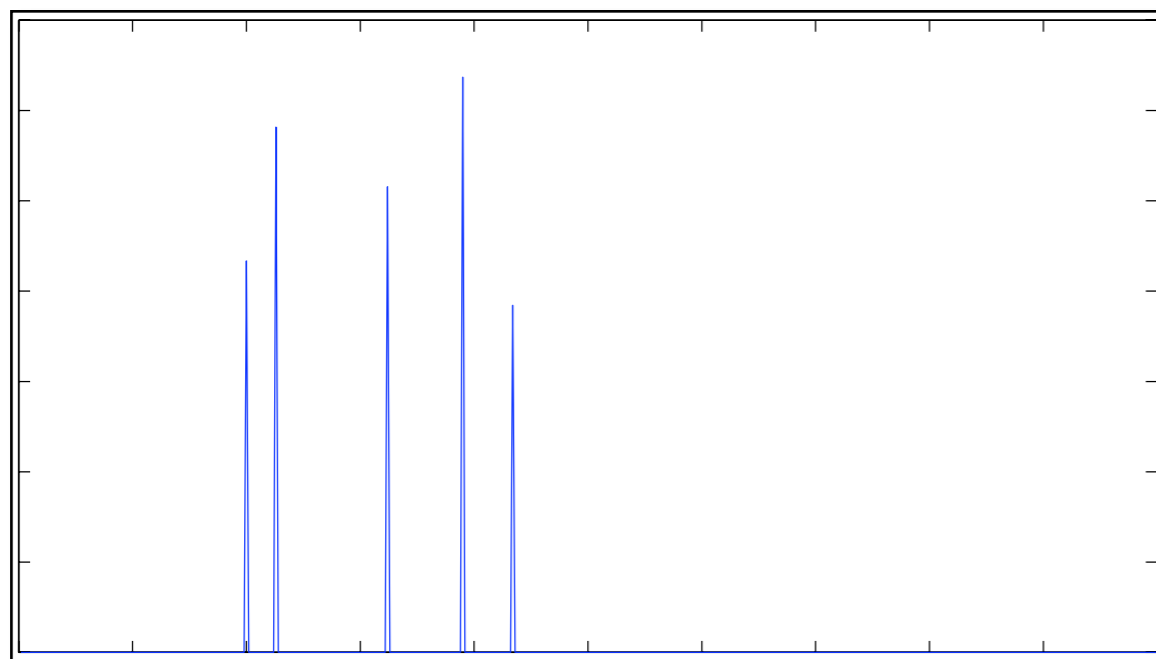
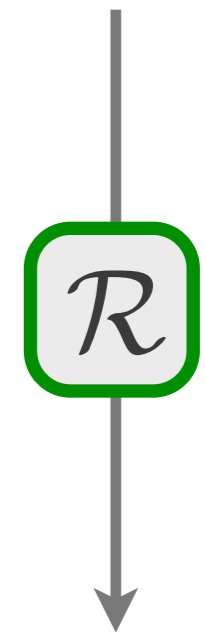
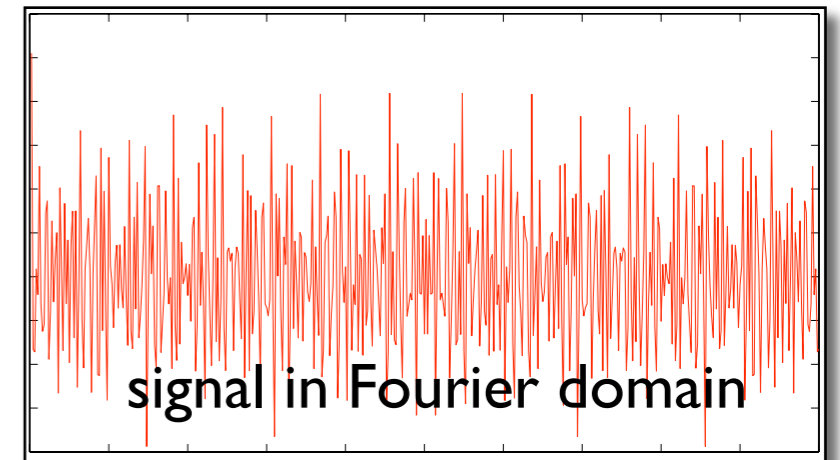
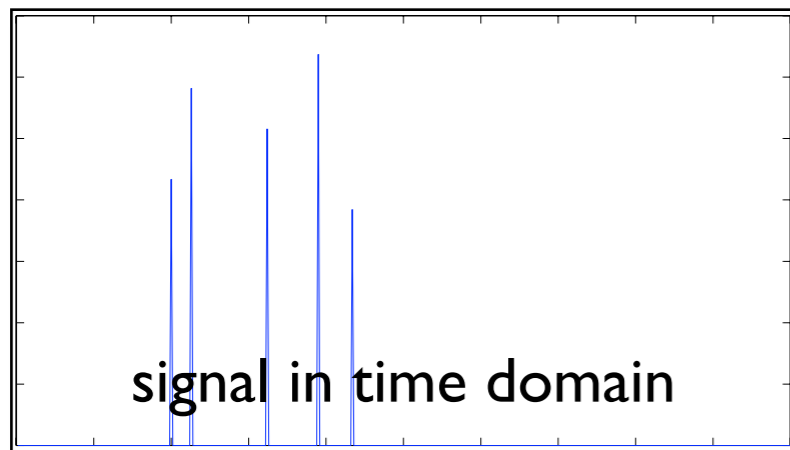
- can exactly "recover"  $\mathbf{x}$  from  $\mathbf{y}$  by solving L1 problem

$$\tilde{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1 = \sum_{i=1}^N |x_i| \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{y},$$









# Compressed Sensing

---

- $x$  has to be sparse
- $\mathbf{A}$  has to be Fourier transform
- Compressed sensing theory gives us strict bounds on regions of recoverability
- Enables deliberate *incomplete computations*

# Compressed Sensing “Computation”

- if we “shift”  $s(k)$  with  $e^{-j \frac{\Delta x}{2\pi} \Lambda}$ , what happens when we recover  $s(x)$  using  $s'(k)$ ?

$$\begin{array}{c} \boxed{\phantom{s'(k)}} \\ s'(k) \end{array} = \begin{array}{c} \boxed{\text{diagonal line}} \\ e^{-j \frac{\Delta x}{2\pi} \Lambda} \end{array} \begin{array}{c} \boxed{\phantom{s(k)}} \\ s(k) \end{array}$$

$$\begin{array}{c} \boxed{\phantom{s(k)}} \\ s(k) \end{array} = \begin{array}{c} \boxed{\mathbf{F}_{\omega=1,4,\dots}} \\ \end{array} \begin{array}{c} \boxed{\phantom{s(x)}} \\ s(x) \end{array}$$

