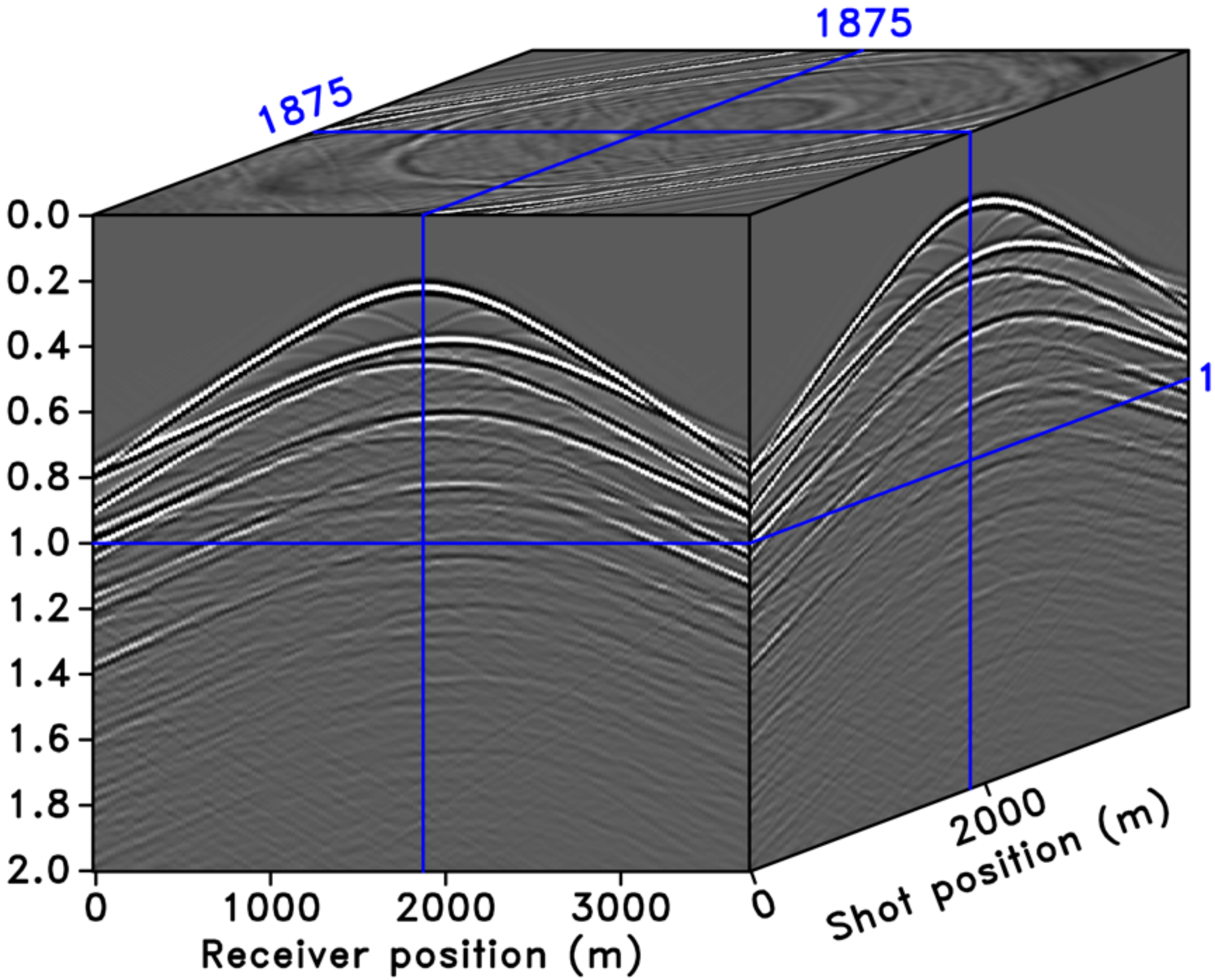
Dimensionality-reduced estimation of *primaries* by *sparse* inversion

Bander Jumah & Felix J. Herrmann

SLIM
University of British Columbia



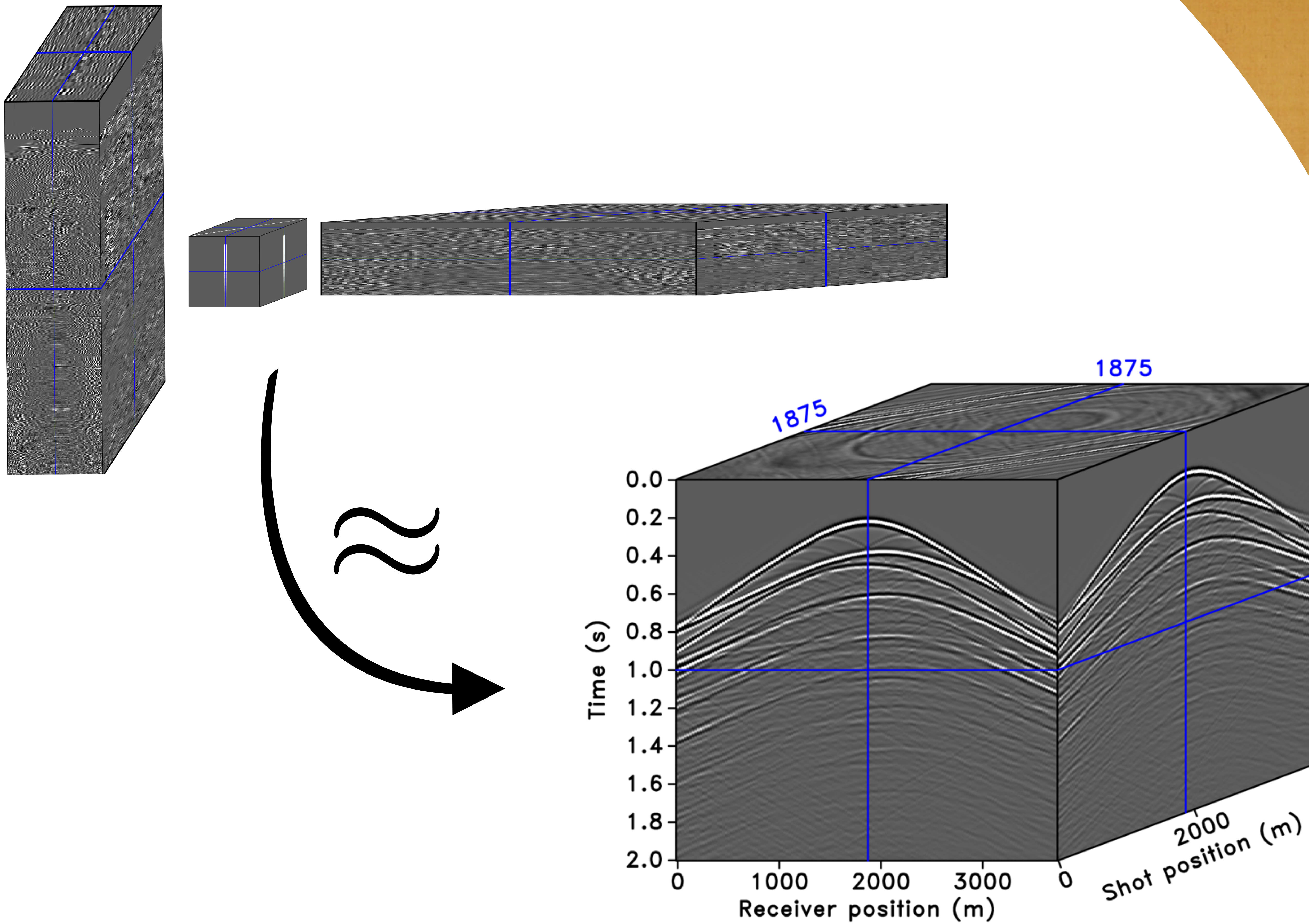
Seismic Data

2D data

- 3D - representations
- dense operations

3D data

- 5D - representations
- giga-bytes to tera-bytes


≈
1875
1875
1
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
Time (s)
0
1000
2000
3000
Receiver position (m)
0
2000
Shot position (m)

SLIM

Motivation

Data-driven methods

- Estimation of Primaries by Sparse Inversion (EPSI)
- Surface-Related Multiple Elimination (SRME)

Curse of dimensionality

- In 3D these methods *suffer* from *exponential* growth in computational & storage *demands*

[Herrmann, F. J., 2010, Randomized sampling and sparsity]

Objective

Reduction in *computational* and *storage* demands:

- *dimensionality*-reduction technique
- *adaptive* low-rank approximation
- *black-box* implementation

EPSI problem

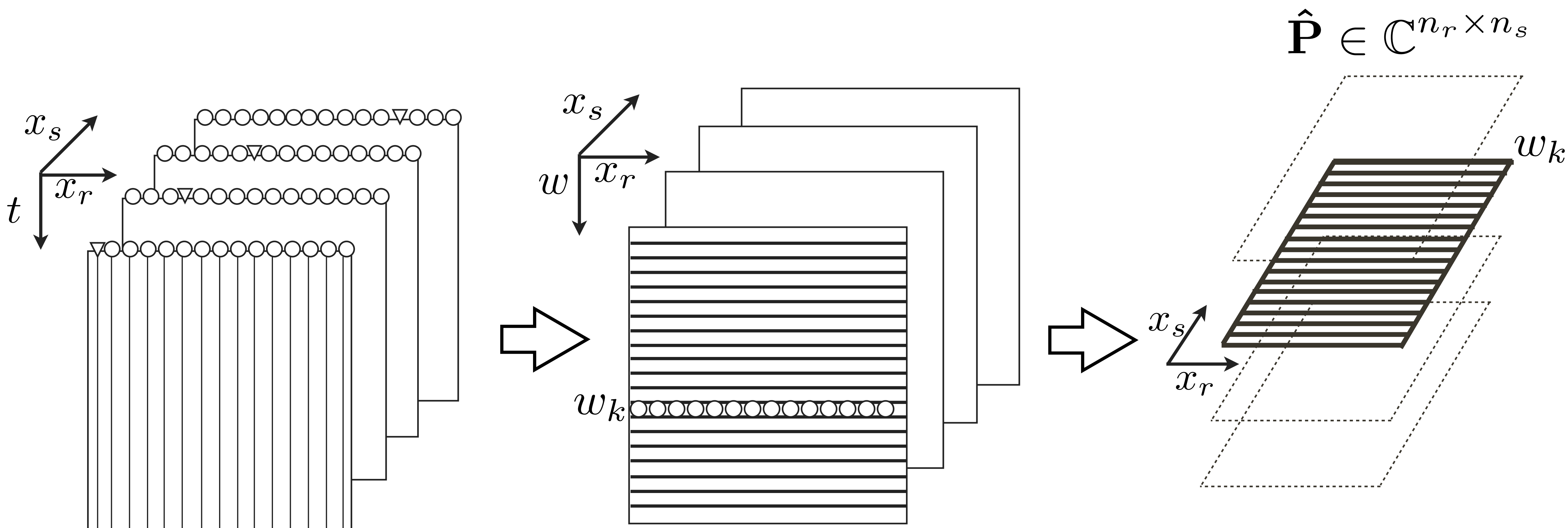
recorded data predicted data

$$\hat{\mathbf{P}} = \hat{\mathbf{G}}(\hat{\mathbf{Q}} + \hat{\mathbf{R}}\hat{\mathbf{P}})$$

$\hat{\mathbf{P}}$	total up-going wave-field
$\hat{\mathbf{Q}}$	down-going source signature
$\hat{\mathbf{R}}$	reflectivity of free surface
$\hat{\mathbf{G}}$	surface-free Green's function

[van Groenestijn and Verschuur, 2009]

Monochromatic "data matrices"



Fourier transform

Extract monochromatic shot records

[Ewoud van Dedem, 2002]

SLIM

EPSI problem

recorded data

predicted data

$$\hat{\mathbf{P}} = \hat{\mathbf{G}}(\hat{\mathbf{Q}} - \hat{\mathbf{P}})$$

$\hat{\mathbf{P}}$ matrix (known)

$\hat{\mathbf{Q}}$ full-rank diagonal matrix (known)

$\hat{\mathbf{R}}$ assume $-\mathbf{I}$

$\hat{\mathbf{G}}$ unknown

EPSI problem

$$\mathbf{Ax} \approx \mathbf{b}$$

$$\tilde{\mathbf{x}} = \underbrace{\arg\min_{\mathbf{x}} ||\mathbf{x}||_{\mathbf{1}}}_{\text{sparsity promoting part}} \quad \text{subject to} \quad \underbrace{||\mathbf{Ax} - \mathbf{b}||_{\mathbf{2}} \leq \sigma}_{\text{data fitting part}}$$

σ : residual between the recorded & predicted data

EPSI problem

EPSI *linear* algebra format:

$$\underbrace{\mathbf{F}_t^* \begin{bmatrix} \left((\widehat{\mathbf{Q}} - \widehat{\mathbf{P}})_1^* \otimes \mathbf{I}\right) & & \\ & \ddots & \\ & & \left((\widehat{\mathbf{Q}} - \widehat{\mathbf{P}})_{n_f}^* \otimes \mathbf{I}\right) \end{bmatrix} \mathbf{F}_t}_{\mathbf{U}} \begin{bmatrix} \mathrm{vec}\,(\mathbf{G}_1) \\ \vdots \\ \mathrm{vec}\,(\mathbf{G}_{n_t}) \end{bmatrix} \approx \begin{bmatrix} \mathrm{vec}\,(\mathbf{P}_1) \\ \vdots \\ \mathrm{vec}\,(\mathbf{P}_{n_t}) \end{bmatrix}$$

$\mathbf{A} = \mathbf{U}\mathbf{C}^*$ $\mathbf{C}$ is curvelet-wavelet transform

$\mathbf{x}$: discrete curvelet-wavelet representation of $\mathbf{G}$

$\mathbf{b}$: vectorized representation of $\mathbf{P}$

SLIM

EPSI problem

Data matrix $\hat{\mathbf{P}}$

- dense
- extremely large

  for each 3D frequency is $10^6 \times 10^6$ where $n_r = n_s = 1000$
- expensive to access & store
- high mat-mat multiplication cost $O(N^3)$

EPSI problem

Challenges in solving the optimization problem

- multiple iterations
- multiple evaluations of $\mathbf{A}$, $\mathbf{A}^*$ and $\mathbf{A}^*\mathbf{A}$

SLIM

Dimensionality-reduction via SVD

Approximate data matrix $\hat{\mathrm{P}}$ with *low-rank* factorization:

$$\hat{\mathbf{P}} = \hat{\mathbf{G}}(\hat{\mathbf{Q}} - \hat{\mathbf{P}})$$

$$\hat{\mathbf{P}} \approx \mathbf{USV}^*$$

$\mathbf{U_{n_r \times k}}$ left singular vectors

$\mathbf{S_{k \times k}}$ singular values

$\mathbf{V^*_{k \times n_s}}$ right singular vectors

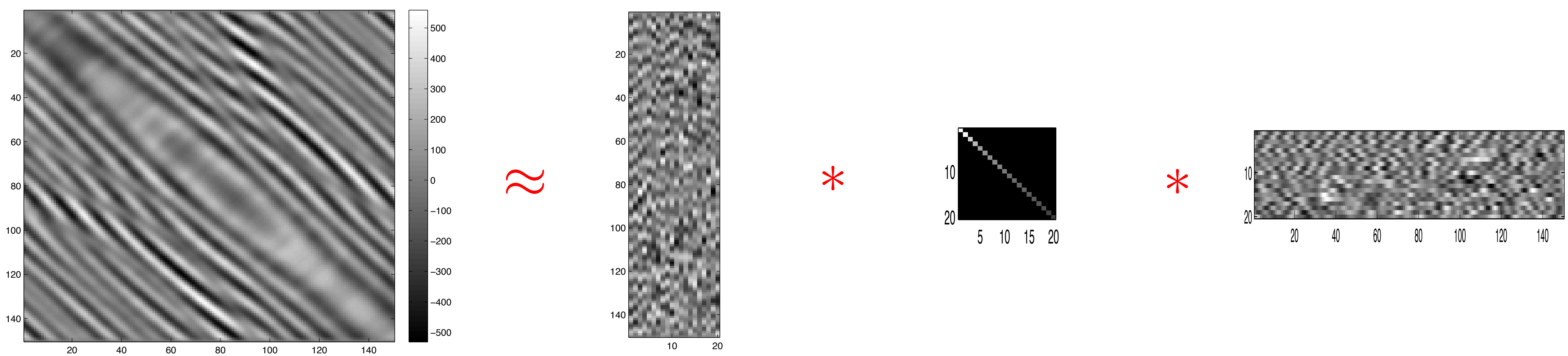
k : approximate rank

$k << min(n_r, n_s)$

Dimensionality-reduction via SVD

Approximate data matrix $\hat{\mathbf{P}}$ with *low-rank* factorization:

$\hat{\mathbf{P}}$ $\qquad$ $\mathbf{U}$ $\qquad$ $\mathbf{S}$ $\qquad$ $\mathbf{V}^*$



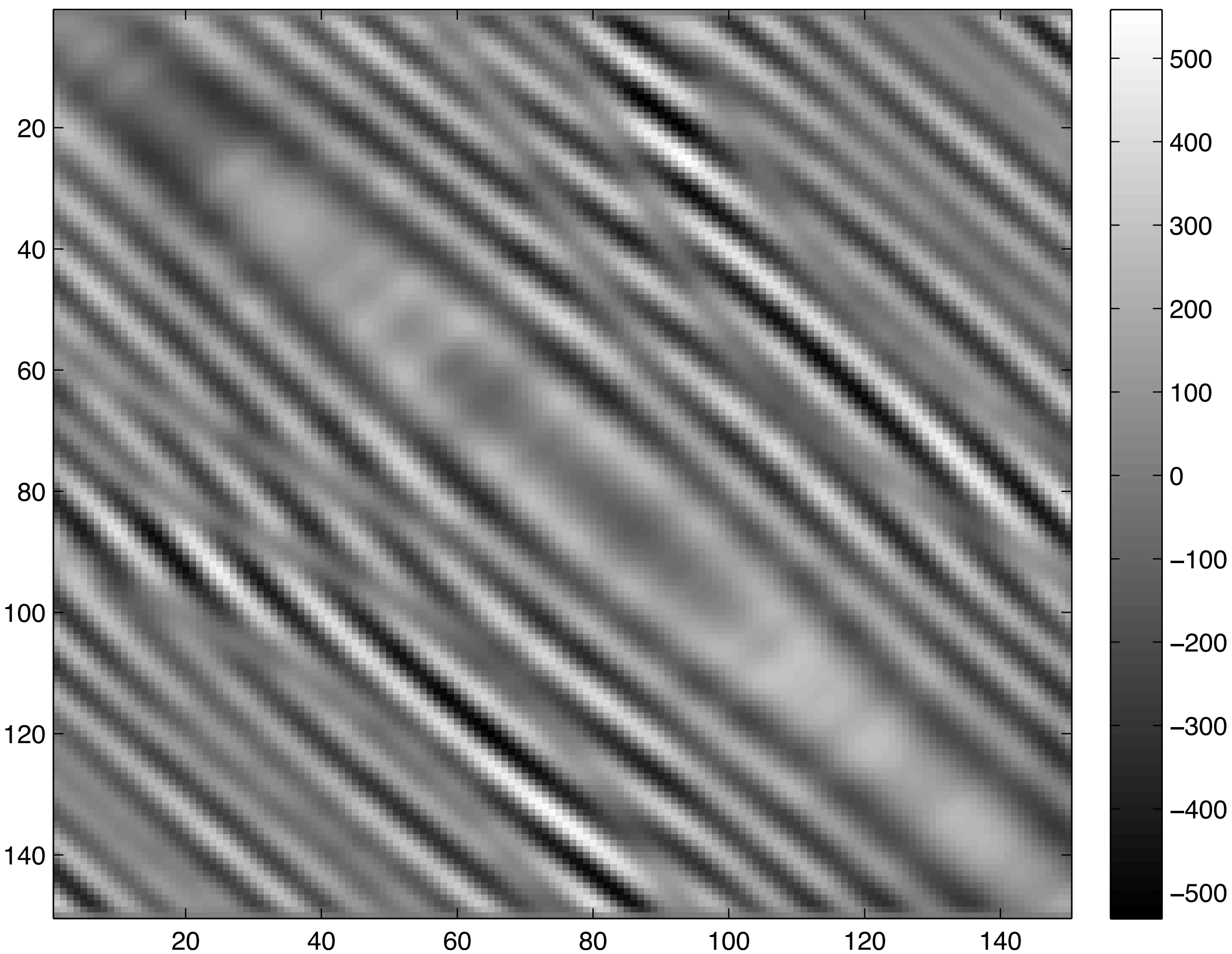
$n_r \times n_s$ $\qquad$ $n_r \times k$ $\qquad$ $k \times k$ $\qquad$ $k \times n_s$

