Low-Rank Matrix Recovery for Parallel Architectures

Oscar López and Rajiv Kumar

SLIM
University of British Columbia

Motivation

- Large-scale data processing
  - interpolate missing data
  - source separation

- Exploit *low-rank* structure of seismic data (2D & 3D)
  - *SVD-free* rank penalization techniques

- Need to improve time complexity
  - use efficient optimization schemes
  - design for parallel architectures

Contributions

Decoupling Method

- decompose into independent ℓ_2- minimization problems
- solve in parallel architectures
- parameter free approach

Outline

- Methodology
  - decoupling method
- Numerical Experiments

SLIM

Methodology

want to minimize nuclear norm

$$\min_{\mathbf{X}} \|\mathbf{X}\|_* \;\; \text{s.t.} \;\; \|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_F^2 \leq \sigma$$

where $\|\mathbf{X}\|_* = \sum_{i=1}^{m} \lambda_i = \|\lambda\|_1$

and λ_i are the *singular* values.

Methodology

want to minimize nuclear norm

$$\min_{\mathbf{X}} \|\mathbf{X}\|_* \;\; \text{s.t.} \;\; \|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_F^2 \leq \sigma$$

Assume uniform noise model (evenly distributed through matrix)

where $\|\mathbf{X}\|_* = \sum_{i=1}^{m} \lambda_i = \|\lambda\|_1$

and λ_i are the *singular* values.

SLIM

Nuclear Norm via Factorization

$$\mathbf{X} = \mathbf{L}\mathbf{R}^H$$

Nuclear norm is given as

$$\|\mathbf{X}\|_* = \min_{\mathbf{L}\mathbf{R}^H = \mathbf{X}} \frac{1}{2}(\|\mathbf{L}\|_F^2 + \|\mathbf{R}\|_F^2)$$

where $\|\cdot\|_F^2$ is sum of squares of all entries.

Nuclear Norm Minimization- Factorized Form

Choosing $r \ll \min(m,n)$, we now solve

$$\min_{\mathbf{L}\in\mathbb{R}^{n\times r},\mathbf{R}\in\mathbb{R}^{m\times r}} \frac{1}{2}(\|\mathbf{L}\|_F^2 + \|\mathbf{R}\|_F^2) \quad \text{s.t.} \quad \|\mathcal{A}(\mathbf{L}\mathbf{R}^H) - \mathbf{b}\|_F^2 \leq \sigma$$

SPG-LR implementation: **Aravkin, Kumar, Mansour, Recent and Herrmann. "Fast Methods For Denoising Matrix Completion Formulations, With Applications To Robust Seismic Data Interpolation". SIAM 2014**

(more details of this approach in Rajiv's talk)

Nuclear Norm Minimization- Factorized Form

Choosing $r \ll \min(m,n)$, we now solve

$$\min_{\mathbf{L}\in\mathbb{R}^{n\times r},\mathbf{R}\in\mathbb{R}^{m\times r}} \underbrace{\frac{1}{2}(\|\mathbf{L}\|_F^2 + \|\mathbf{R}\|_F^2)}_{\text{bi-convex function}} \quad \text{s.t.} \quad \|\mathcal{A}(\mathbf{L}\mathbf{R}^H) - \mathbf{b}\|_F^2 \le \sigma$$

Alternating approach: optimize over a single factor at a time

Kumar, Lopez, Davis, Aravkin and Herrmann. "Beating level set methods for 3D seismic data interpolation: a primal-dual alternating approach". IEEE 2016

Kumar, Lopez, Davis, Aravkin and Herrmann. "Beating level set methods for 3D seismic data interpolation: a primal-dual alternating approach". IEEE 2016

Alternating Nuclear Norm Minimization

1. Input: $\mathcal{A}, \mathbf{b}$
2. Initialize: $\mathbf{L}^0$
3. for $t = 0, ..., T-1$ do
4. $$\mathbf{R}^{t+1} = \min_{\mathbf{R} \in \mathbb{R}^{m \times r}} \frac{1}{2} ||\mathbf{R}||_F^2 \quad \text{s.t.} \quad ||\mathcal{A}(\mathbf{L}^t \mathbf{R}^H) - \mathbf{b}||_F^2 \leq \sigma_t$$
5. $$\mathbf{L}^{t+1} = \min_{\mathbf{L} \in \mathbb{R}^{n \times r}} \frac{1}{2} ||\mathbf{L}||_F^2 \quad \text{s.t.} \quad ||\mathcal{A}(\mathbf{L}(\mathbf{R}^{t+1})^H) - \mathbf{b}||_F^2 \leq \sigma_t$$
6. end for
7. Return $\tilde{\mathbf{X}} = \mathbf{L}^T (\mathbf{R}^T)^H$