# Compressed Sensing “Computation”

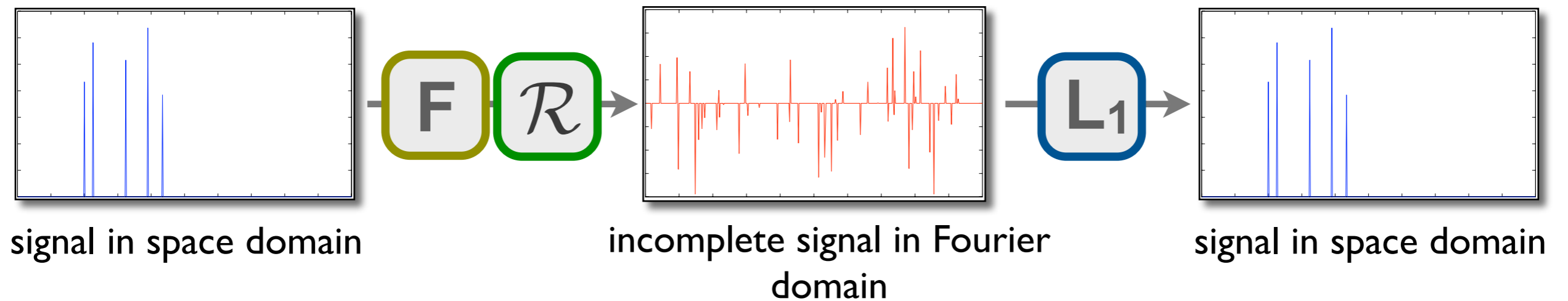
- if we “shift”  $s(k)$  with  $e^{-j\frac{\Delta x}{2\pi}\Lambda}$ , what happens when we recover  $s(x)$  using  $s'(k)$ ?

$$\begin{array}{c} \boxed{\phantom{s'(k)}} \\ s'(k) \end{array} = \begin{array}{c} \boxed{\text{diagonal line}} \\ e^{-j\frac{\Delta x}{2\pi}\Lambda} \end{array} \begin{array}{c} \boxed{\phantom{s(k)}} \\ s(k) \end{array}$$

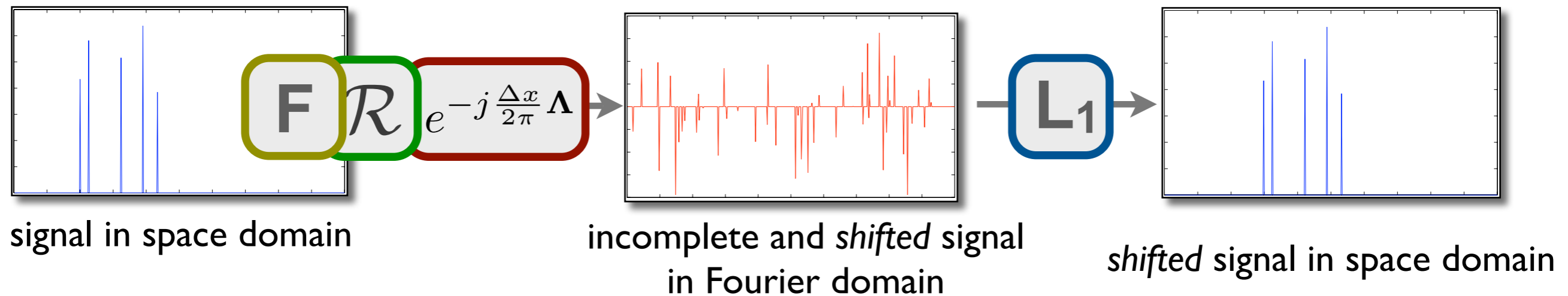
$$\begin{array}{c} \boxed{\phantom{s(k)}} \\ s(k) \end{array} = \begin{array}{c} \boxed{\mathbf{F}_{\omega=1,4,\dots}} \\ \end{array} \begin{array}{c} \boxed{\phantom{s(x)}} \\ s(x) \end{array}$$

- Answer: we recover a shifted  $s(x)$ !

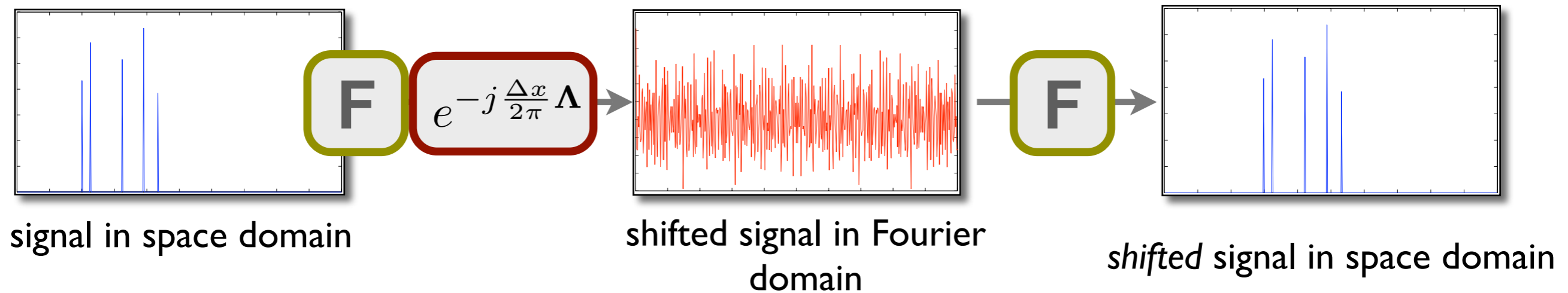
# Compressed Sensing



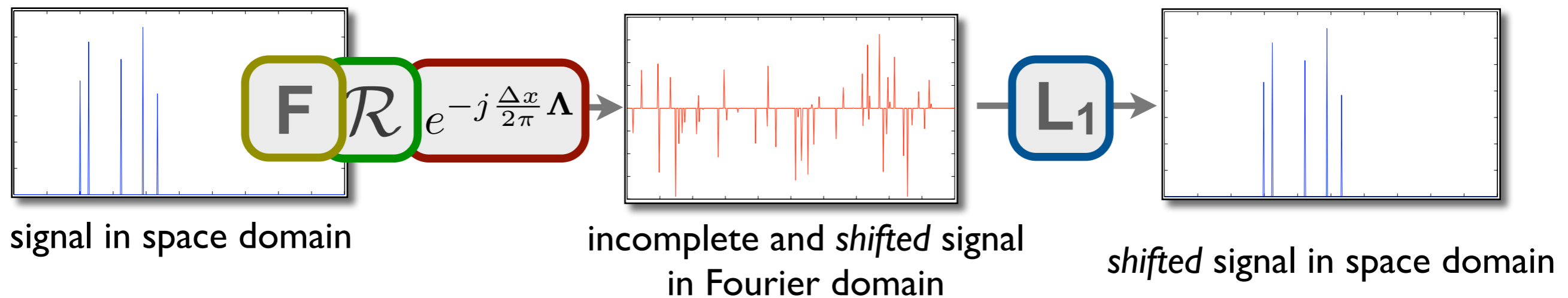
# Compressed Processing



# Straightforward Computation



# Compressed Processing



# Compressed Sensing “Computation”

---

- In a nutshell:
  - Trades the cost of L1 solvers for a compressed operator that is cheaper to compute, store, and synthesize
- L1 solver research is currently a hot topic in applied mathematics

Tibshirani, R., 1996, Least absolute shrinkage and selection operator, Software: <http://www-stat.stanford.edu/~tibs/lasso.html>.

Candès, E. J., and J. Romberg, 2005,  $\ell_1$ -magic. Software: <http://www.acm.caltech.edu/limagic/>.

Donoho, D. L., I. Drori, V. Stodden, and Y. Tsaig, 2005, SparseLab, Software: <http://sparselab.stanford.edu/>.

Figueiredo, M., R. D. Nowak, and S. J. Wright, 2007, Gradient projection for sparse reconstruction, Software: <http://www.lx.it.pt/~mtf/GPSR/>.

Koh, K., S. J. Kim, and S. Boyd, 2007, Simple matlab solver for  $\ell_1$ -regularized least squares problems, Software: <http://www-stat.stanford.edu/~tibs/lasso.html>.