k : approximate rank

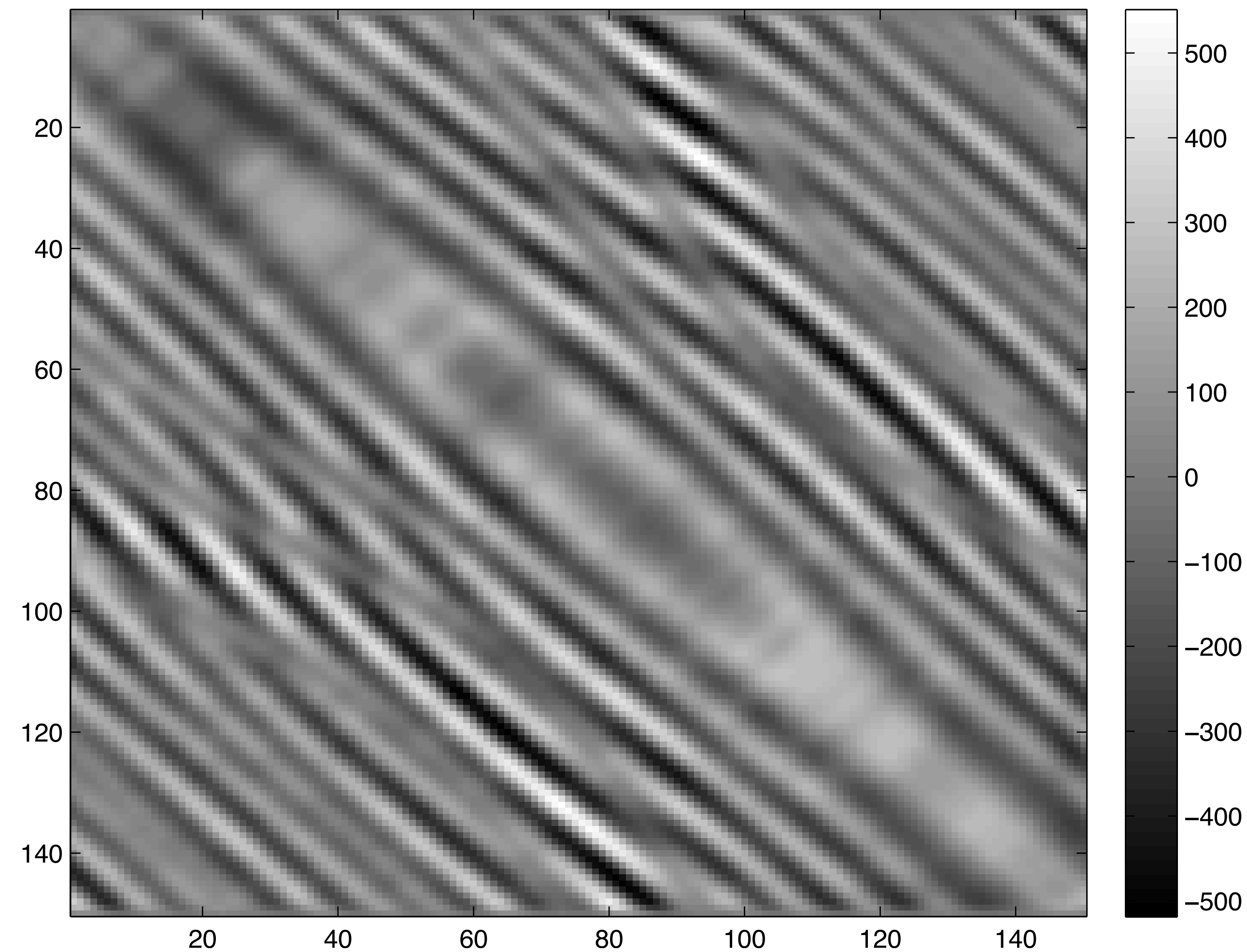
$k << min(n_r, n_s)$

Full vs approximated data

$\hat{\mathbf{P}}$



Approximated $\hat{\mathbf{P}}$



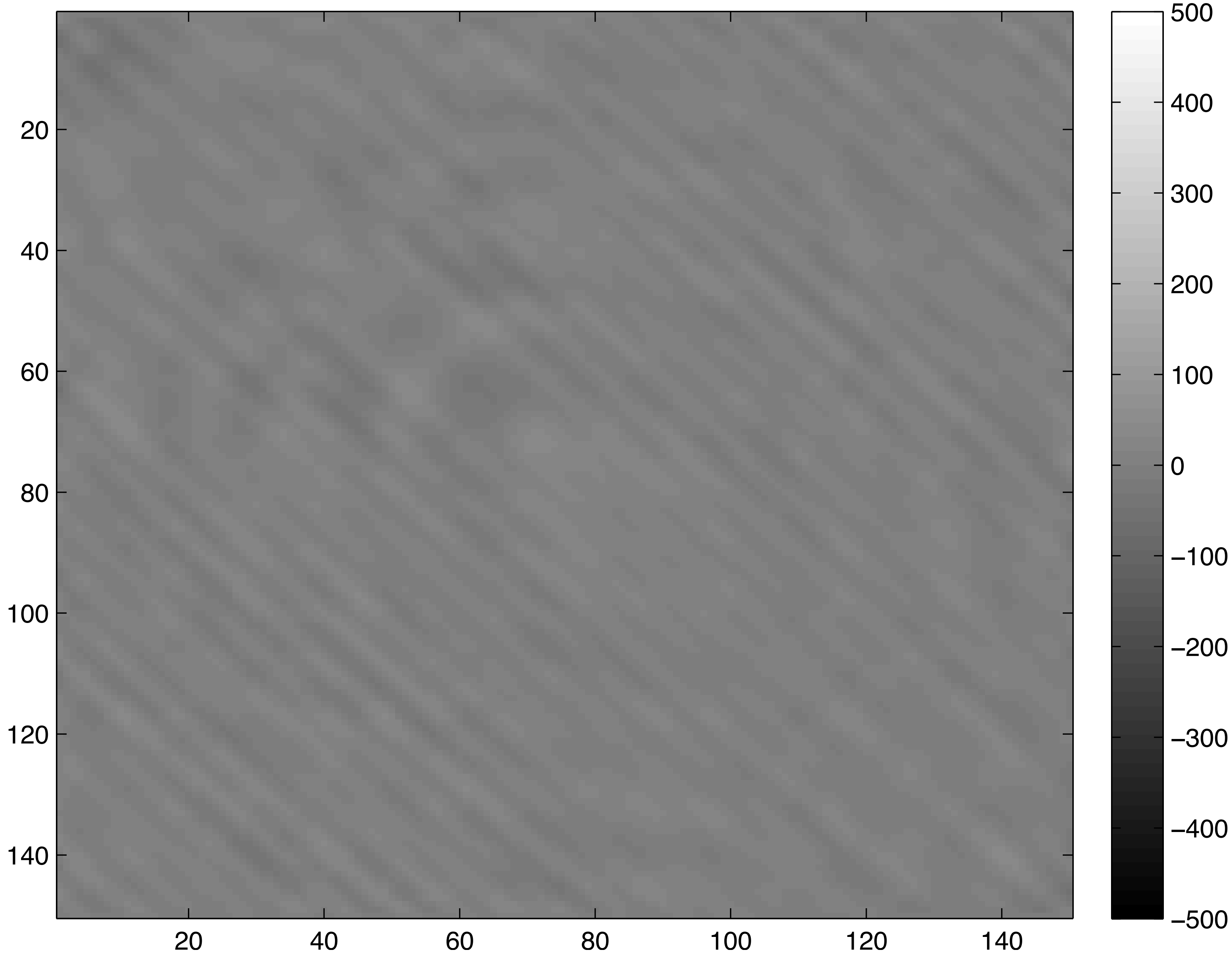
$n_s = n_r = 150$

$k = 20 = 14\%$

$SNR = 16dB$

Full vs approximated data

$$\widehat{\mathbf{P}} - \text{ approximated } \widehat{\mathbf{P}}$$



SNR = 16 dB

Multiplication speed up

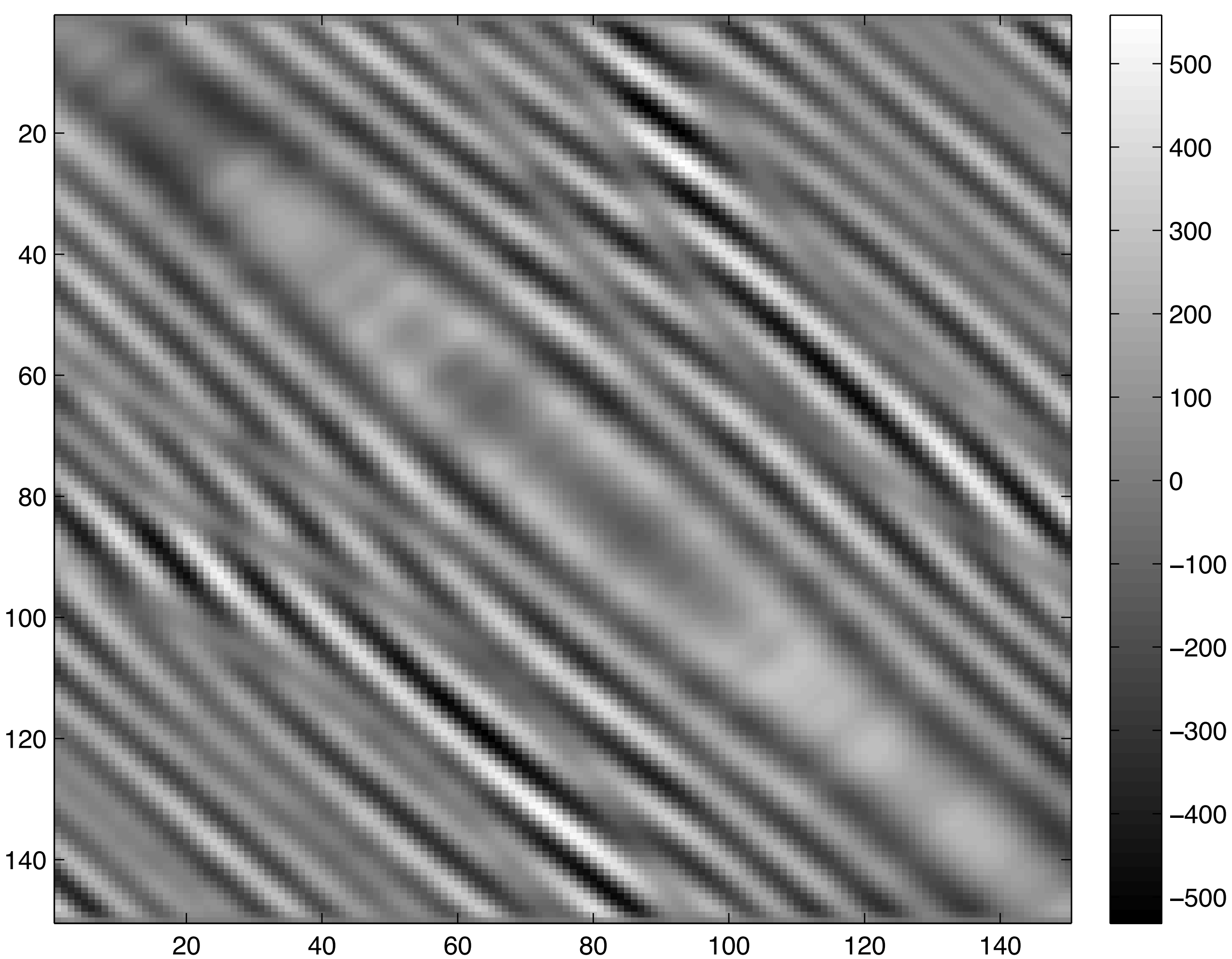
7.5x

Memory usage

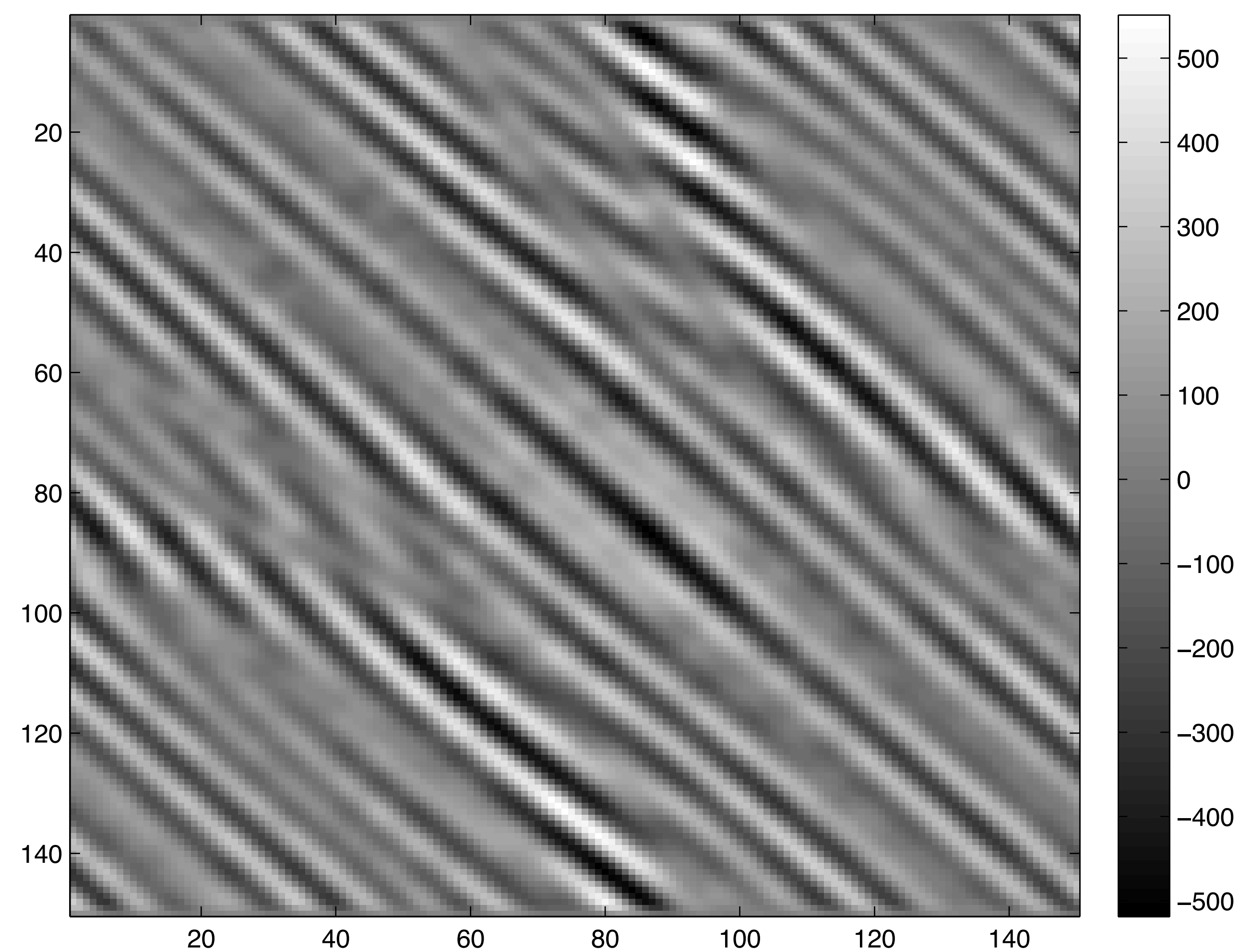
70% less

Full vs approximated data

$\hat{\mathbf{P}}$

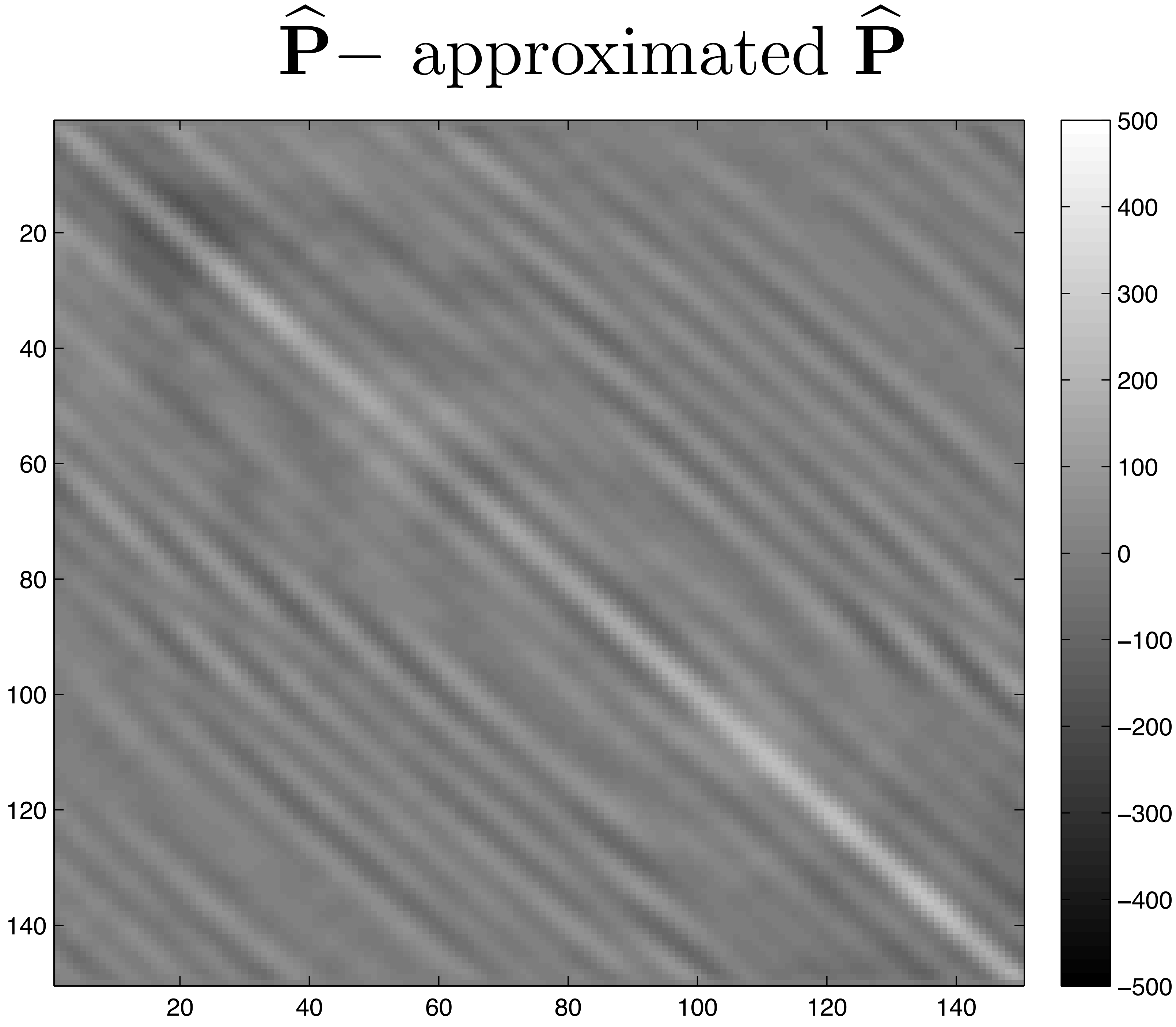


Approximated $\hat{\mathbf{P}}$



$$n_s = n_r = 150$$
$$k = 8 = 5\%$$
$$SNR = 8dB$$

Full vs approximated data



SNR = 8 dB

Multiplication speed up

18x

Memory usage

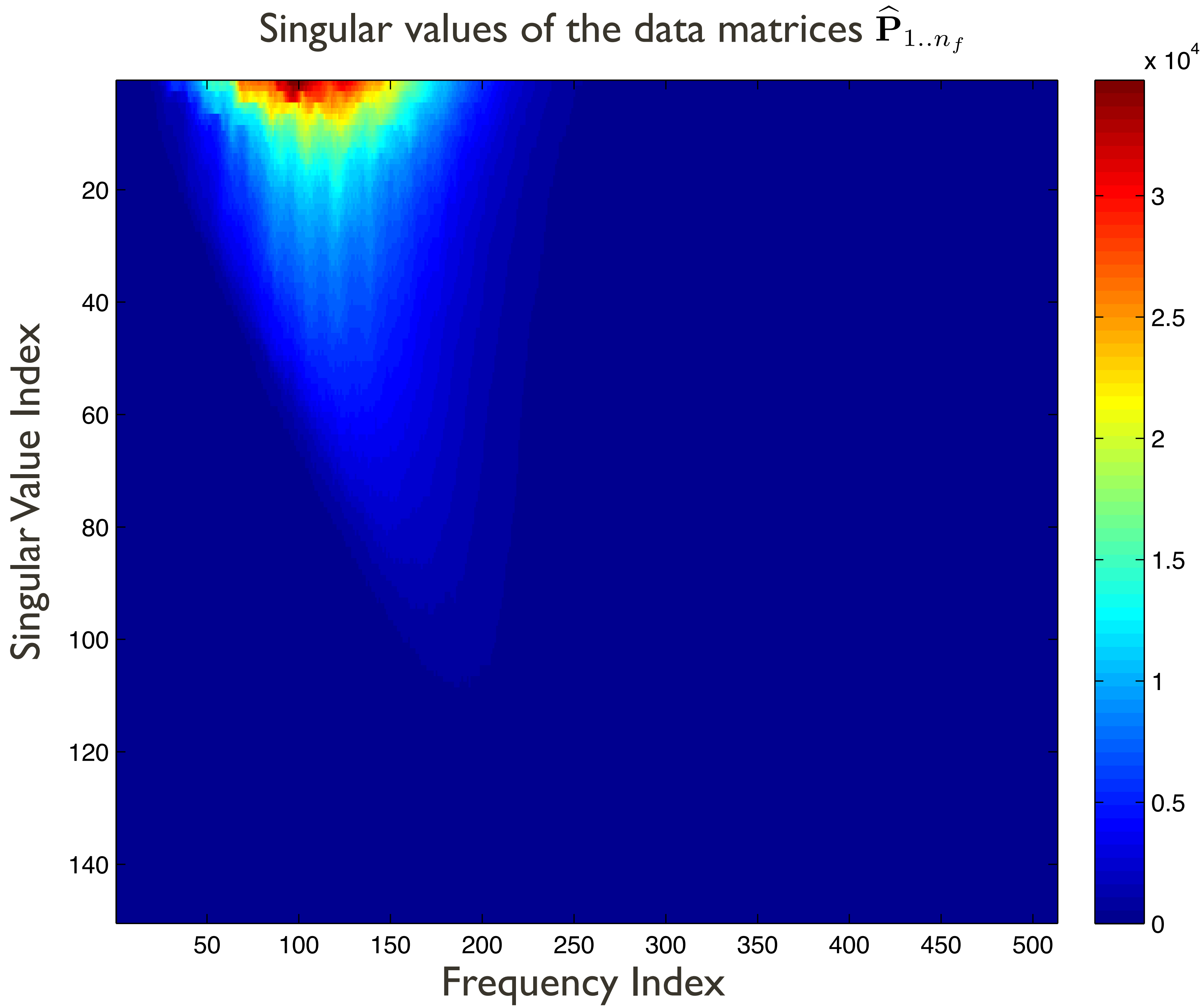
90% less

Dimensionality-reduction via SVD

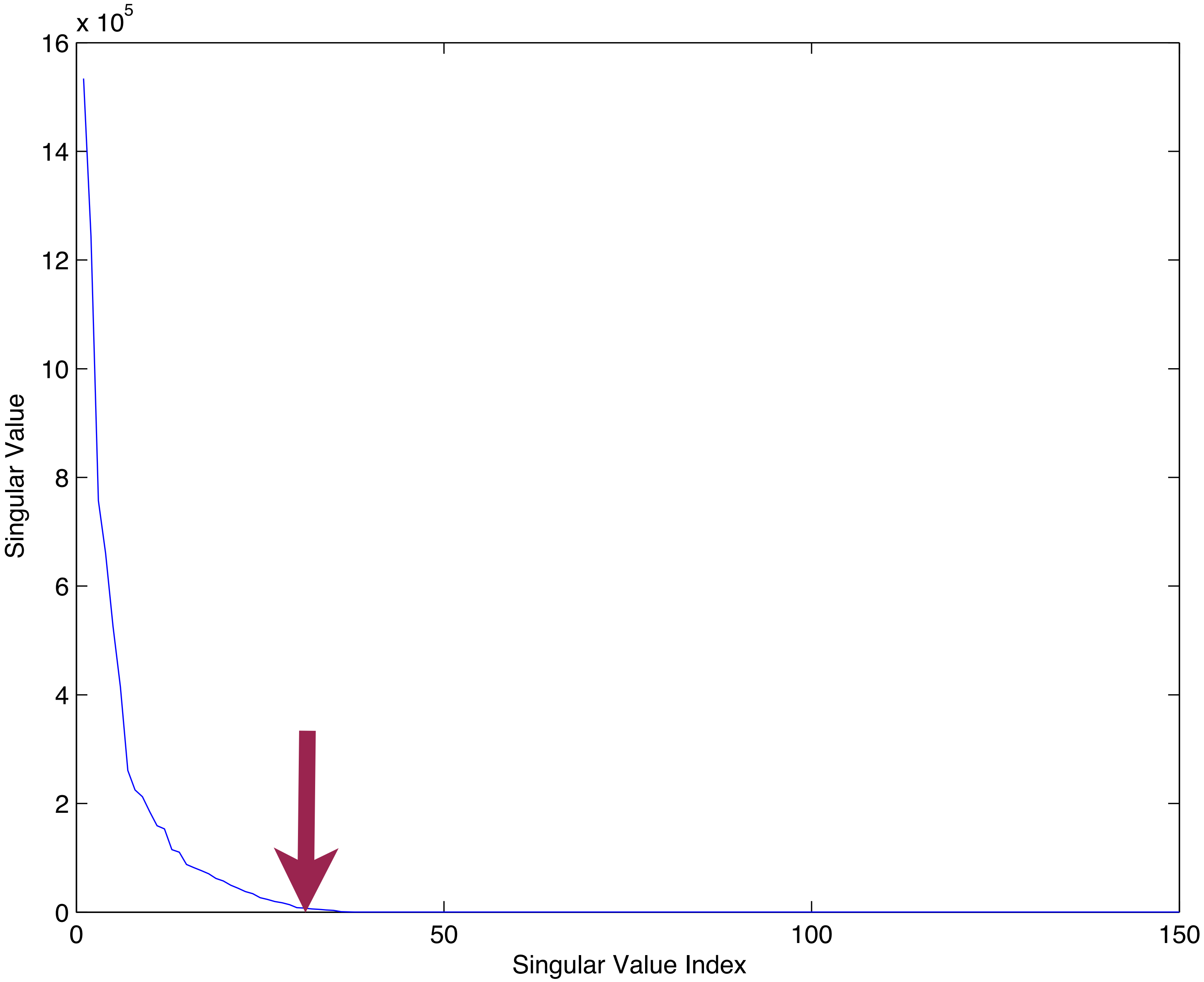
Advantages of using low-rank factorization

	Full data	**Low-rank approximation**
Matrix-Matrix multiplication	$O(N^3)$	$O(kN^2)$
Storage (bytes)	$O(N^2)$	$O(2Nk + k^2)$

Singular values of the data matrix

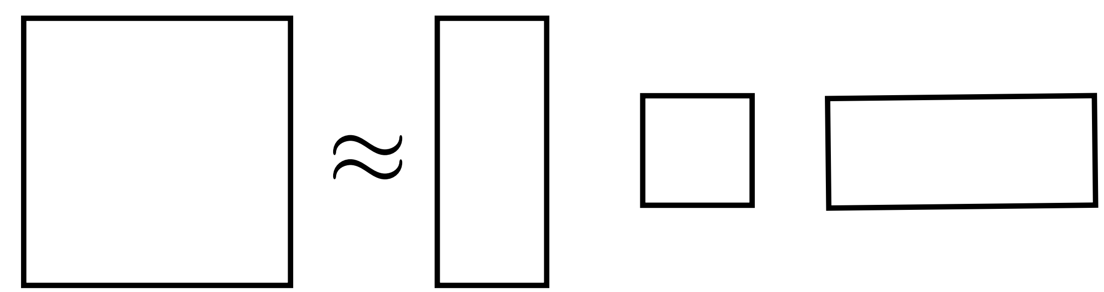


Decay of singular-values



full matrix size$= 150 \times 150$

rank $k = 30 = 20\%$

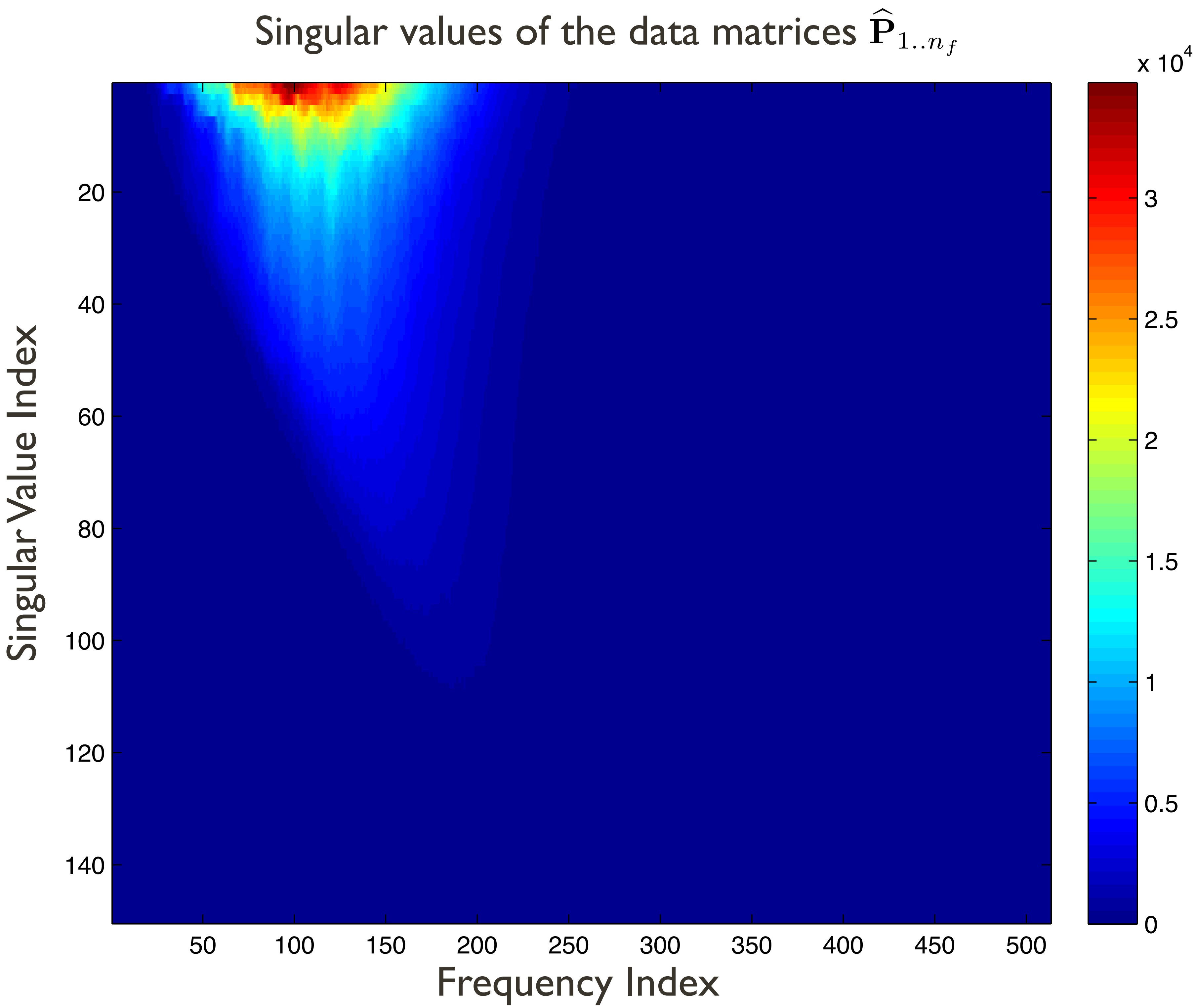


$$\hat{\mathbf{P}}_{150\times150} \approx \mathbf{U}_{150\times30} \; \mathbf{S}_{30\times30} \; \mathbf{V^*}_{30\times150}$$

60% compression

$5\times$ faster mat-mat product

Singular values of the data matrix



Objective:
Approximate *all* frequency slices

Problem:
SVD is expensive

Randomized SVD

More efficient in handling large data

Parallel computing environments

- *fast* matrix-vector products

Full data accessed *only* (1-2) times

- *slow* communication and secondary storage

[Halko, N., P. G. Martinsson, and J. A. Tropp, 2011]

Dimensionality-reduction via *R*-SVD

Two-stage approach:

1. *Capture* action of the data $\hat{\mathbf{P}}$ matrix on $k+l$ *random vectors*

$$\hat{\mathbf{Y}} = \hat{\mathbf{P}}\hat{\mathbf{W}}$$

$\hat{\mathbf{W}}$: Gaussian random matrix

l is a small over sampling parameter (1-3)

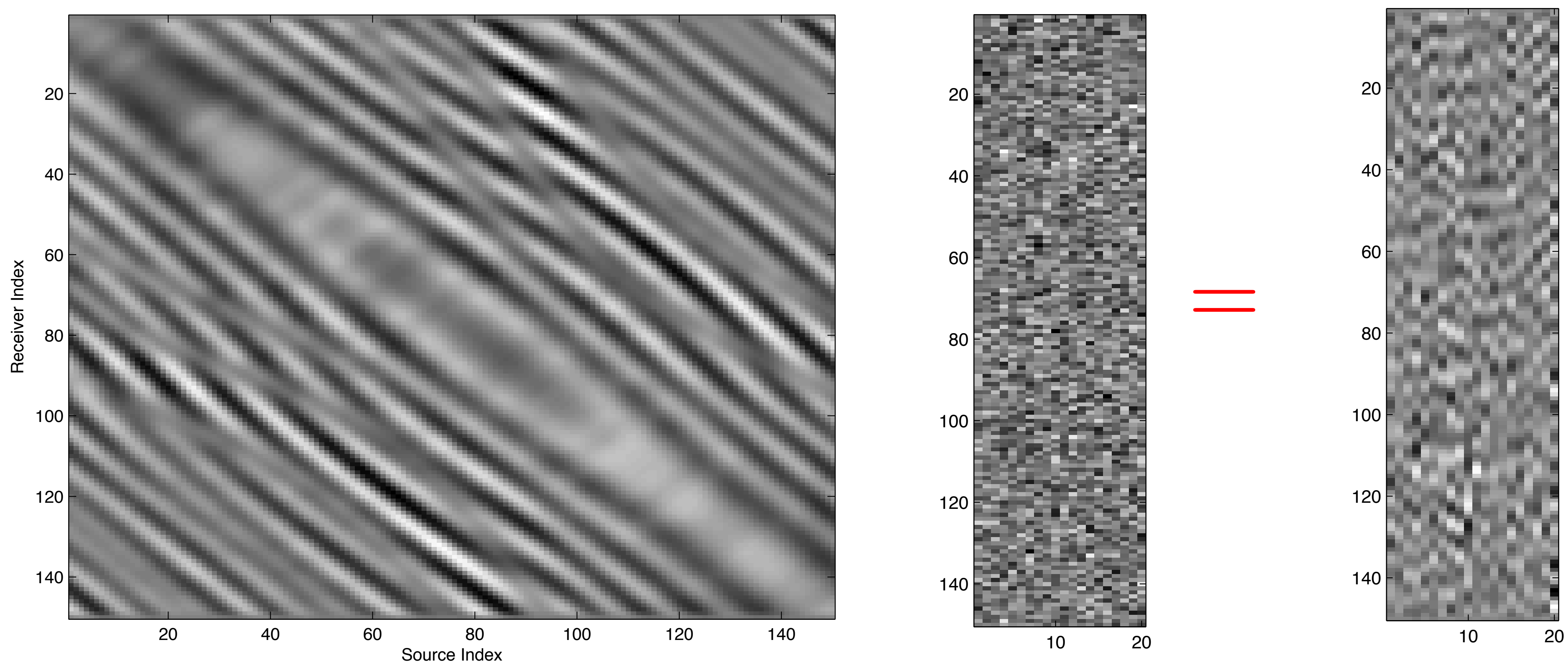
2. *Form* a SVD on $\hat{\mathbf{Y}}$

[Halko, N., P. G. Martinsson, and J. A. Tropp, 2011]

Dimensionality-reduction via *R*-SVD

Stage 1: Capturing the action of $\hat{\mathbf{P}}$

$\hat{\mathbf{P}}$ $\hat{\mathbf{W}}$ $\hat{\mathbf{Y}} = \hat{\mathbf{P}}\hat{\mathbf{W}}$



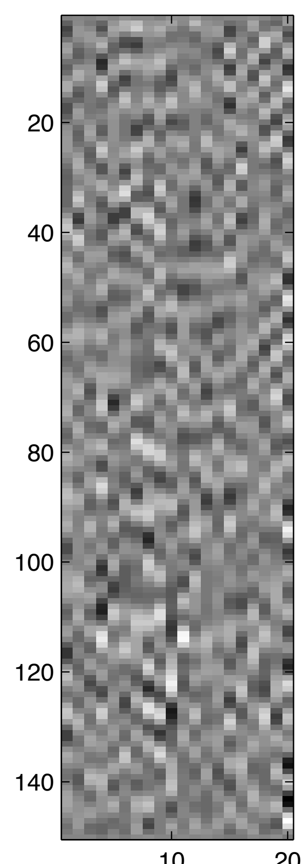
$n_r \times n_s$ $n_s \times k$ $n_r \times k$