# Compressed Wavefield Extrapolation

- Recall the similarity between  $\mathbf{W}^\pm$  and  $\mathbf{S}$

$$\mathbf{W}^\pm = \begin{bmatrix} | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \end{bmatrix} \begin{bmatrix} \diagdown \\ \diagdown \\ \diagdown \\ \diagdown \\ \diagdown \end{bmatrix} \begin{bmatrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{bmatrix}$$

$\mathbf{L} \quad e^{-j\sqrt{\Lambda}\Delta x_3} \quad \mathbf{L}^T$

$$\mathbf{H}_2 = \begin{bmatrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \\ | & | & | & | & | \end{bmatrix} \begin{bmatrix} \diagdown \\ \diagdown \\ \diagdown \\ \diagdown \\ \diagdown \end{bmatrix} \begin{bmatrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{bmatrix}$$

$\mathbf{F} \quad e^{-j\frac{\Delta x}{2\pi}\Lambda} \quad \mathbf{F}^T$

$$\mathbf{D} = \begin{bmatrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{bmatrix}$$

# Compressed Wavefield Extrapolation

□ Structure of  $\mathbf{H}_1$        $\mathbf{H}_2 = \mathbf{L}\mathbf{\Lambda}\mathbf{L}^T$        $\mathbf{H}_1 = \mathbf{L}\mathbf{\Lambda}^{1/2}\mathbf{L}^T$

- analytically

$$\mathcal{H}_2 = \mathcal{H}_1 \mathcal{H}_1$$

$$\mathcal{H}_2 = k^2(\mathbf{x}) + \partial_\mu \partial_\mu$$

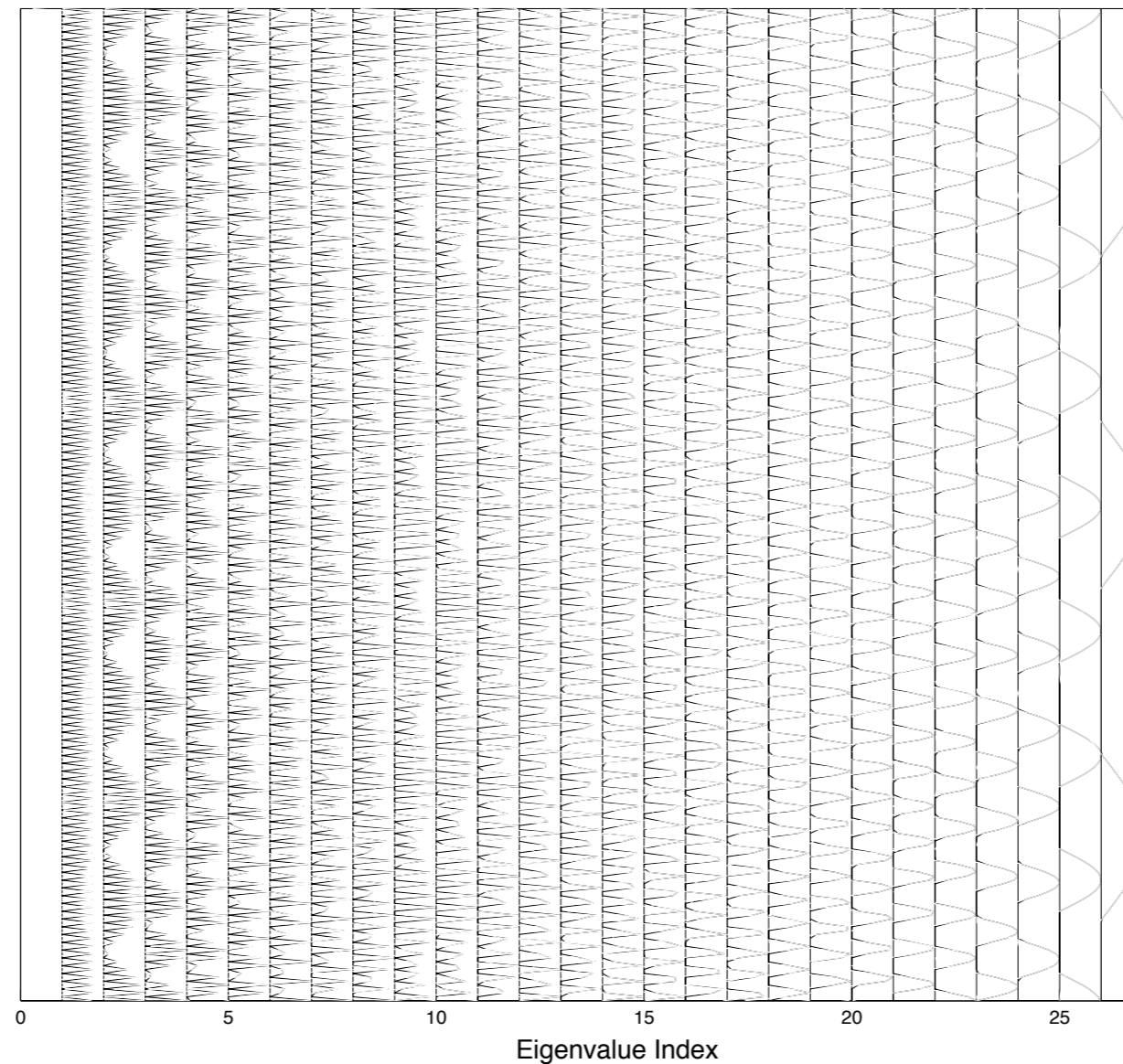
- discretely

$$\mathbf{H}_2 = \mathbf{C} + \mathbf{D}_2$$

$$\mathbf{H}_2 = \begin{bmatrix} \left(\frac{\omega}{c_1}\right)^2 & 0 & \dots & 0 \\ 0 & \left(\frac{\omega}{c_2}\right)^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \left(\frac{\omega}{c_{n_1}}\right)^2 \end{bmatrix} + \mathbf{D}_2$$