Dimensionality-reduction via *R*-SVD

Stage 1: Capturing the action of $\hat{\mathbf{P}}$

2. Form a low-rank QR factorization $\hat{\mathbf{Y}} \approx \mathbf{QR}$

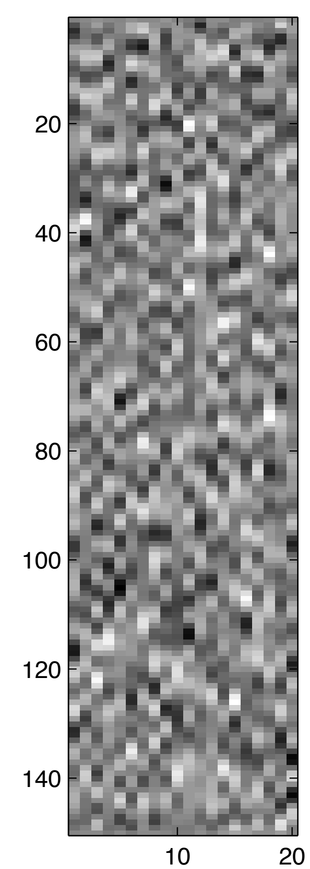
$\hat{\mathbf{Y}}$



$n_r \times k$

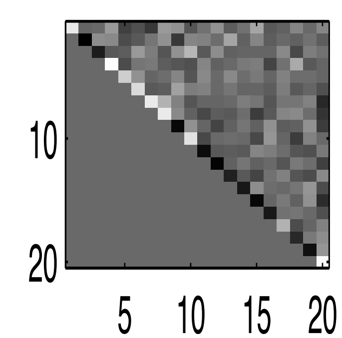
$=$

$\mathbf{Q}$



$n_r \times k$

$\mathbf{R}$



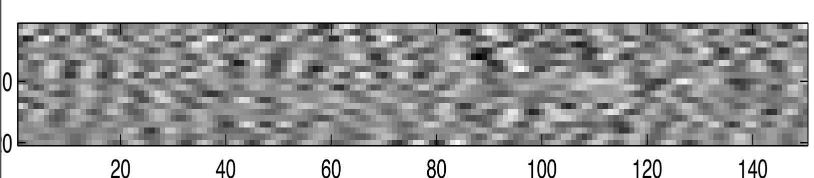
$k \times k$

Dimensionality-reduction via *R*-SVD

Stage 2 : Compute an approximate SVD of $\hat{\mathbf{P}}$

1. Form $\mathbf{B} = \mathbf{Q}^*\hat{\mathbf{P}}$

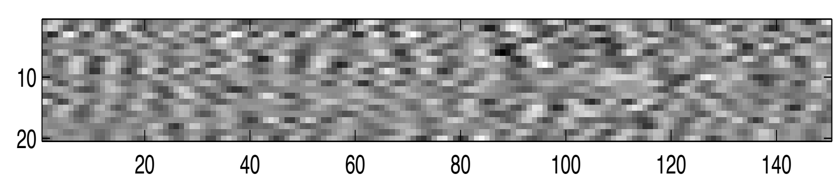
$\mathbf{B}$



$k \times n_s$

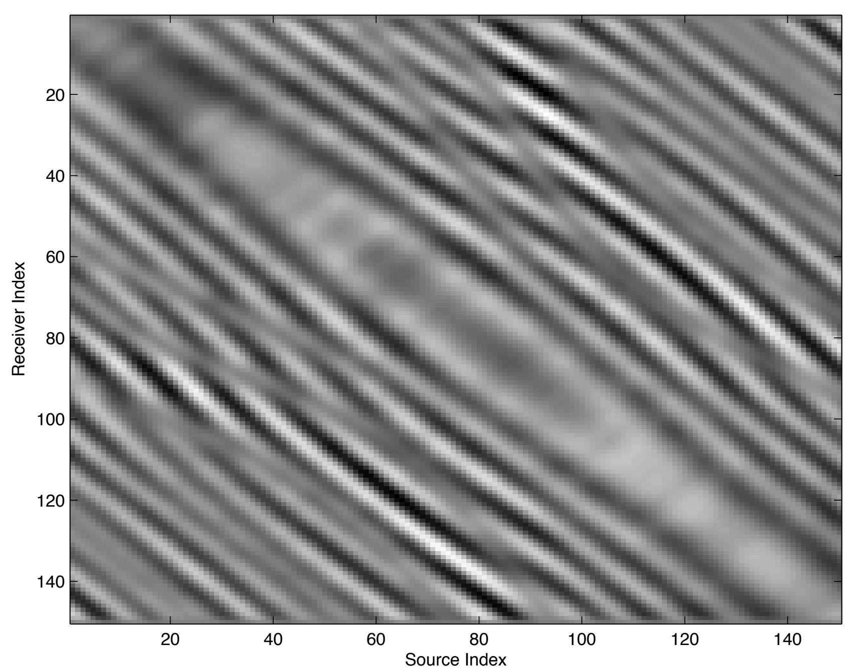
=

$\mathbf{Q}^*$



$k \times n_r$

$\hat{\mathbf{P}}$



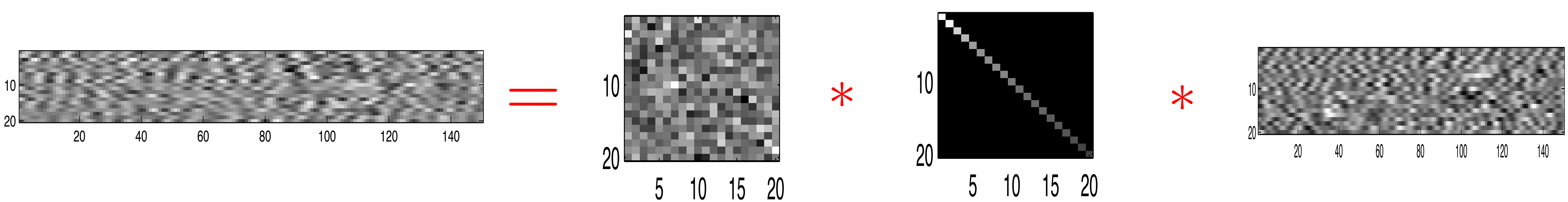
$n_r \times n_s$

Dimensionality-reduction via *R*-SVD

Stage 2 : Compute an approximate SVD of $\hat{\mathbf{P}}$

2. Compute **SVD** of the small matrix $\mathbf{B} = \widetilde{\mathbf{U}}\mathbf{\Sigma}\mathbf{V}^*$

$\mathbf{B}$ $\qquad$ $\widetilde{\mathbf{U}}$ $\qquad$ $\mathbf{S}$ $\qquad$ $\mathbf{V}^*$



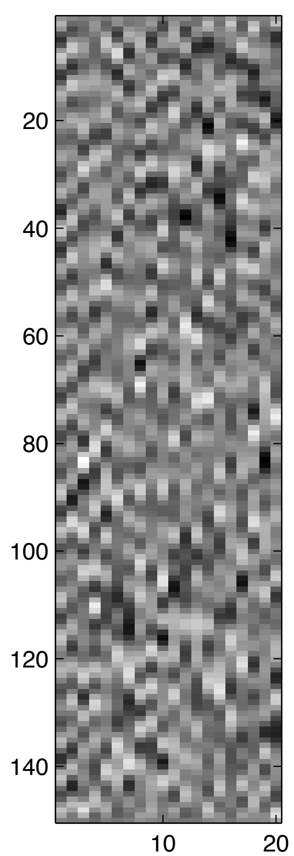
$k * n_s$ $\qquad$ $k * k$ $\qquad$ $k * k$ $\qquad$ $k * n_s$

Dimensionality-reduction via *R*-SVD

Stage 2 : Compute an approximate SVD of $\hat{\mathbf{P}}$

3. Compute $\mathbf{U} = \mathbf{Q}\widetilde{\mathbf{U}}$

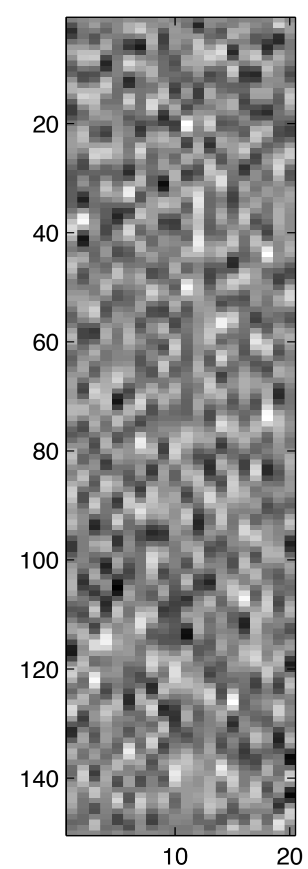
$\mathbf{U}$



$n_r \times k$

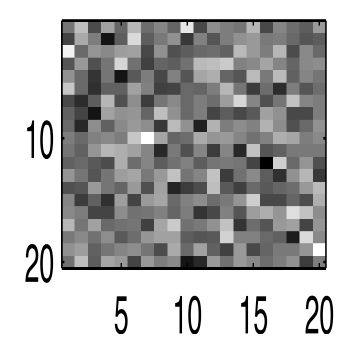
$=$

$\mathbf{Q}$



$n_r \times k$

$\widetilde{\mathbf{U}}$

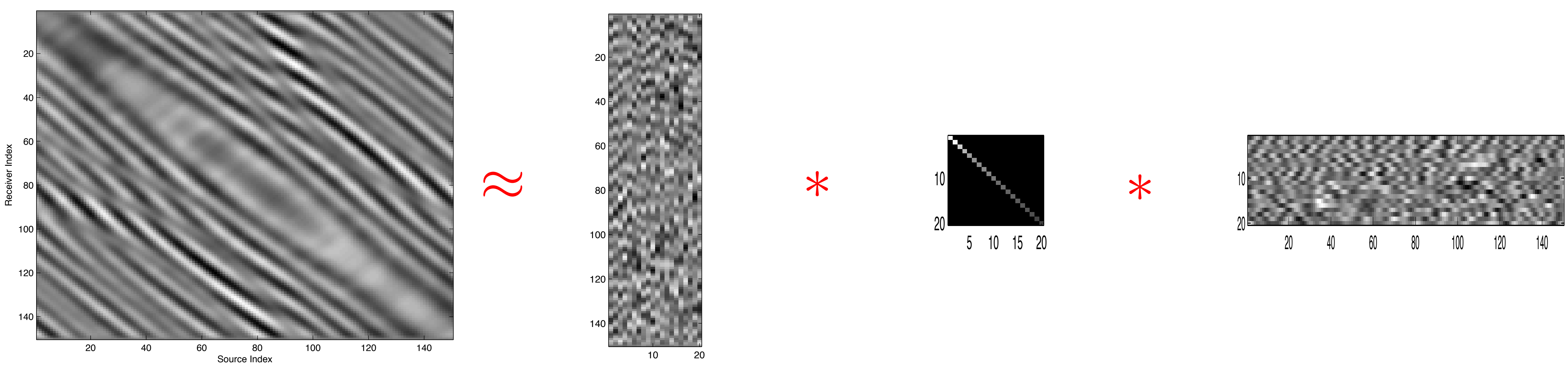


$k \times k$

Dimensionality-reduction via *R*-SVD

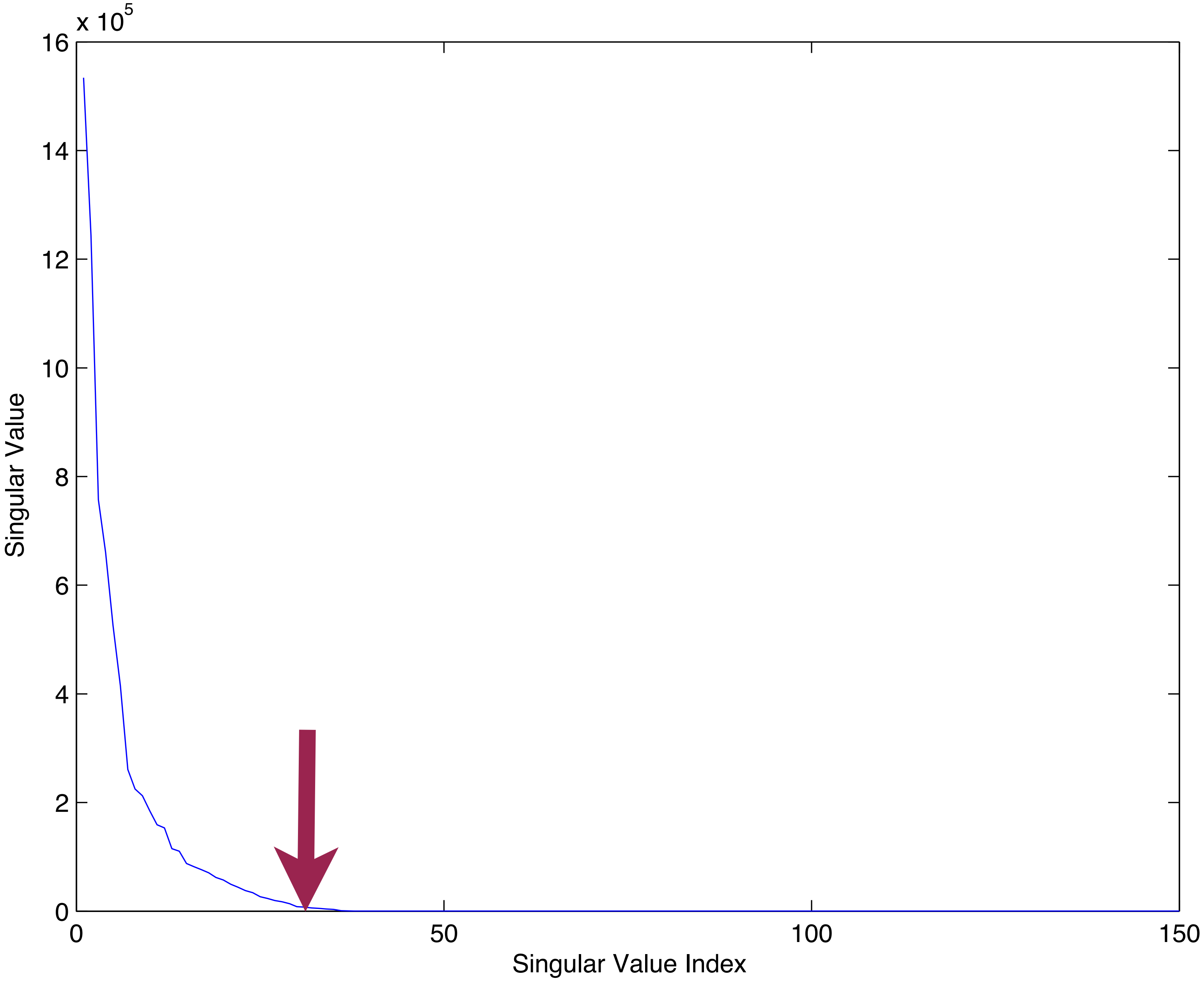
Stage 2 : Compute an approximate SVD of $\hat{\mathbf{P}}$

$$\hat{\mathbf{P}} \approx \mathbf{U} * \mathbf{S} * \mathbf{V}^*$$



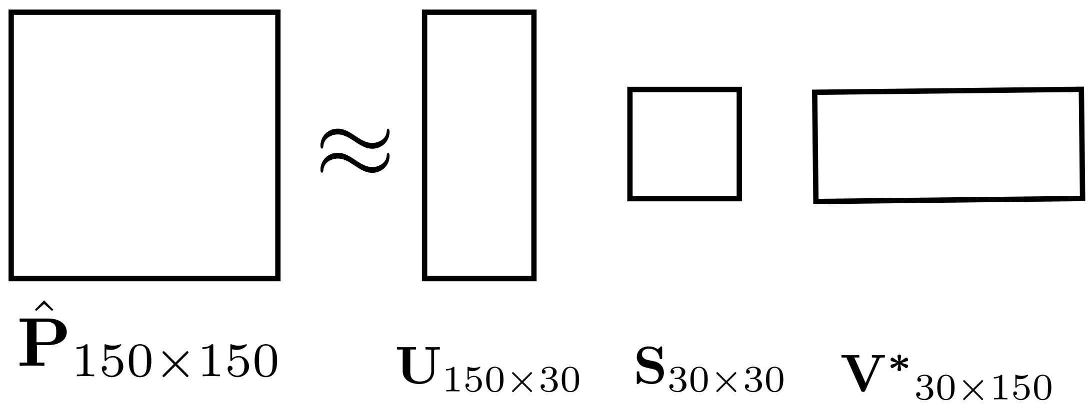
$n_r \times n_s$ | $n_r \times k$ | $k \times k$ | $k \times n_s$

Decay of singular-values



full matrix size$= 150 \times 150$

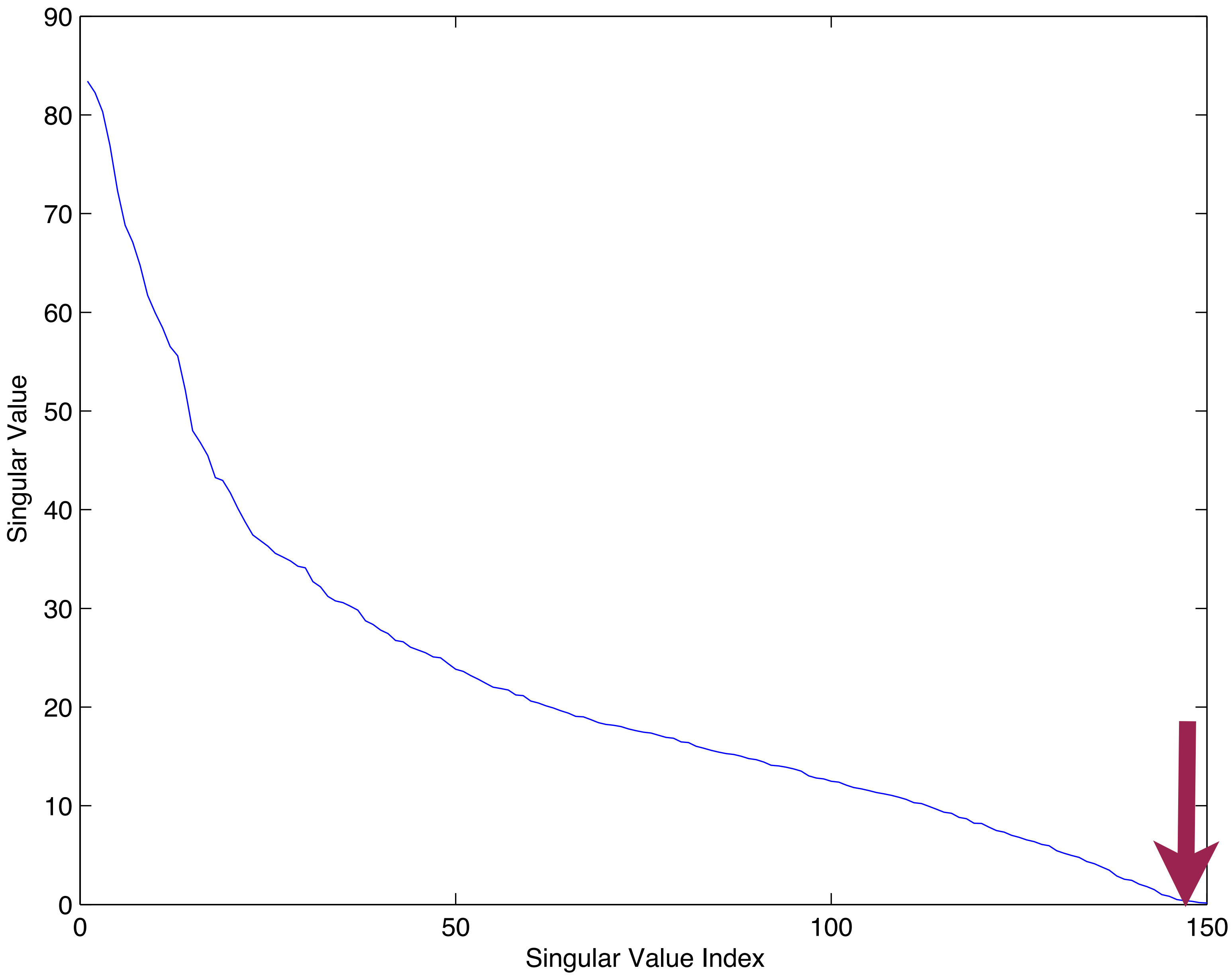
rank $k = 30 = 20\%$



60% compression

$5\times$ faster mat-mat product

Slow decay of singular-values



full matrix size$= 150 \times 150$
rank $k \approx 150$

$$\hat{\mathbf{P}}_{150\times150} \approx \mathbf{U}_{150\times150}\,\mathbf{S}_{150\times150}\,\mathbf{V^*}_{150\times150}$$

No Compression
Slower operations

SLIM

Slow decay of singular-values

Power Iteration

$$\mathbf{Y} = (\widehat{\mathbf{P}}\widehat{\mathbf{P}}^*)^{q}\widehat{\mathbf{P}}\widehat{\mathbf{W}}$$