# Compressed Wavefield extrapolation

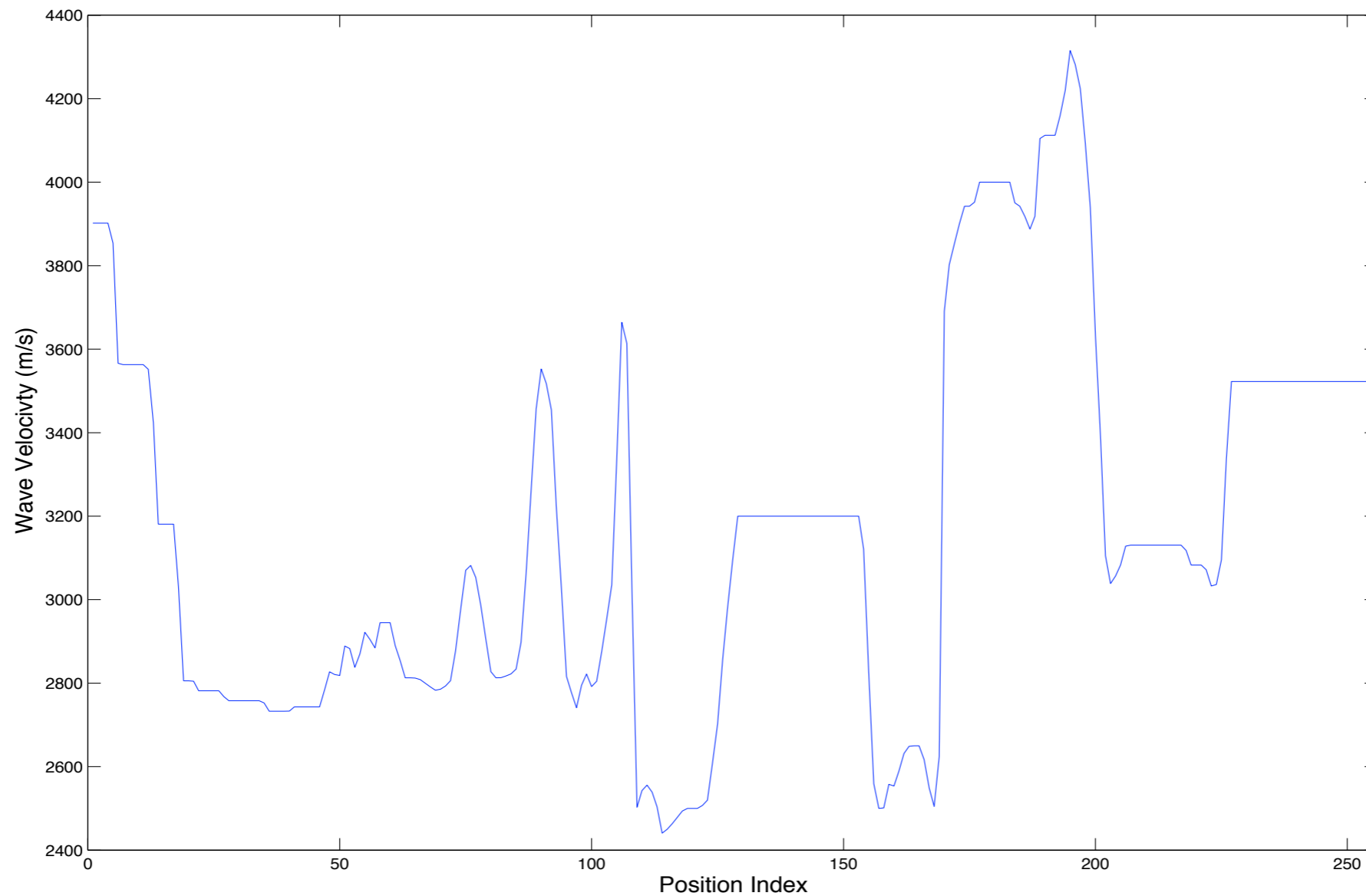
eigenfunctions of  $\mathbf{H}_2$  at 30 Hz for constant velocity medium



Asymptotically identical to the Cosine transform

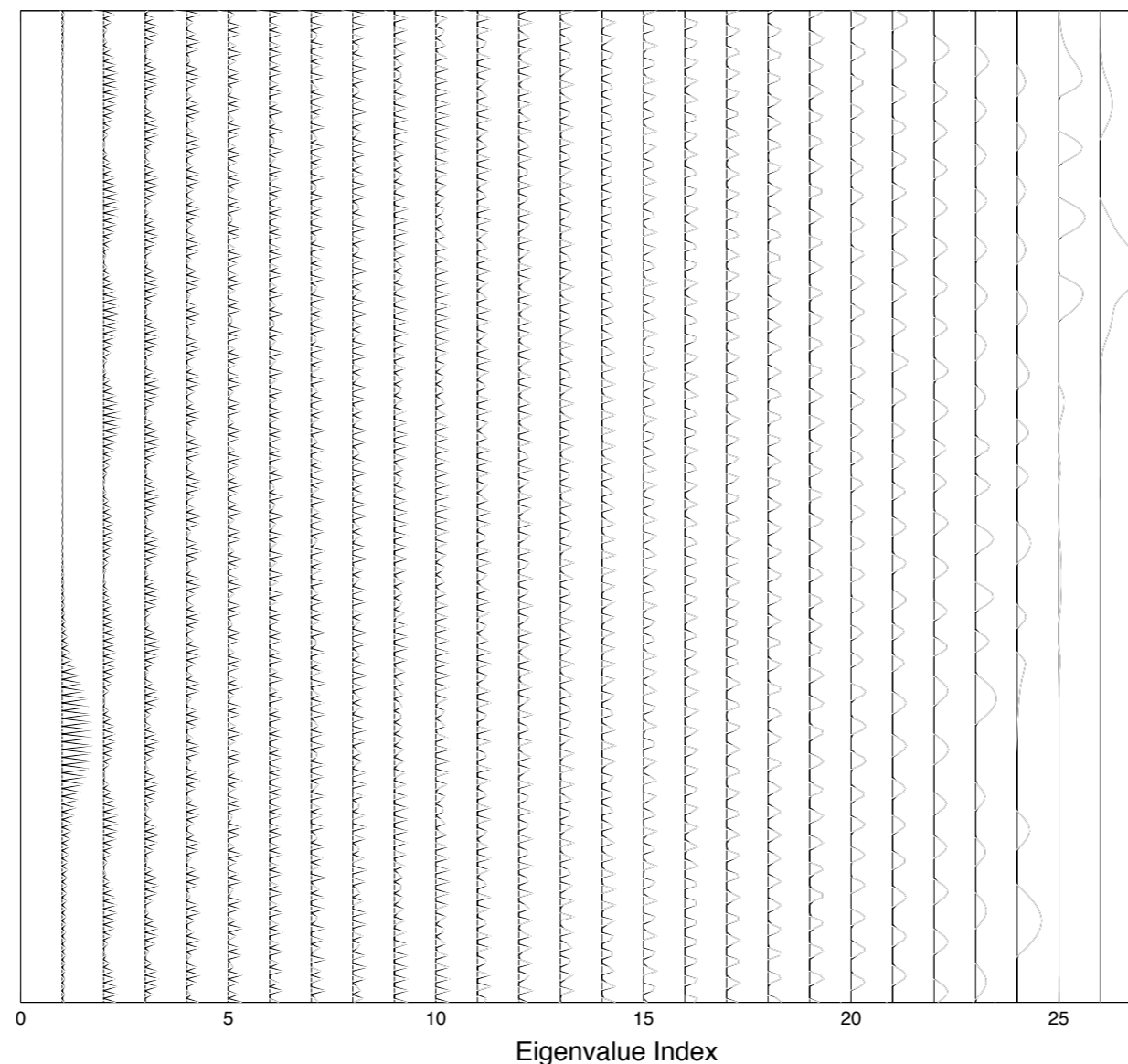
# Compressed Wavefield extrapolation

eigenfunctions of  $\mathbf{H}_2$  at 30 Hz for Marmousi velocity medium



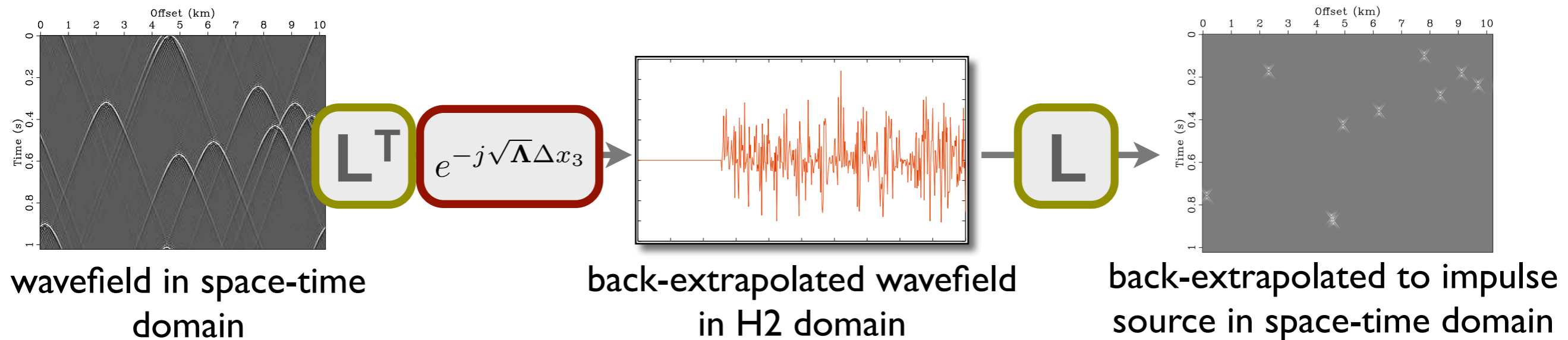
# Compressed Wavefield extrapolation

eigenfunctions of  $\mathbf{H}_2$  at 30 Hz for Marmousi velocity medium

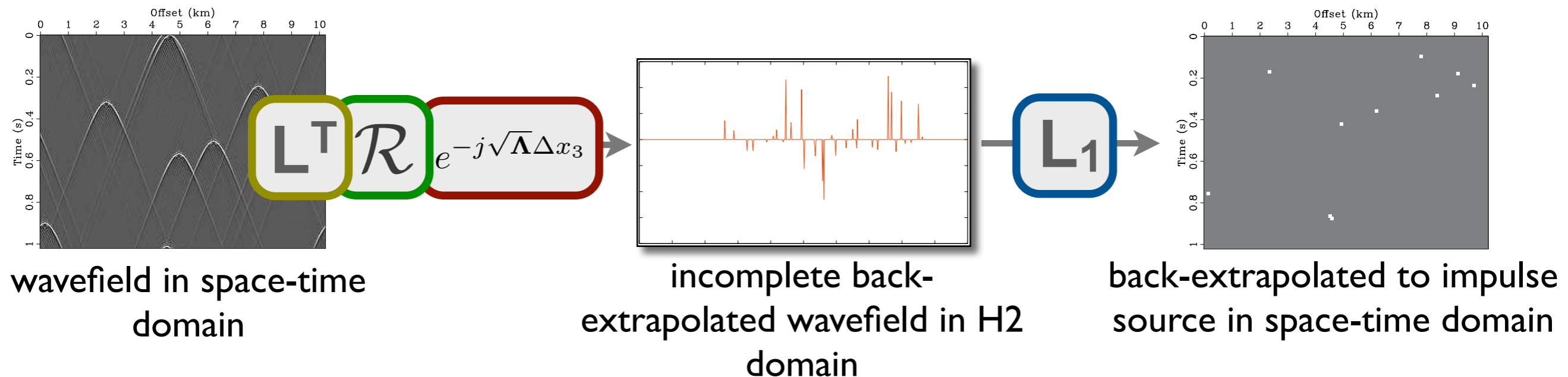


fairly close to the Cosine transform

# Straightforward 1-Way inverse Wavefield Extrapolation

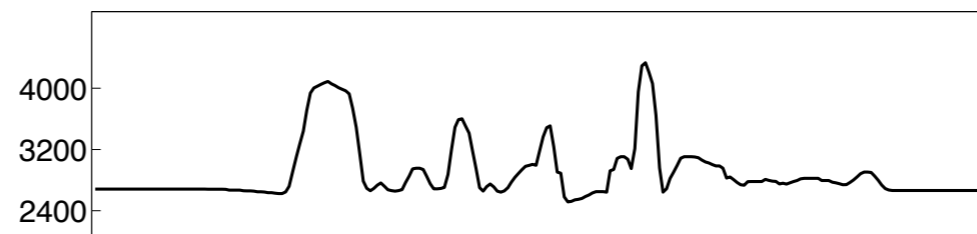
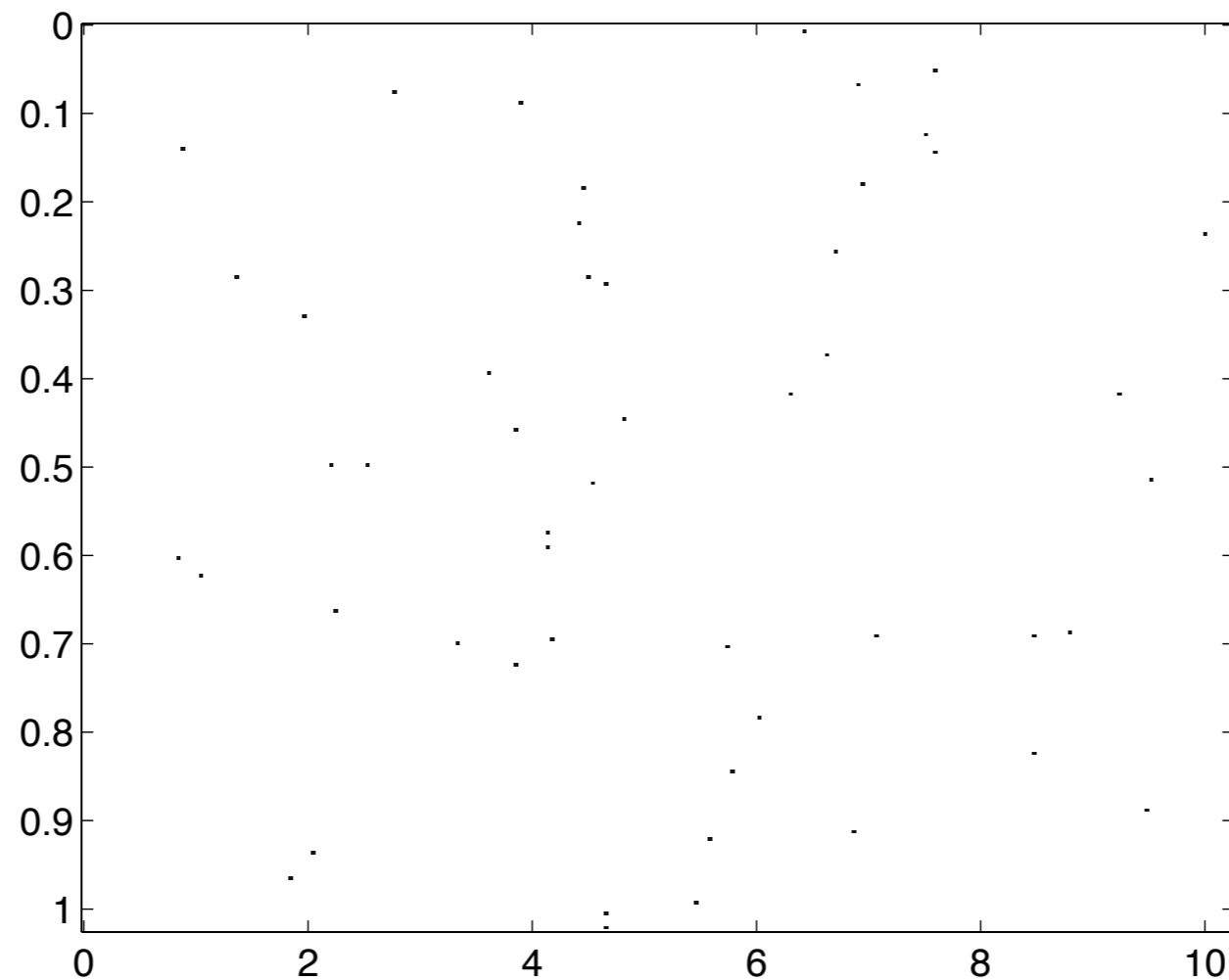