Small singular values

Interfere with the approximation

Solution

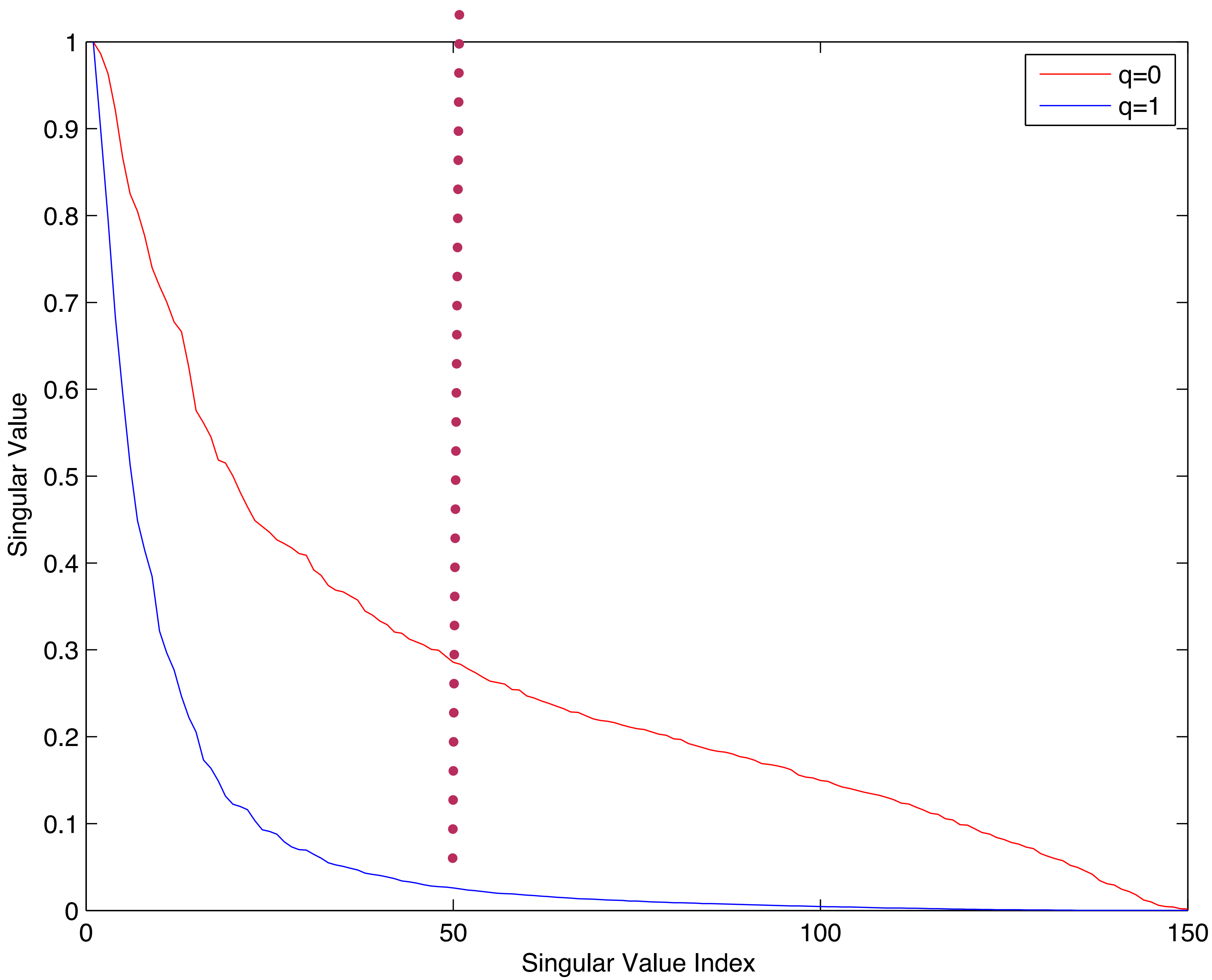
Reduce their weight

Cost

q passes over the data

SLIM

Power Iteration



$q = 0$

$$\text{SNR} = 4 \text{ dB}$$

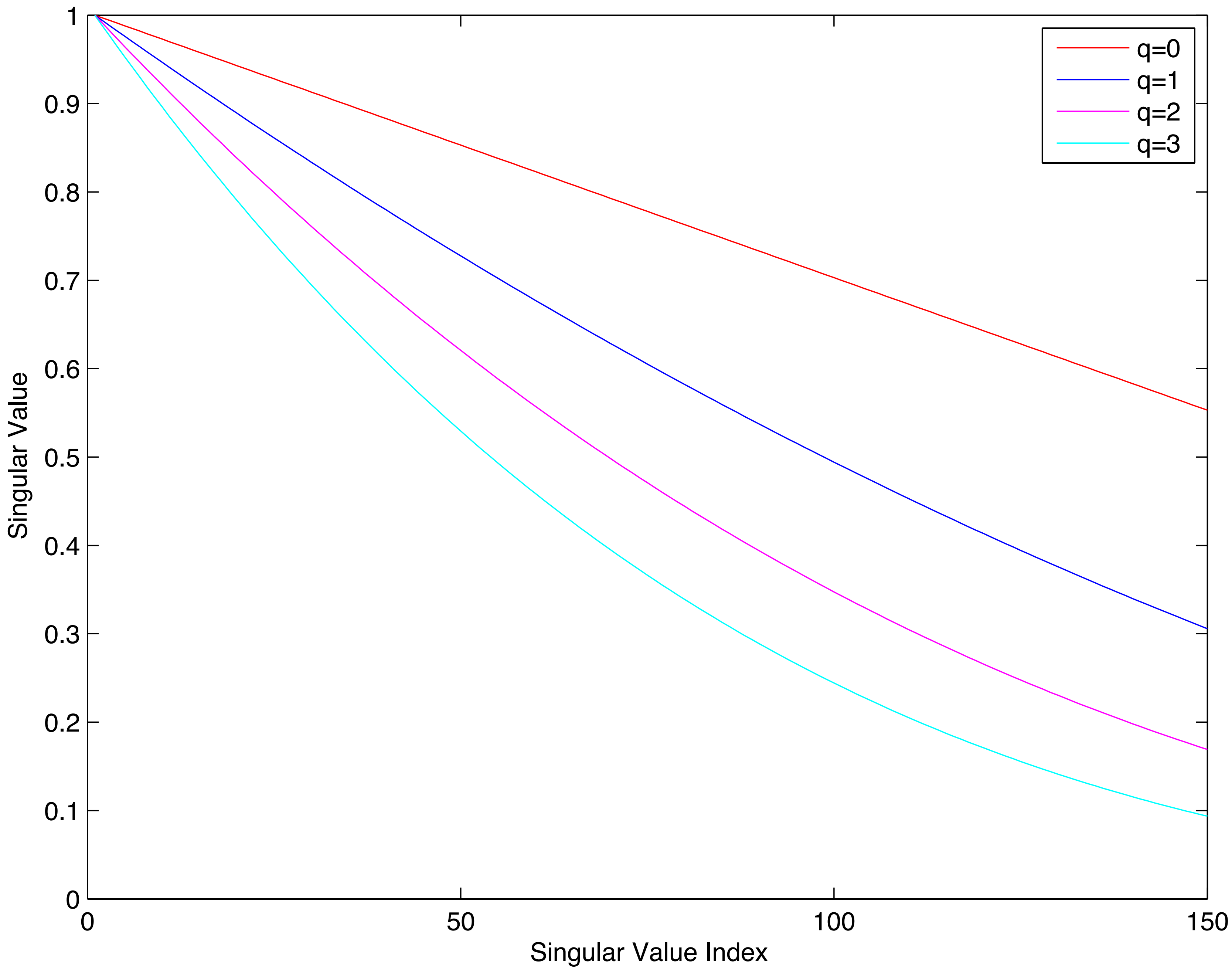
$q = 1$

$$\text{SNR} = 12 \text{ dB}$$

Cost

- Additional q passes

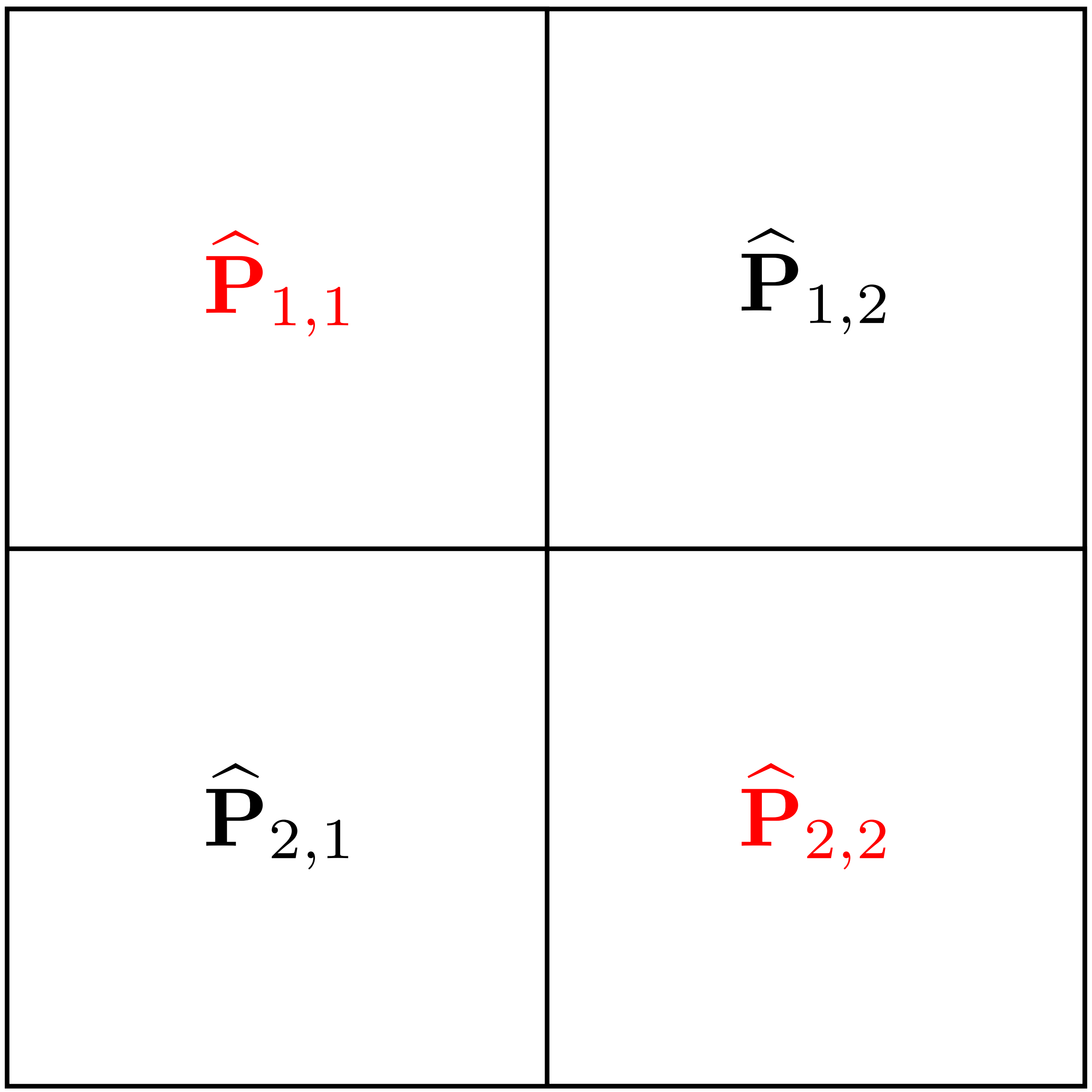
No decay of singular-values



Power Iteration

Not Effective

Hierarchical semi-separable representation



Dense matrices

Level 1 : HSS partitioning

High-rank blocks

Next HSS level

Low-rank blocks

R-SVD

[P.G. Martinsson, 2010]

Hierarchical semi-separable representation

$\widehat{\mathbf{P}}_{1,1}$	$(\mathbf{USV}^*)_{1,2}$
$(\mathbf{USV}^*)_{2,1}$	$\widehat{\mathbf{P}}_{2,2}$