# Compressed 1-Way Wavefield Extrapolation



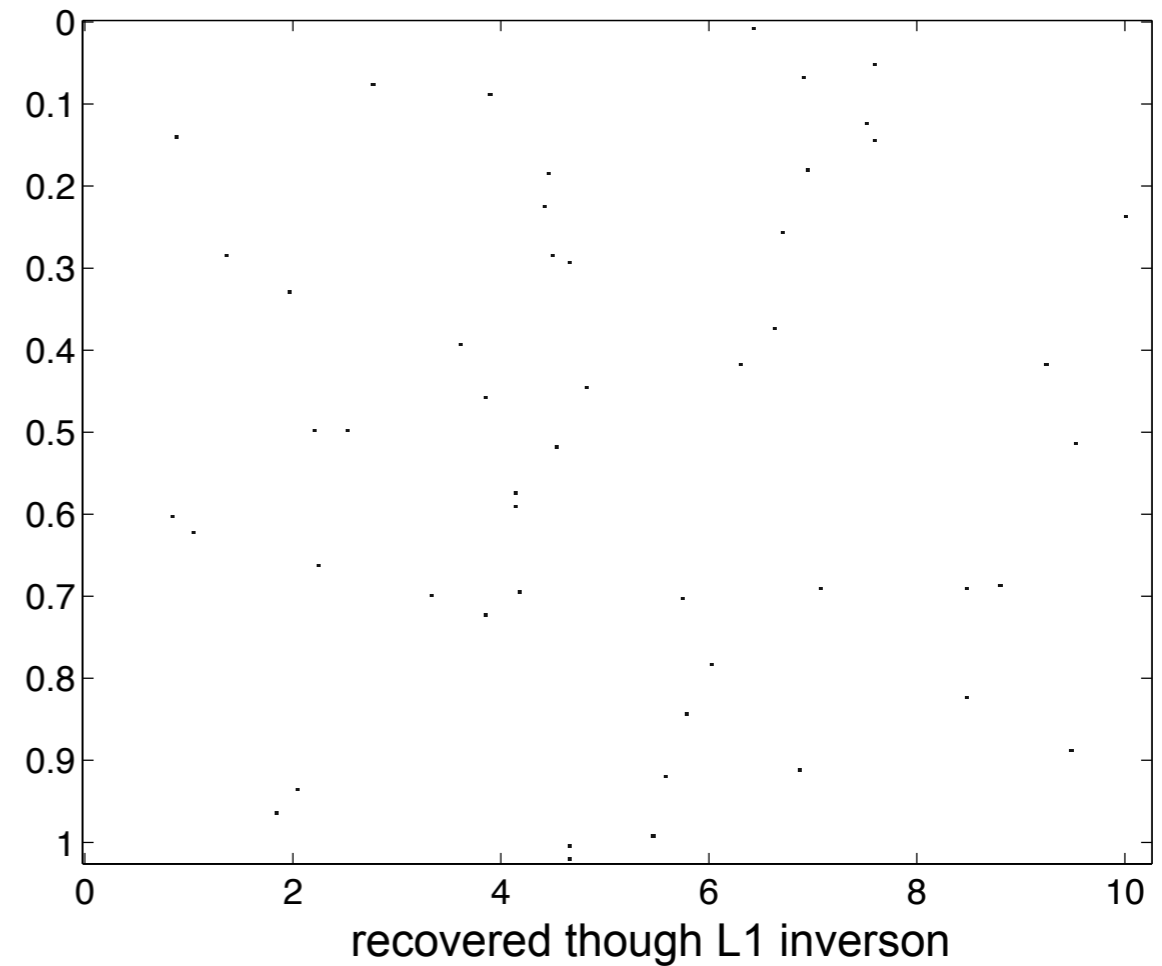
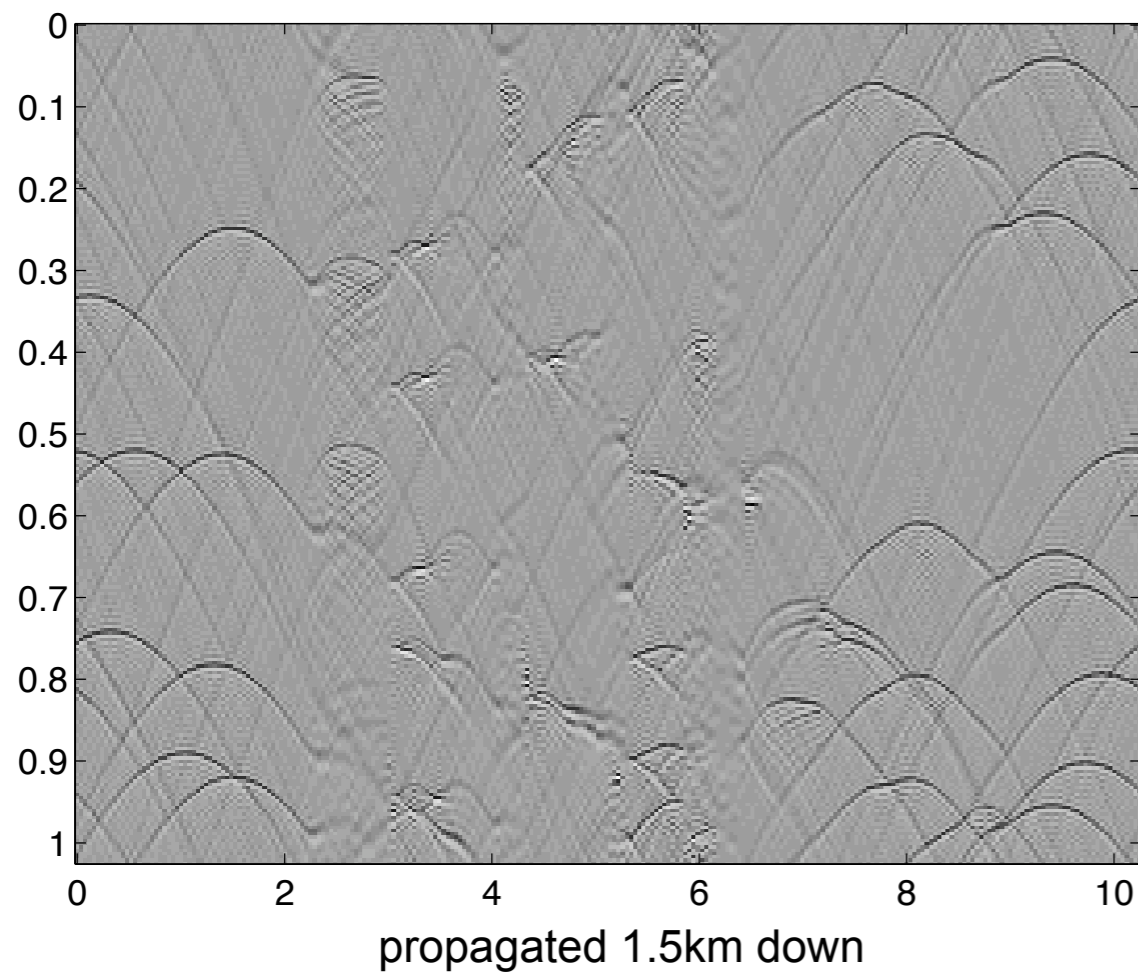
# Compressed wavefield extrapolation