Dense matrices

Level 1 : HSS partitioning

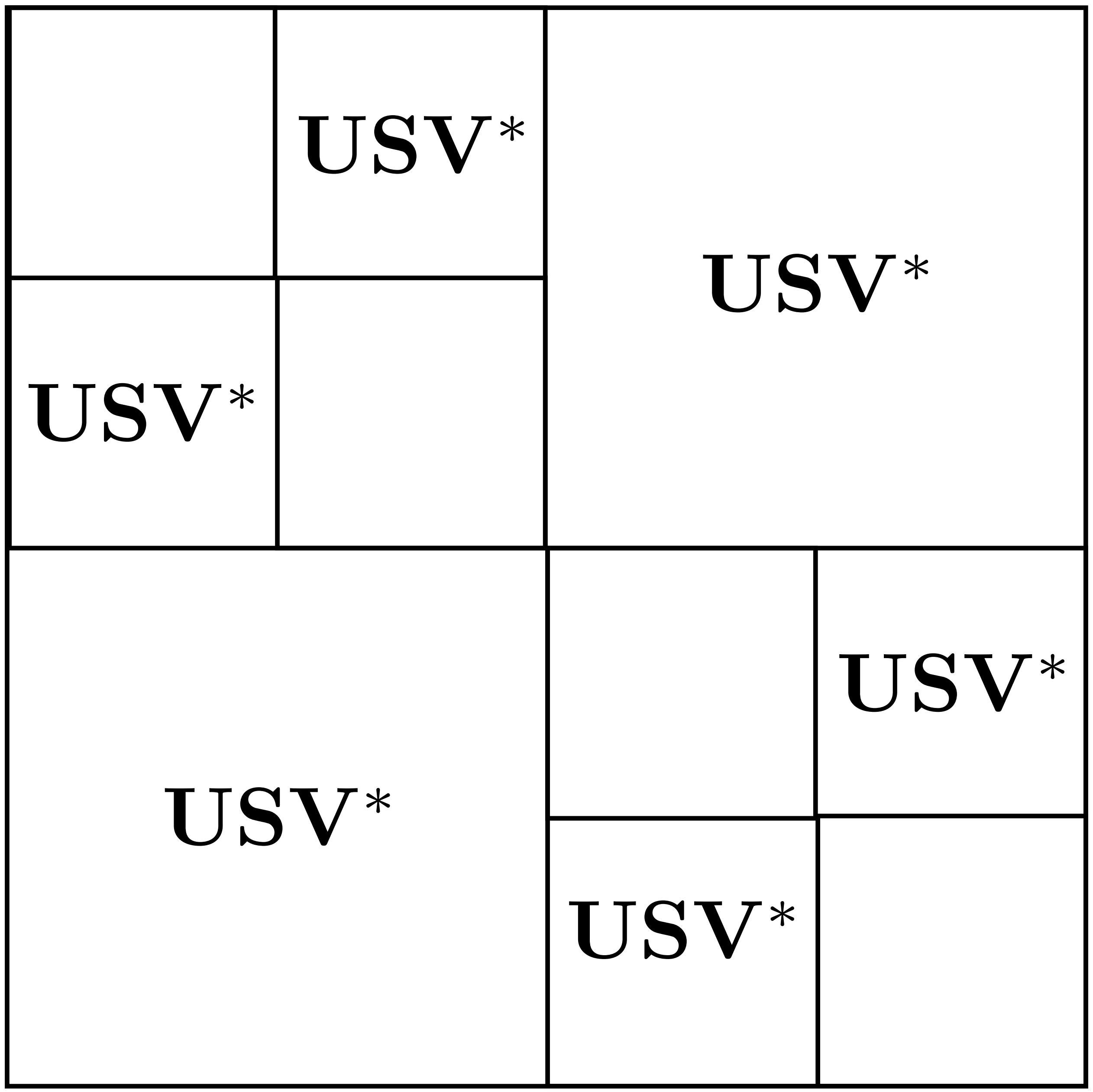
High-rank blocks

Next HSS level

Low-rank blocks

R-SVD

Hierarchical semi-separable representation



Dense matrices

Level 2 : HSS partitioning

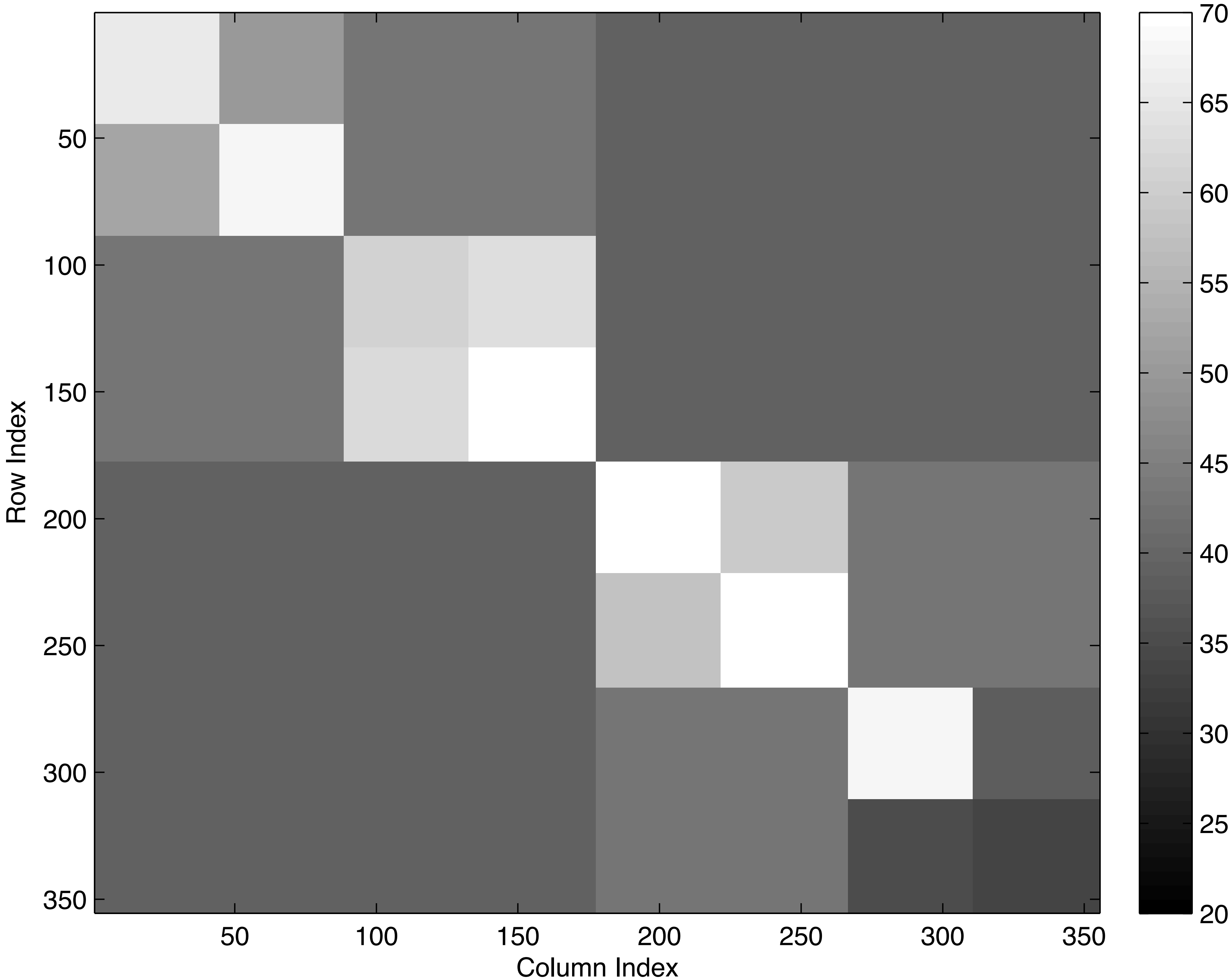
High-rank blocks

Next HSS level

Low-rank blocks

R-SVD

Hierarchical semi-separable representation



Frequency slice

Level 2 : HSS partitioning

High-rank blocks

White

Low-rank blocks

Black

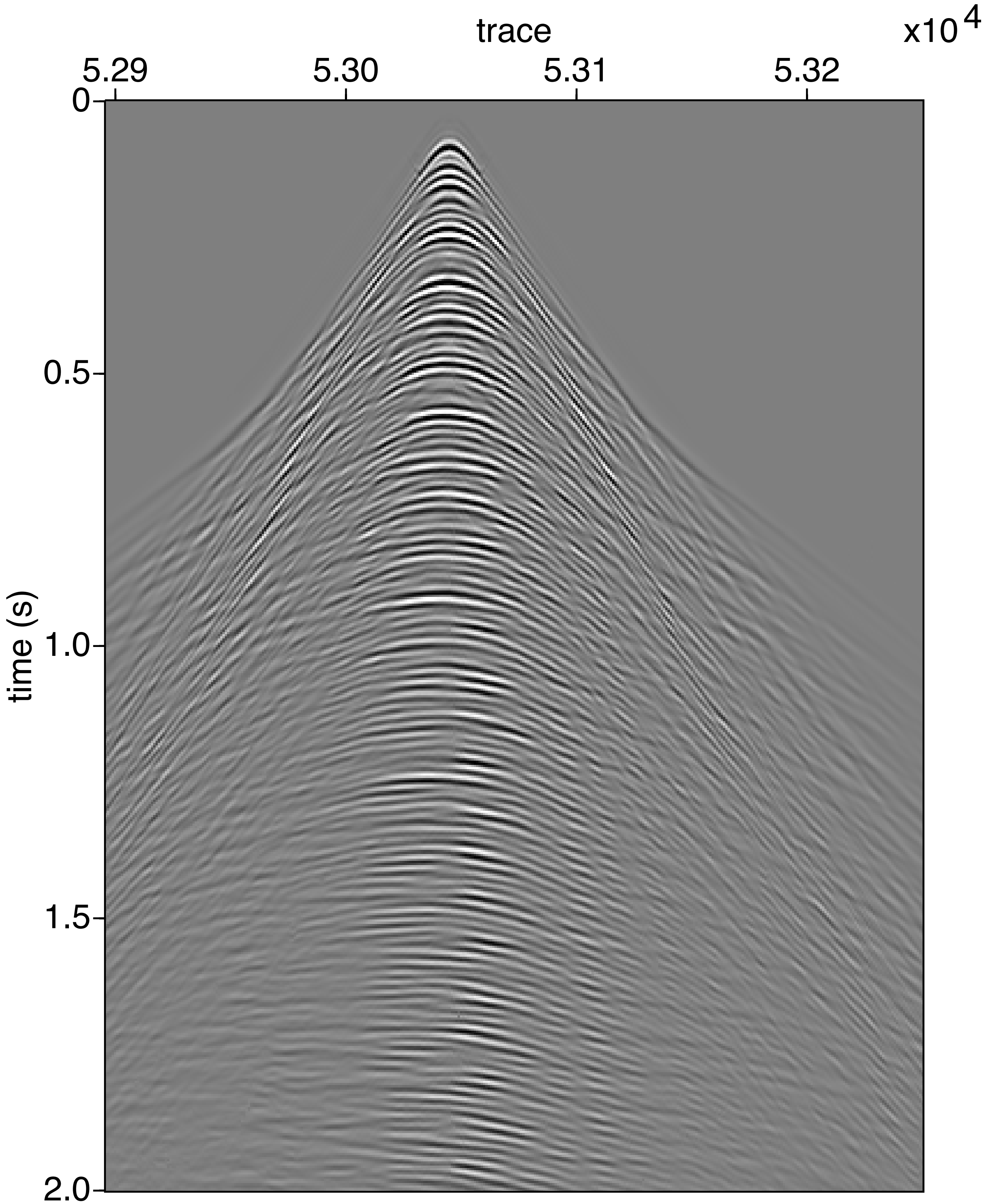
SLIM

EPSI problem

$$\mathbf{Ax} \approx \mathbf{b}$$

$$\tilde{\mathbf{x}} = \underbrace{\arg\min_{\mathbf{x}} ||\mathbf{x}||_{\mathbf{1}}}_{\text{sparsity promoting part}} \quad \text{subject to} \quad \underbrace{||\mathbf{Ax} - \mathbf{b}||_{\mathbf{2}} \leq \sigma}_{\text{data fitting part}}$$

σ : residual between the recorded & predicted data



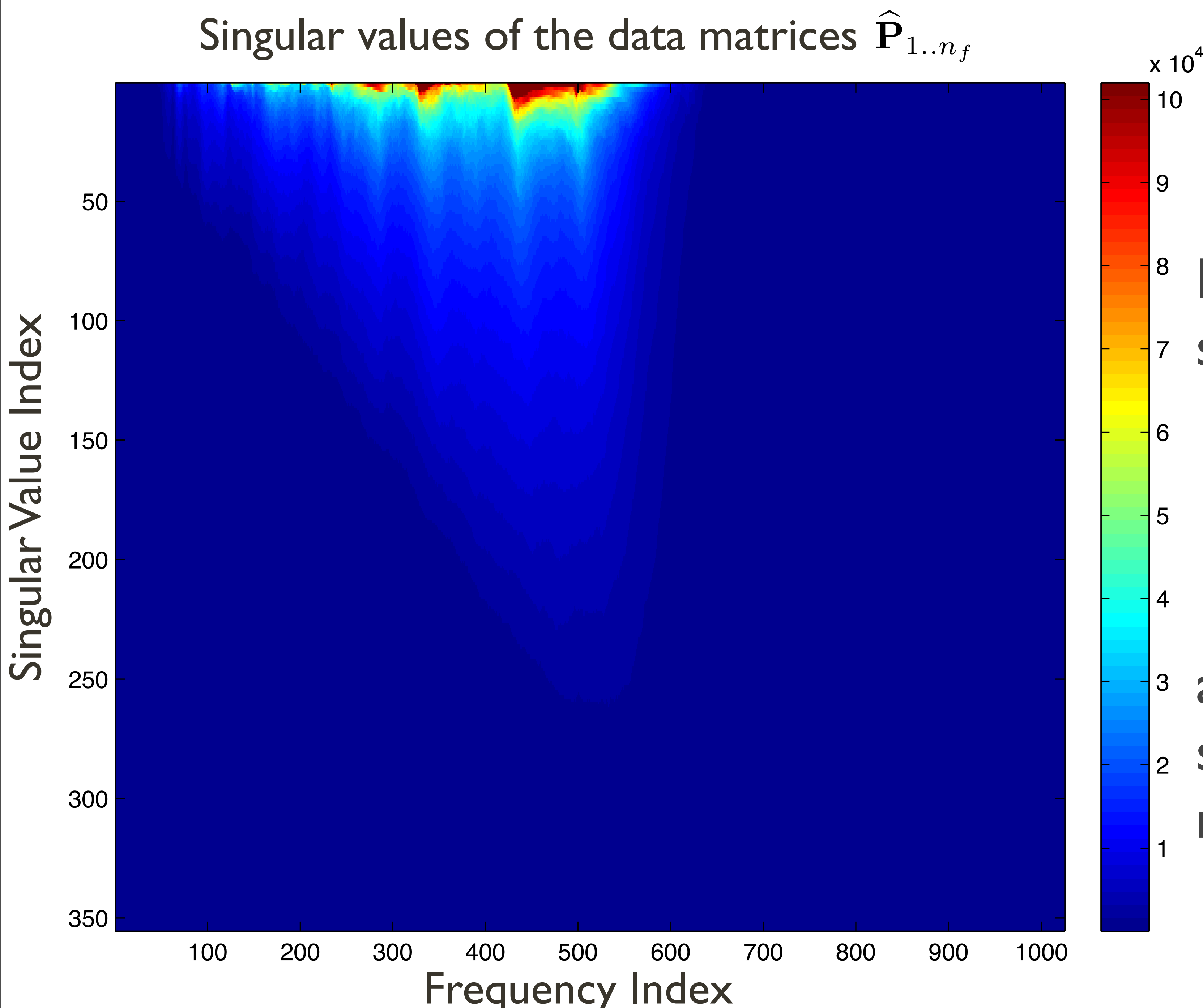
Gulf of Suez

Total Data

shot gather

$$n_r = 355$$
$$n_s = 355$$
$$n_t = 1024$$
$$dt = .004s$$

Singular values of the data matrix



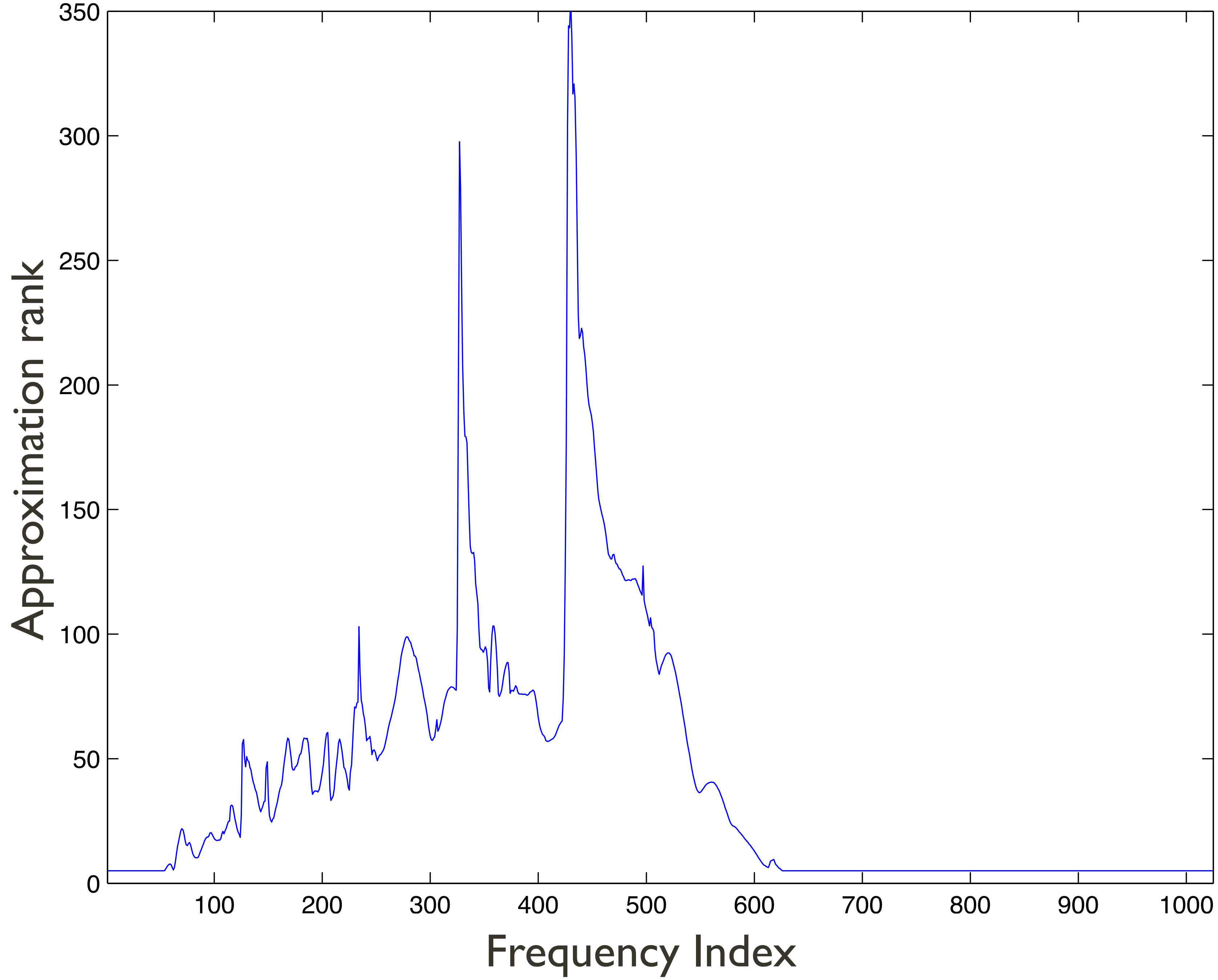
Adaptive approximation

For each frequency find rank k such that

$$\|\mathbf{P} - \mathbf{USV}^*\| \leq \epsilon$$

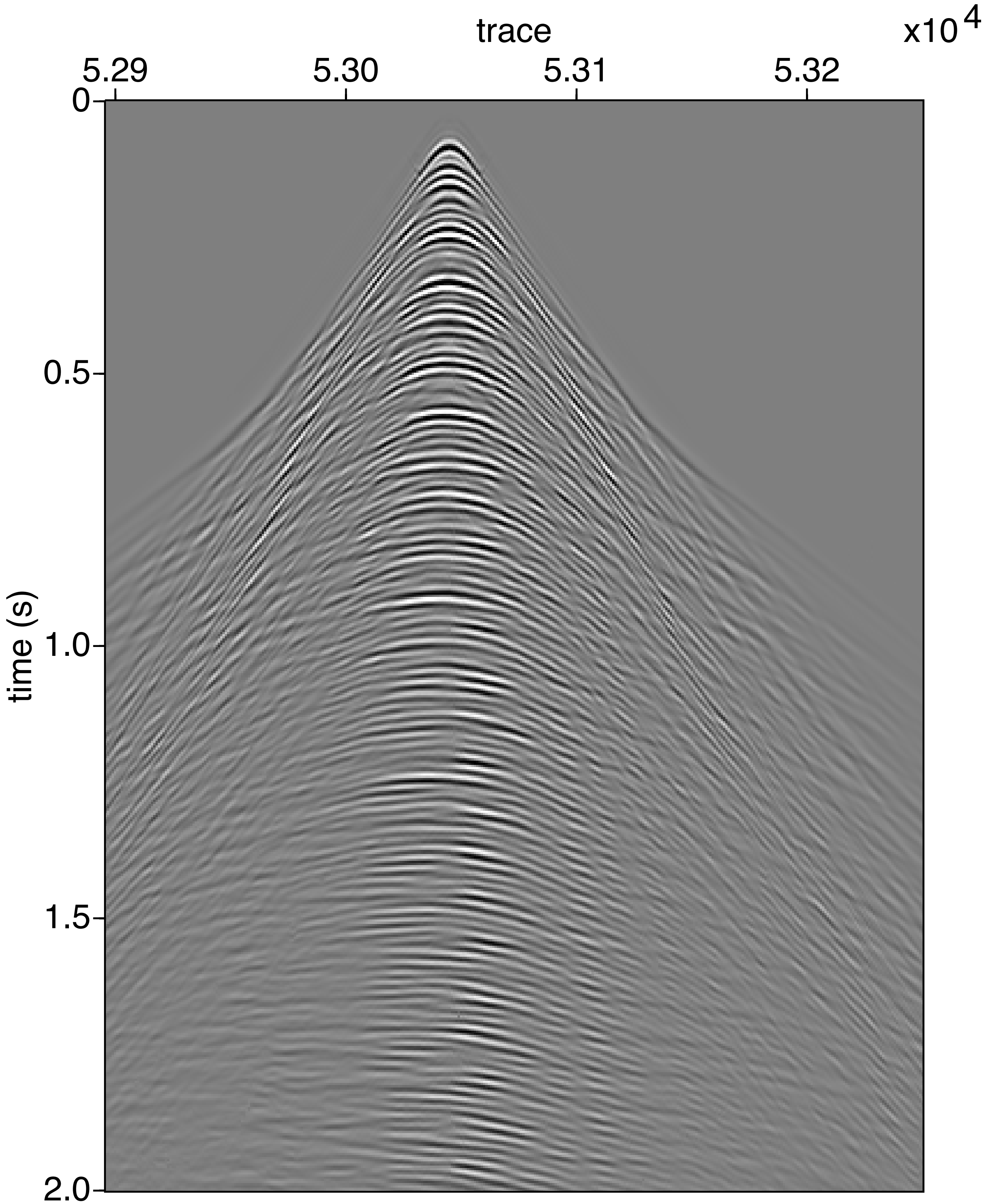
and the sum of all k ranks used is smaller than some % of total number of columns