simple 1-D space/time propagation example with point scatters



# Compressed wavefield extrapolation

simple 1-D space/time propagation example with point scatters



Restricted L transform to  $\sim 0.01$  of original coefficients

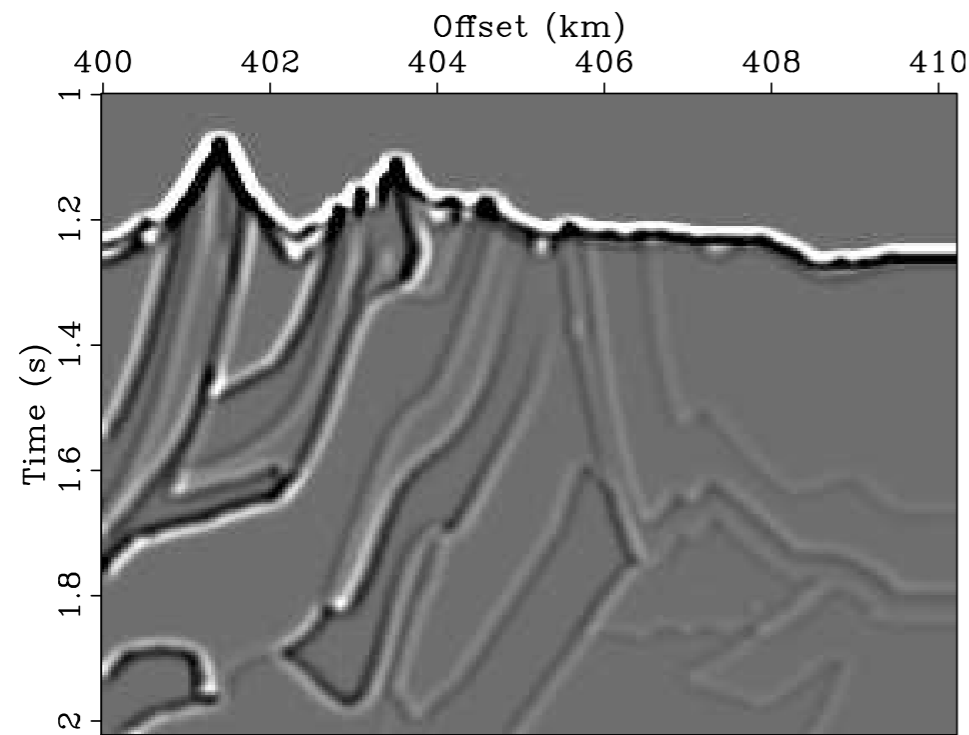
# Sparsity through curvelets

---

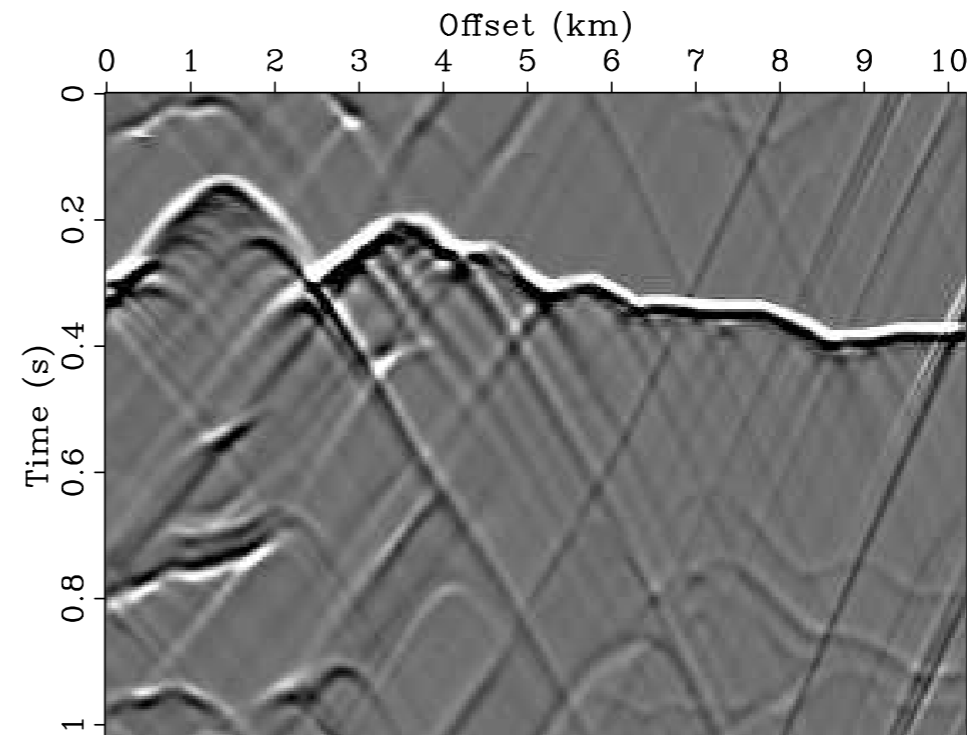
- for extrapolation to reflectivity, we first transform signal into a sparsifies reflectivity
- we know reflectivity are sparse in curvelets

Candès, E. J., and L. Demanet, 2005, The curvelet representation of wave propagators is optimally sparse: Communications on Pure and Applied Mathematics, **58**, 1472–1528.

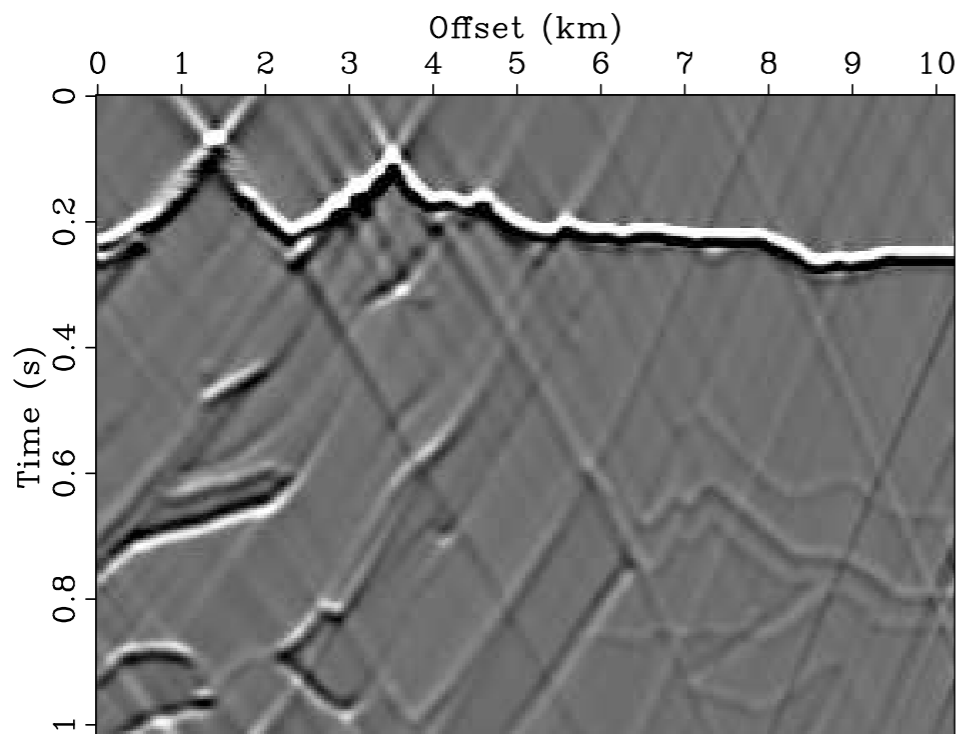
# Example (Canadian overthrust)