Adaptive rank selection



sum of all k ranks used is 9% of total number of columns

SLIM

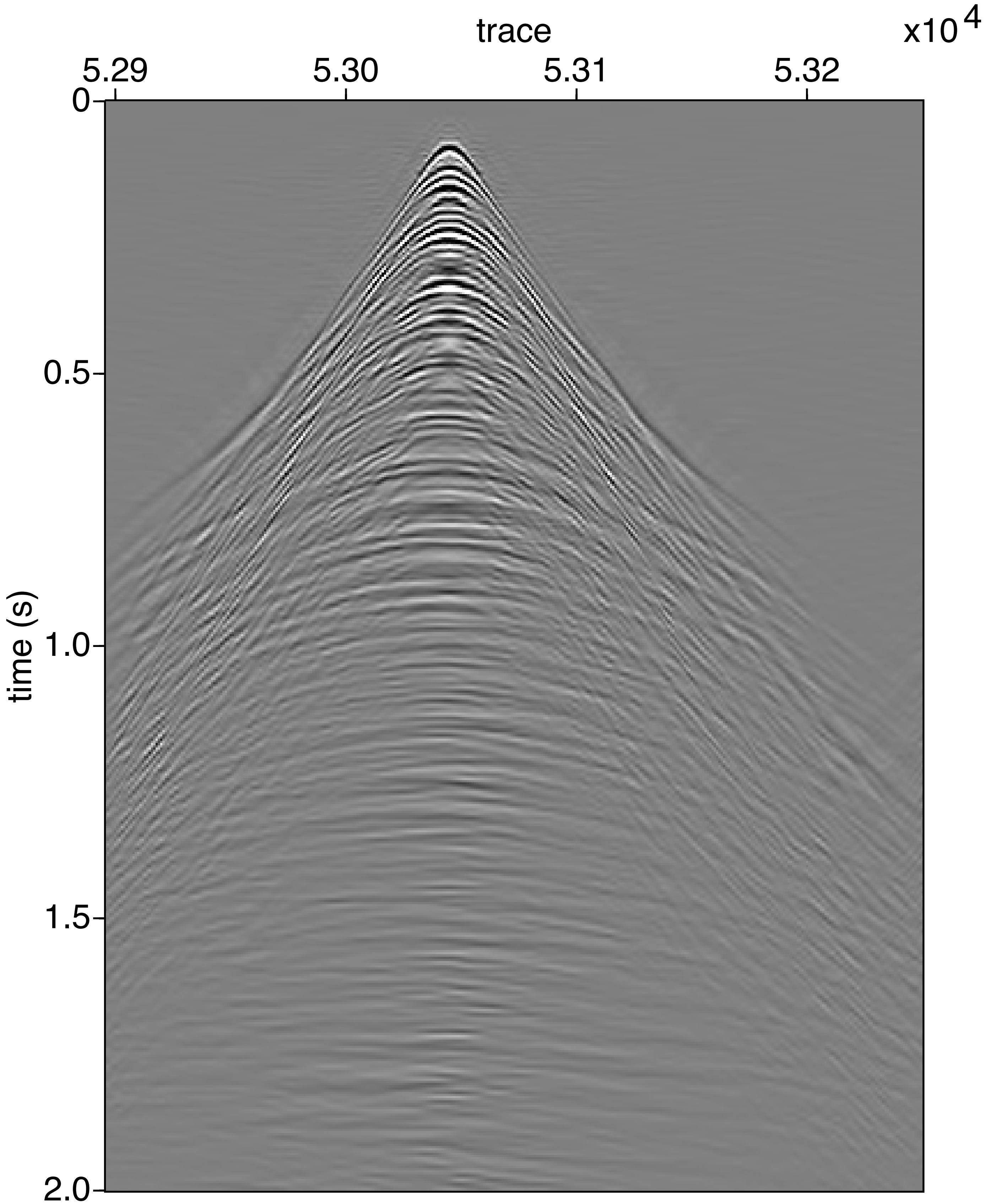


Gulf of Suez

Total Data

shot gather

$$n_r = 355$$
$$n_s = 355$$
$$n_t = 1024$$
$$dt = .004s$$



Gulf of Suez

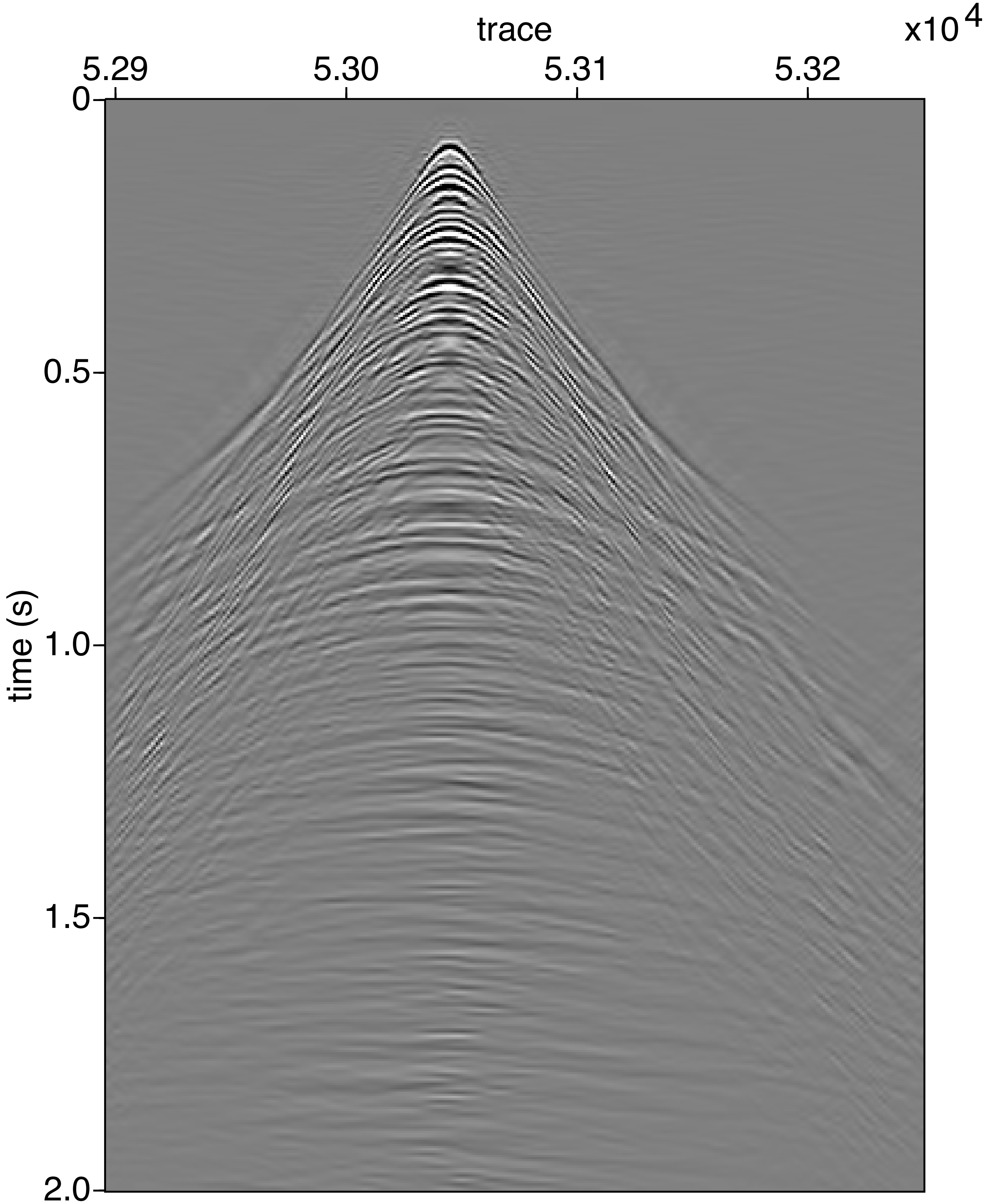
Full Data

Primary IR (G)

shot gather

2D Curvelet (Src-Rcv)

150 iterations



Gulf of Suez

20% of Data

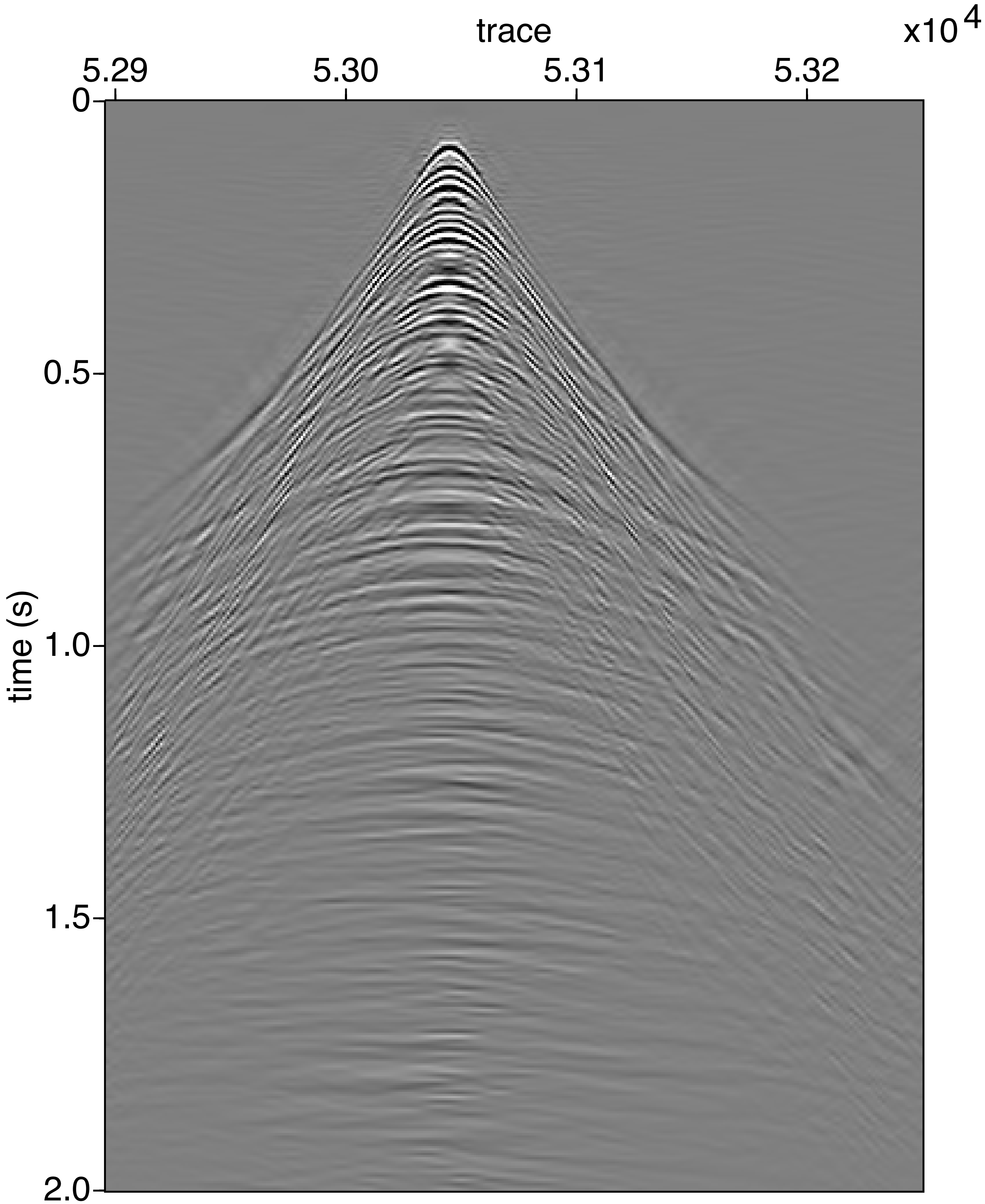
Primary IR (G)

SNR = 27dB

shot gather

2D Curvelet (Src-Rcv)

150 iterations



Gulf of Suez

12% of rank budget

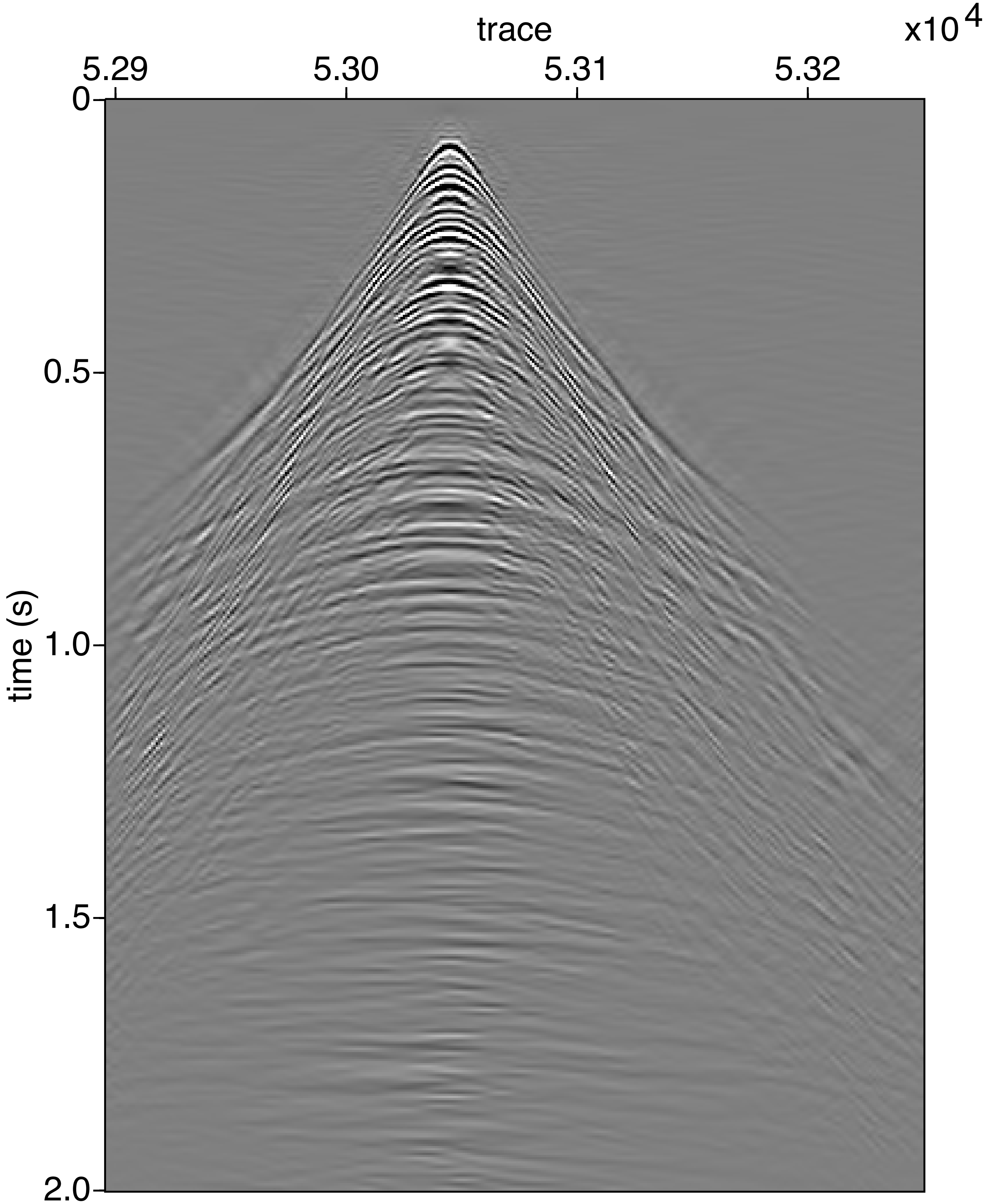
Primary IR (G)

SNR = 17dB

shot gather

2D Curvelet (Src-Rcv)

150 iterations



Gulf of Suez

8% of rank budget

Primary IR (G)

SNR = 12dB

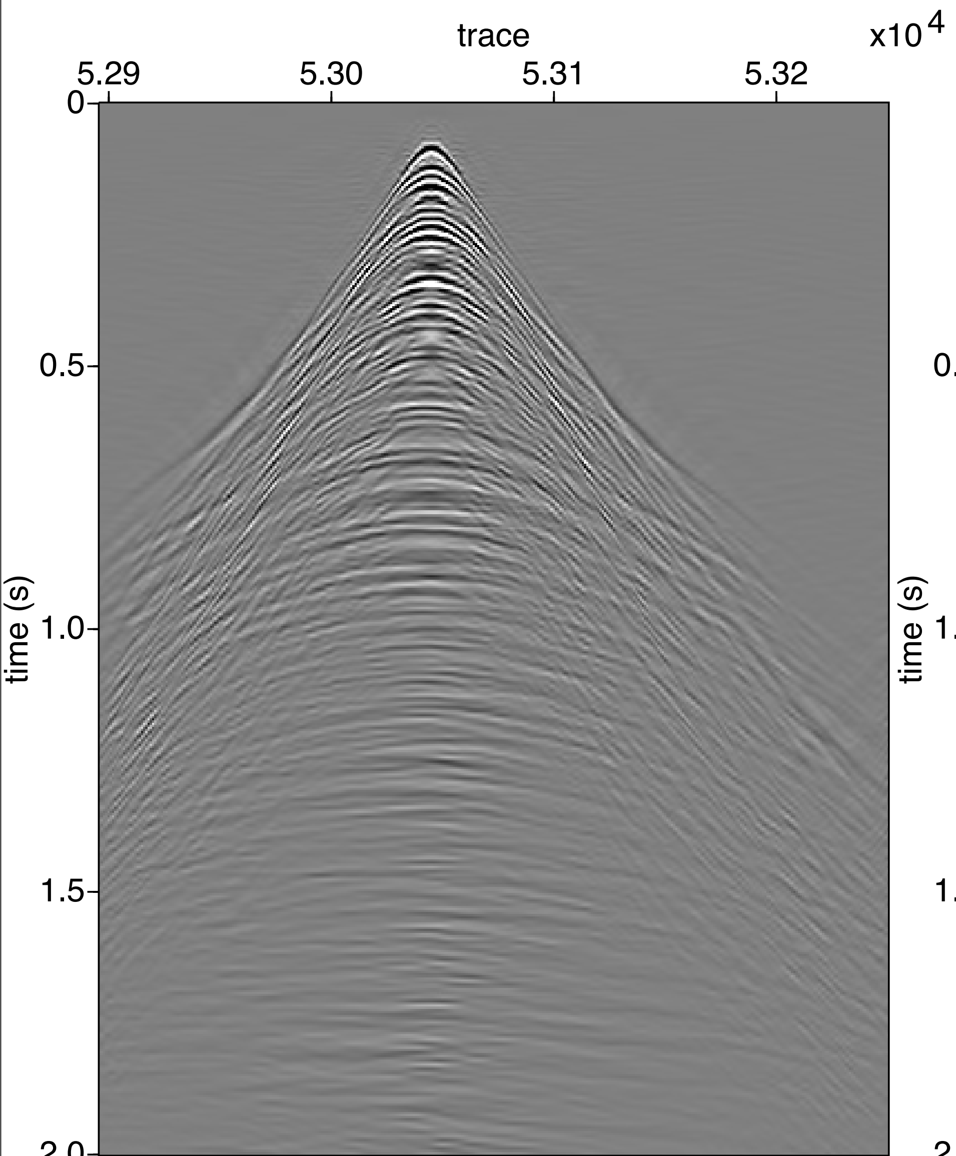
shot gather

2D Curvelet (Src-Rcv)