original reflectivity



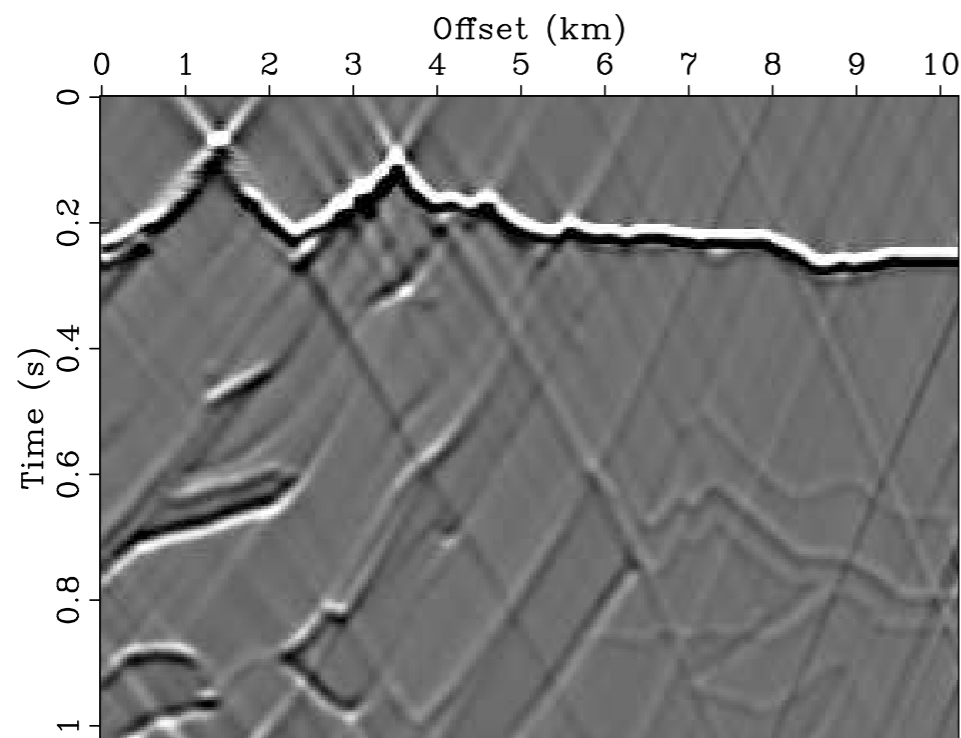
downward extrapolated 50m



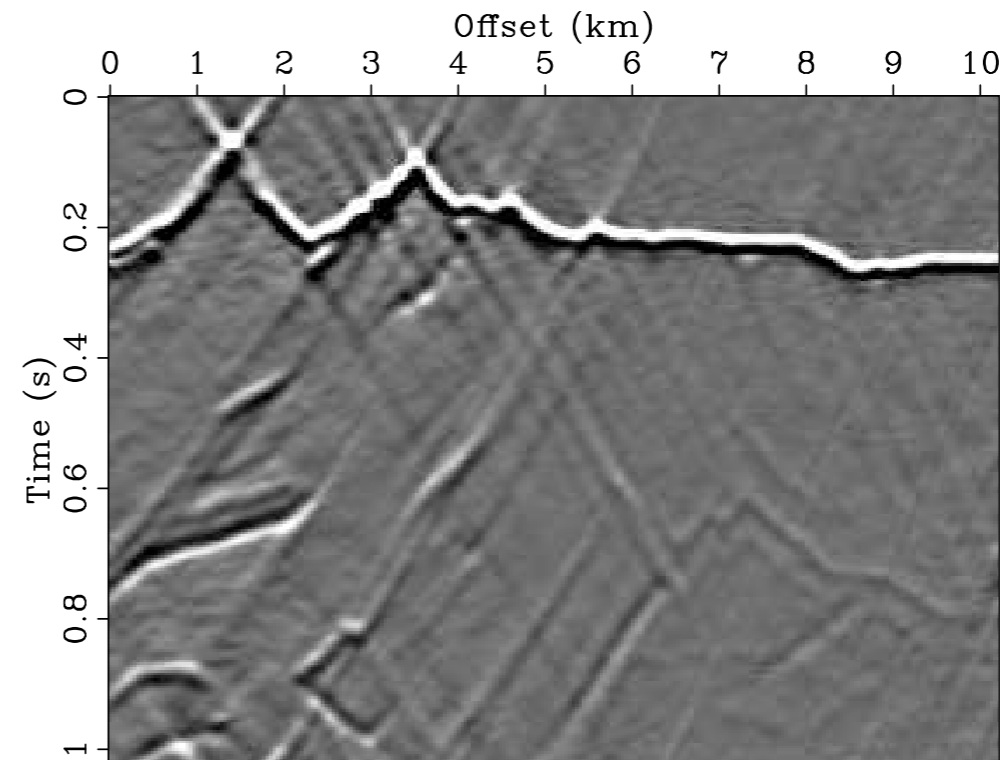
inverse extrapolated explicitly

# Example (Canadian overthrust)

---



inverse extrapolated explicitly



inverse extrapolated with  
compressed computation

~15% coefficients used

# Discussions

---

- Bottom line: synthesis, operation, and storage cost savings *versus* L1-solver cost
- require good sparsity-promoting basis (ie Curvelets)
- potential to apply same technique to a variety of different operators

# Conclusions

---

- 1) Take linear operator with suitable structure for compressed sensing, having a diagonalizing basis which is incoherent with the signal basis
- 2) Compressed sensing theory tells us how much computation we can throw away while still recovering full signal with L1 solver
- 3) Then we can take advantage of results in compressed sampling for compressed computation
  
- Take home point:
  - *Exploit compressed sensing theory for gains in scientific computation*

# Still awake?

---

- Check-out the full paper at:

Lin, T.T.Y. and F. Herrmann, 2007, Compressed wavefield extrapolation: Geophysics, 72, SM77-SM93

# Compressed wavefield extrapolation

---

$$\begin{cases} \mathbf{y} &= \mathbf{R} e^{-j\omega \sqrt{\Lambda} \Delta x_3} \mathbf{L}^T \mathbf{u} \\ \tilde{\mathbf{x}} &= \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{R} \mathbf{L}^T \mathbf{x} = \mathbf{y} \\ \tilde{\mathbf{u}}' &= \tilde{\mathbf{x}} \end{cases}$$

- Randomly subsample in the Modal domain
- Recover by norm-one minimization
- Capitalize on
  - the incoherence between modal functions and impulse sources
  - reduced explicit matrix size

# Compressed wavefield extrapolation with curvelets

---

$$\begin{cases} \mathbf{y} &= \mathbf{R}e^{-j\omega\sqrt{\Lambda}\Delta x_3}\mathbf{L}^T\mathbf{C}^T\mathbf{u} \\ \tilde{\mathbf{x}} &= \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{R}\mathbf{L}^T\mathbf{C}^T\mathbf{x} = \mathbf{y} \\ \tilde{\mathbf{u}}' &= \tilde{\mathbf{x}} \end{cases}$$

- Original and reconstructed signals remain in the curvelet domain
- Original curvelet transform must be done outside of the algorithm