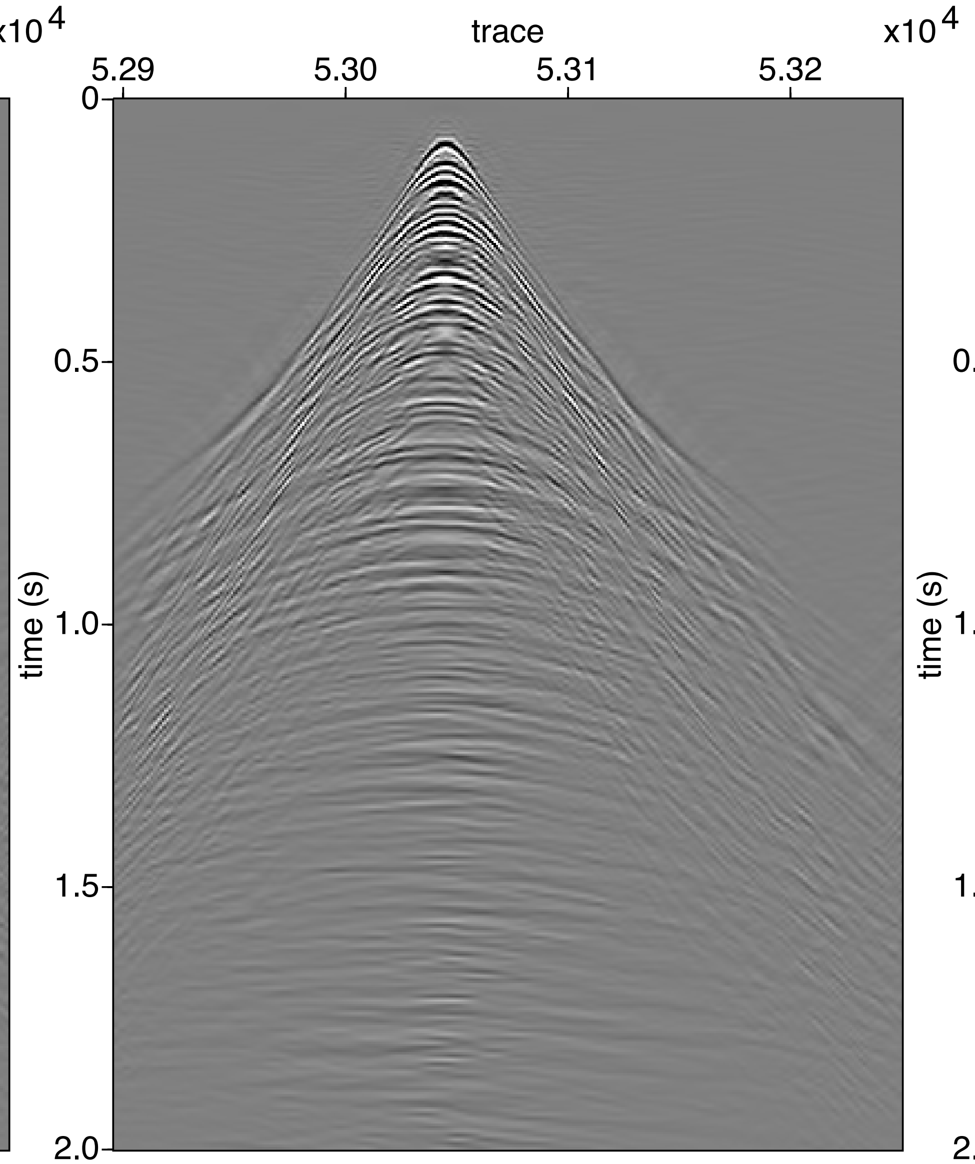
150 iterations

Difference in EPSI Result

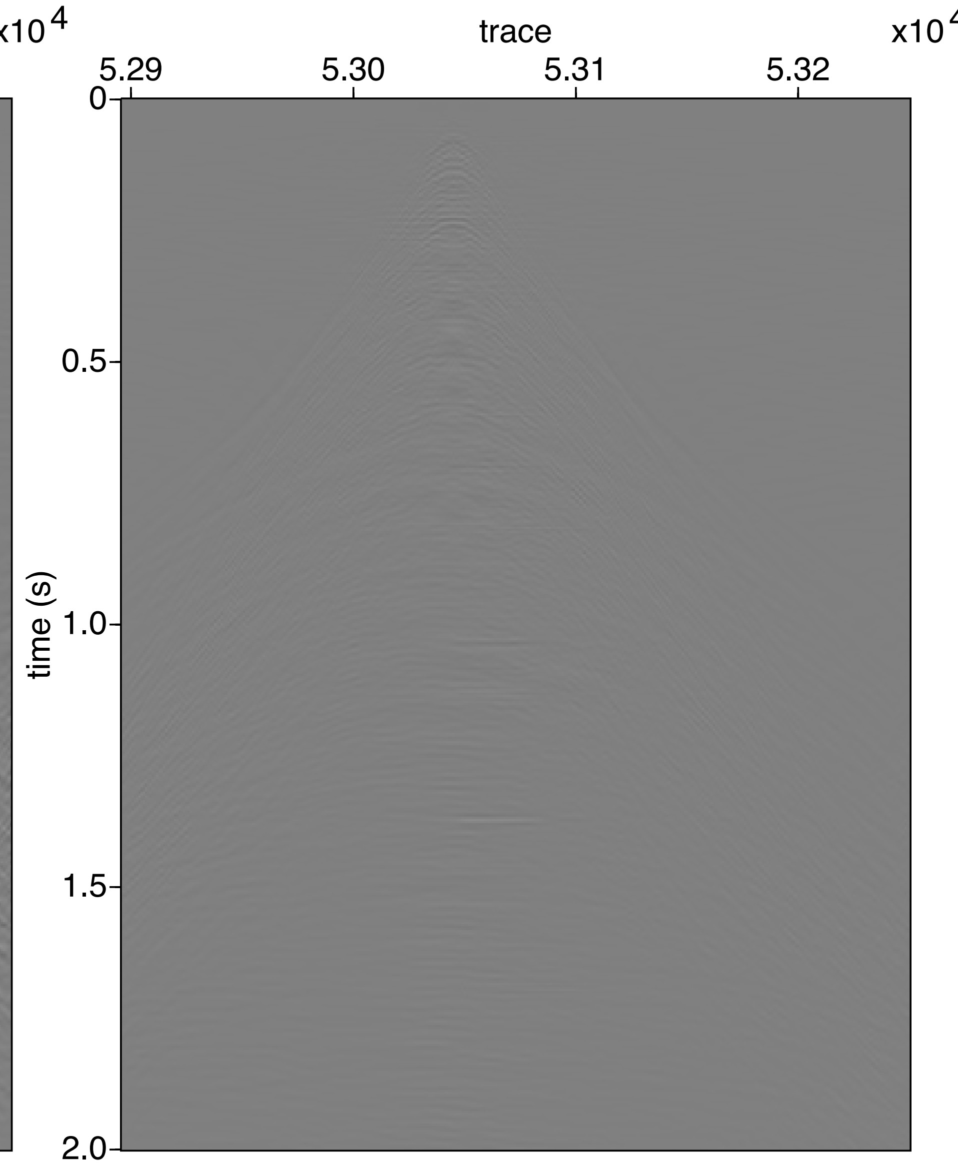
20% rank budget



Primary IR
full data



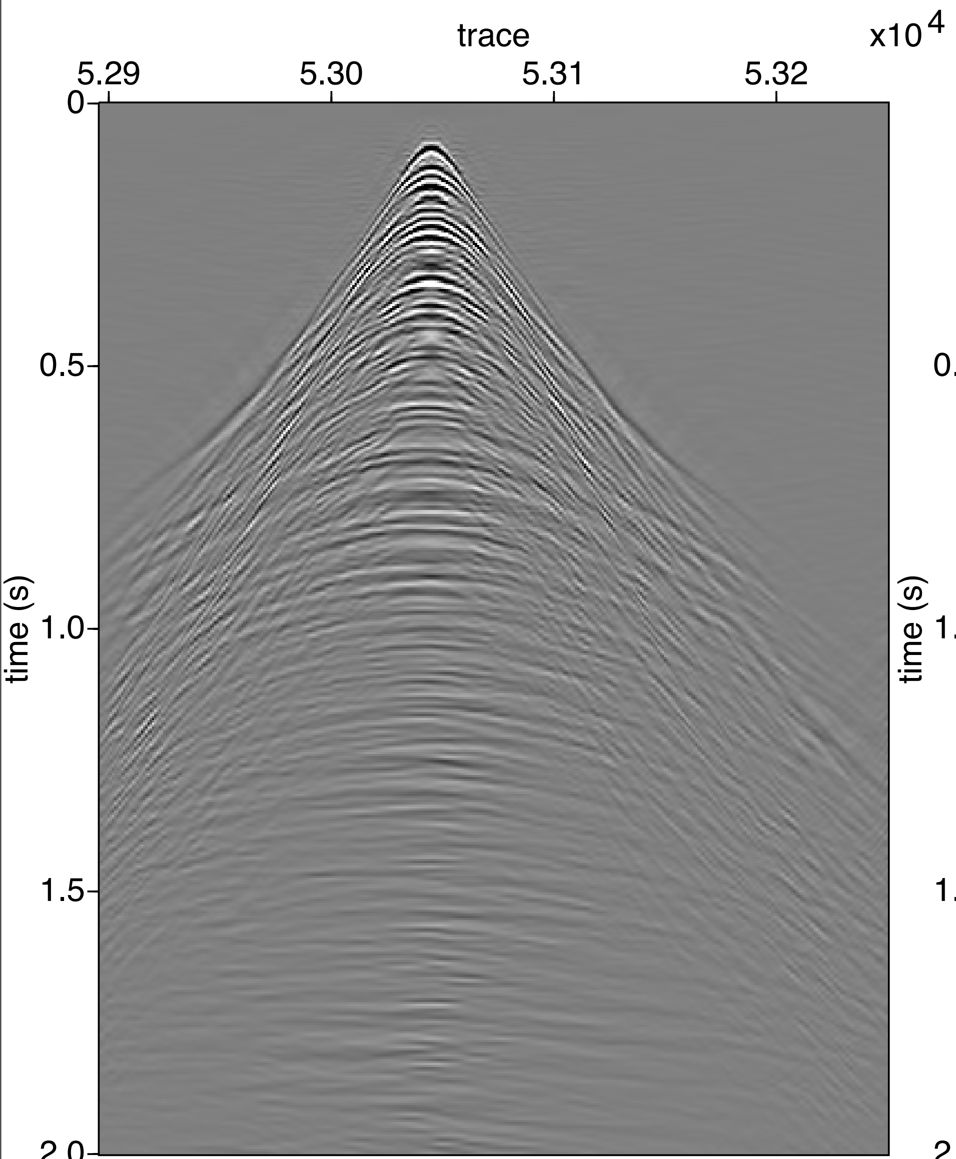
Primary IR
approximated Data



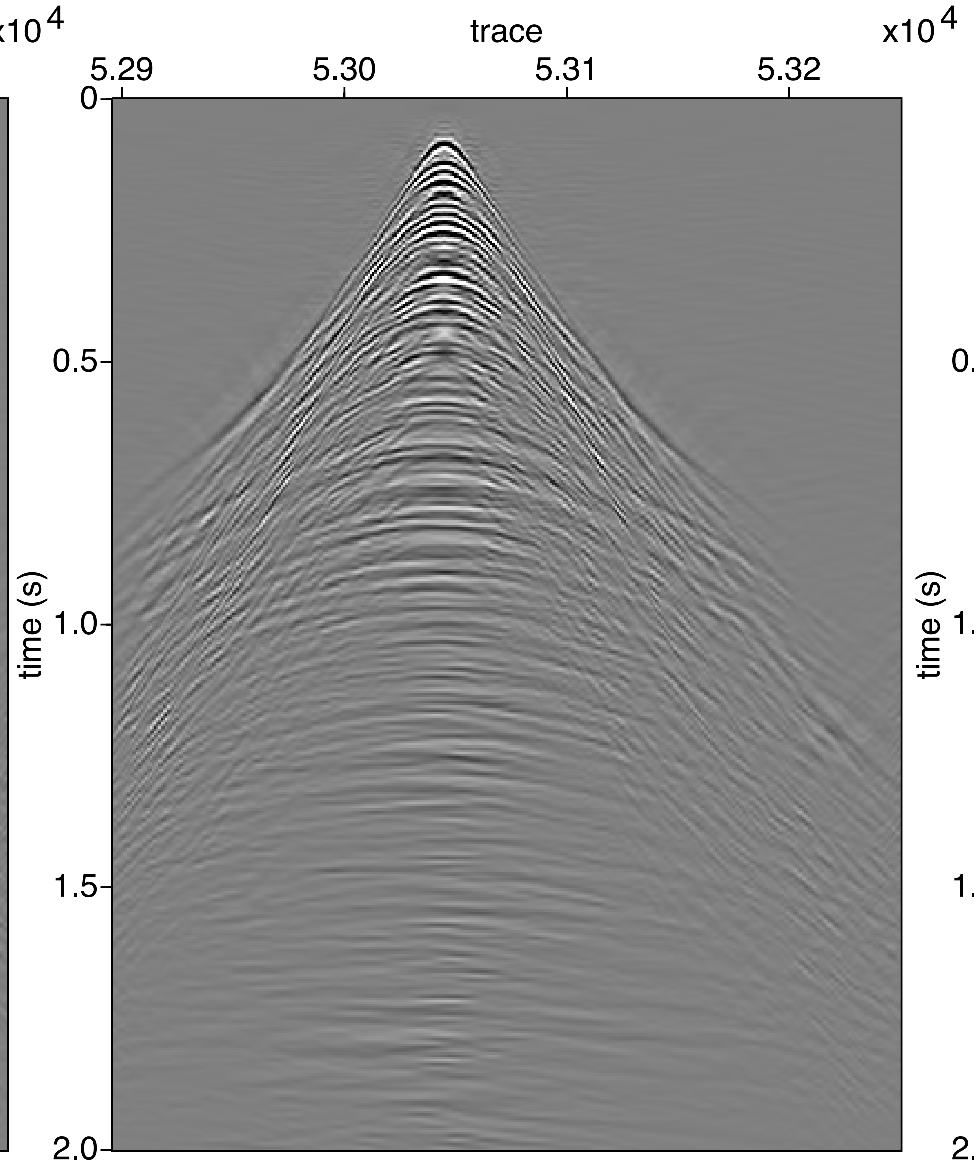
Difference

Difference in EPSI Result

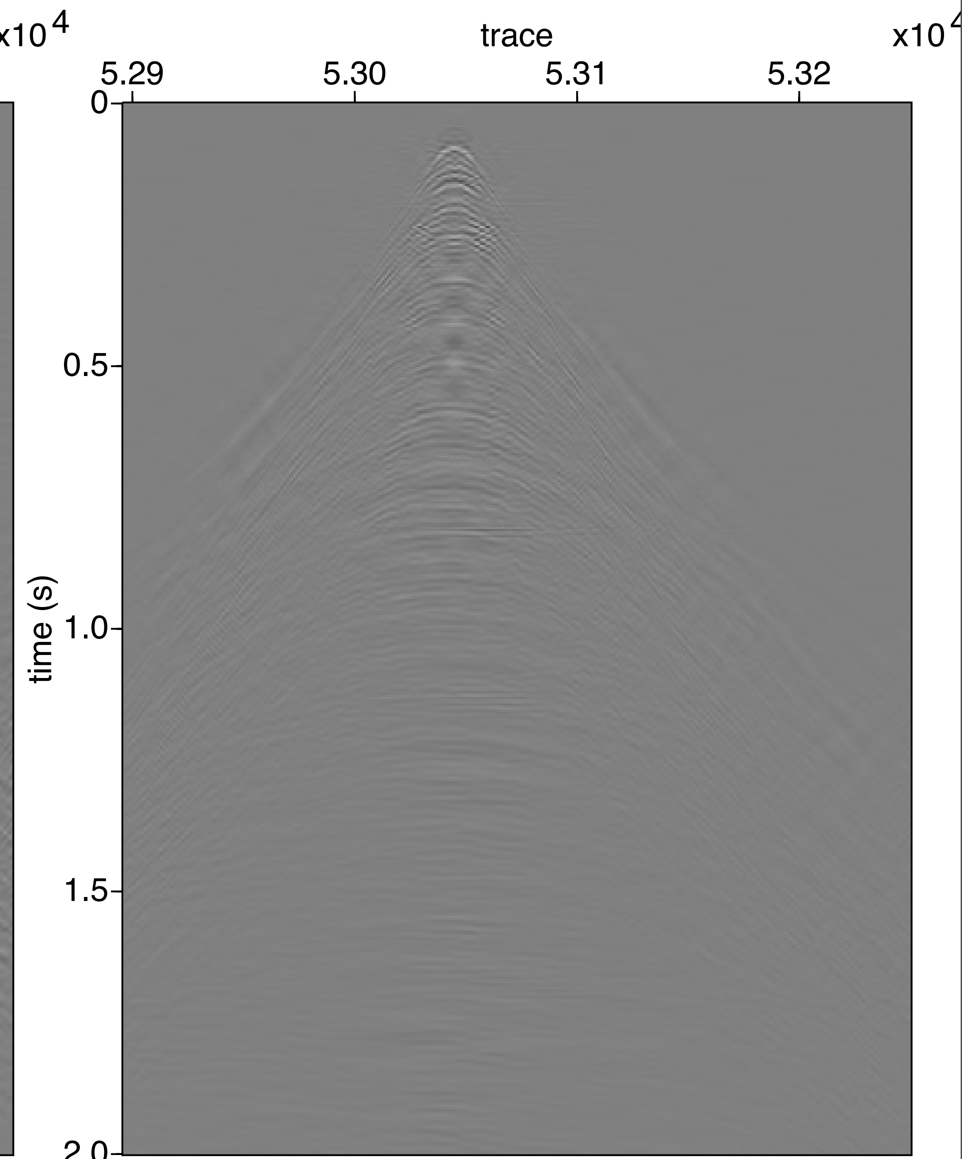
12% rank budget



Primary IR
full data



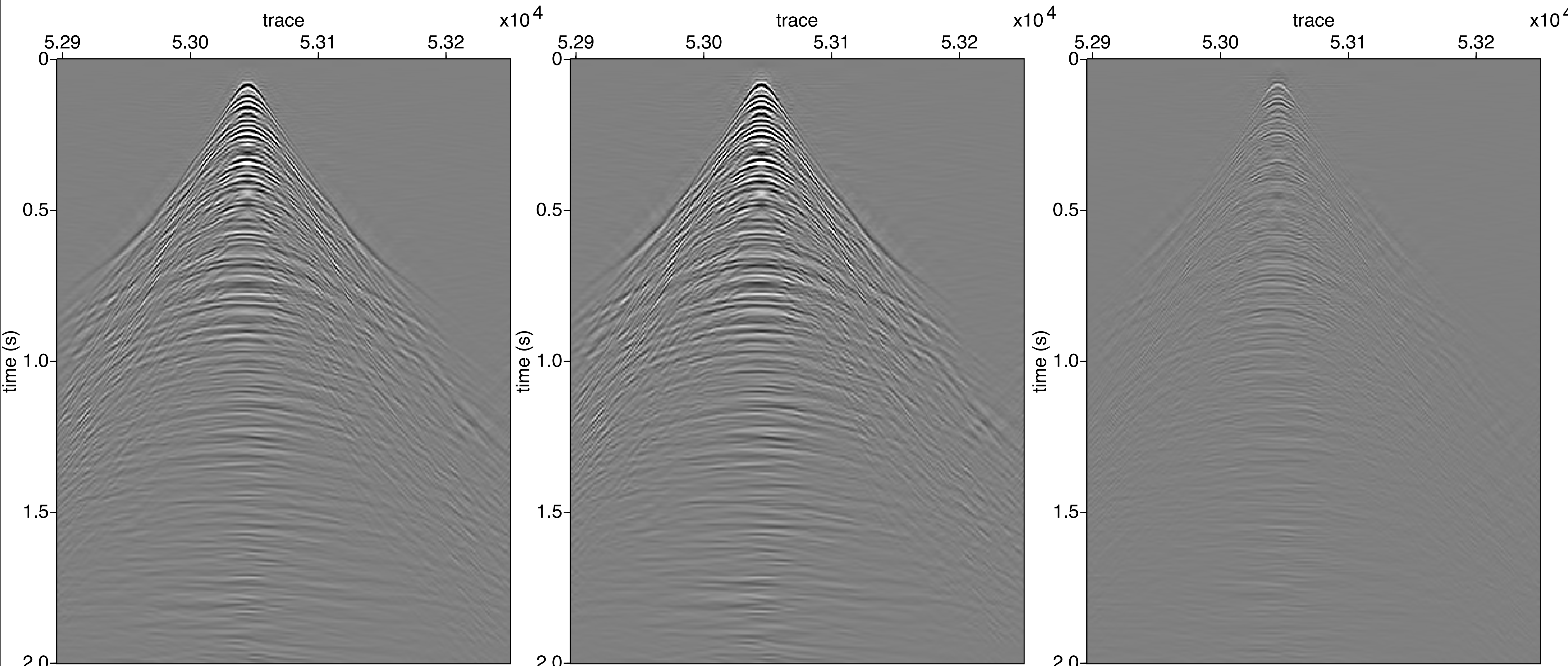
Primary IR
approximated Data



Difference

Difference in EPSI Result

8% rank budget



Primary IR
full data

Primary IR
approximated Data

Difference

Performance Summary

Rank Percentage	50%	20%	12%	8%
SNR (dB)	30	27	17	12
Multiplication Speedup	1.6x	2x	3.5x	5.7x
Memory savings	40%	50%	71%	82%

Conclusions

- Data driven methods - e.g. EPSI - suffers from the 'curse of dimensionality' when moving to 3D
- We utilize insights from *random*-matrix theory to *approximate* action of the *data* matrix
- *Slow* decaying singular values
    - *power* Iterations
    - *HSS* representations
- *Reductions* in *multiplication* and *storage* costs

References

Berg, E. v., and M. P. Friedlander, 2008, Probing the Paretofrontier for basis pursuit solutions: SIAM Journal on Scien-tific Computing, 31, 890–912.

Berkhout, A. J., and D. J. Verschuur, 1997, Estimation of multi-ple scattering byiterative inversion, part I: theoretical considerations: Geophysics, 62, 1586–1595.

Candes, E., and B. Recht, 2009, Exact matrix completion viaconvex optimization: Foundations of Computational Mathe-matics, 9, 717–772.

Gandy, S., B. Recht, and I. Yamada, 2011, Tensor completionand low-n-rank tensor recovery via convex optimization:Inverse Problems, 27, 025010.Habashy, T. M., A. Abubakar, G. Pan, and A. Belani, 2010

Halko, N., P. G. Martinsson, and J. A. Tropp, 2011, Find-ing structure with randomness: Probabilistic algorithms forconstructing approximate matrix

Herrmann, F. J., 2010, Randomized sampling and sparsity:Getting more information from fewer samples: Geophysics,75, WB173–WB187.

References

Herrmann, F. J., and D. Wang, 2008, Seismic wavefield inversion with curvelet-domain sparsity promotion: SEG Technical Program Expanded Abstracts, SEG, 2497–2501.

Lin, T., and F. J. Herrmann, 2009, Unified compressive sensingframework for simultaneous acquisition with primary esti-mation: SEG Technical Program Expanded Abstracts, SEG,3113–3117.

Minato, S., T. Matsuoka, T. Tsuji, D. Draganov, J. Hunziker,and K. Wapenaar, 2011, Seismic interferometry using multidimensional deconvolution and crosscorrelation for crosswell seismic reflection data without borehole sources: Geophysics, 76, SA19–SA34.

van Groenestijn, G. J. A., and D. J. Verschuur, 2009, Estimating primaries by sparse inversion and application to near-offsetdata reconstruction: Geophysics, 74, A23–A28.

Verschuur, D. J., A. J. Berkhout, and C. P. A. Wapenaar, 1992, Adaptive surface-related multiple elimination: Geophysics,

57, 1166–1177.

P.G. Martinsson, 2010 ,A fast randomized algorithm for computing a Hierarchically Semi-Separable representation of a matrix

Acknowledgements

SINBAD

NSERC CRSNG

This work was in part financially supported by the Natural Sciences and Engineering Research Council of Canada Discovery Grant (22R81254) and the Collaborative Research and Development Grant DNOISE II (375142-08). This research was carried out as part of the SINBAD II project with support from the following organizations: BG Group, BP, Chevron, ConocoPhillips, Petrobras, PGP, PGS, Total SA, and WesternGeco.

SLIM

Thank you

slim.eos.ubc.ca