Kumar, Lopez, Davis, Aravkin and Herrmann. "Beating level set methods for 3D seismic data interpolation: a primal-dual alternating approach". IEEE 2016

Alternating Nuclear Norm Minimization

1. Input: $\mathcal{A}, \mathbf{b}$
2. Initialize: $\mathbf{L}^0$
3. for $t = 0, ..., T-1$ do
4. $$\mathbf{R}^{t+1} = \min_{\mathbf{R} \in \mathbb{R}^{m \times r}} \frac{1}{2}||\mathbf{R}||_F^2 \quad \text{s.t.} \quad ||\mathcal{A}(\mathbf{L}^t \mathbf{R}^H) - \mathbf{b}||_F^2 \leq \sigma_t$$
5. $$\mathbf{L}^{t+1} = \min_{\mathbf{L} \in \mathbb{R}^{n \times r}} \frac{1}{2}||\mathbf{L}||_F^2 \quad \text{s.t.} \quad ||\mathcal{A}(\mathbf{L}(\mathbf{R}^{t+1})^H) - \mathbf{b}||_F^2 \leq \sigma_t$$
6. end for
7. Return $\tilde{\mathbf{X}} = \mathbf{L}^T (\mathbf{R}^T)^H$

Each sub problem is convex

Decoupling Method

$$\mathbf{R}^{t+1} = \underset{\mathbf{R}\in\mathbb{R}^{m\times r}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{R}||_F^2 \quad \text{s.t. } ||\mathcal{A}(\mathbf{L}^t\mathbf{R}^H) - \mathbf{b}||_F^2 \leq \sigma_t$$

Can further decouple each convex sub problem to solve "row-by-row".

Decoupling Method: Matrix Completion

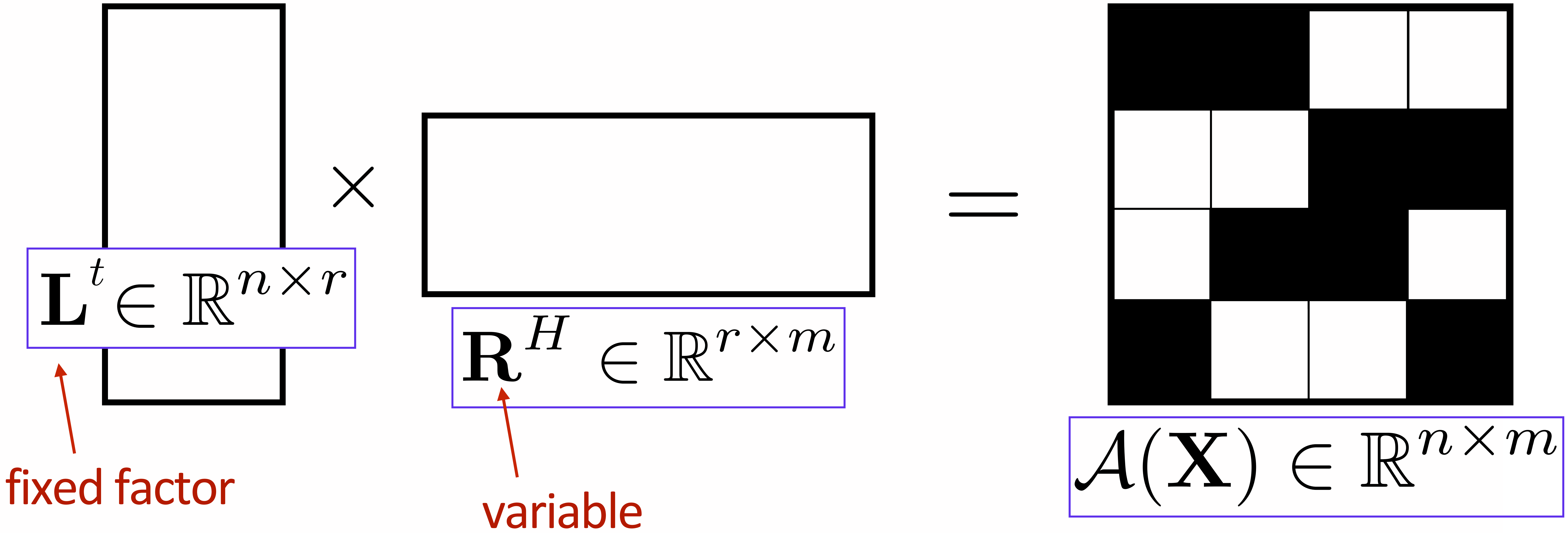
$$\mathbf{R}^{t+1} = \underset{\mathbf{R}\in\mathbb{R}^{m\times r}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{R}||_F^2 \quad \text{s.t. } ||\mathcal{A}(\mathbf{L}^t\mathbf{R}^H) - \mathbf{b}||_F^2 \leq \sigma_t$$

Example: Matrix Completion for data interpolation

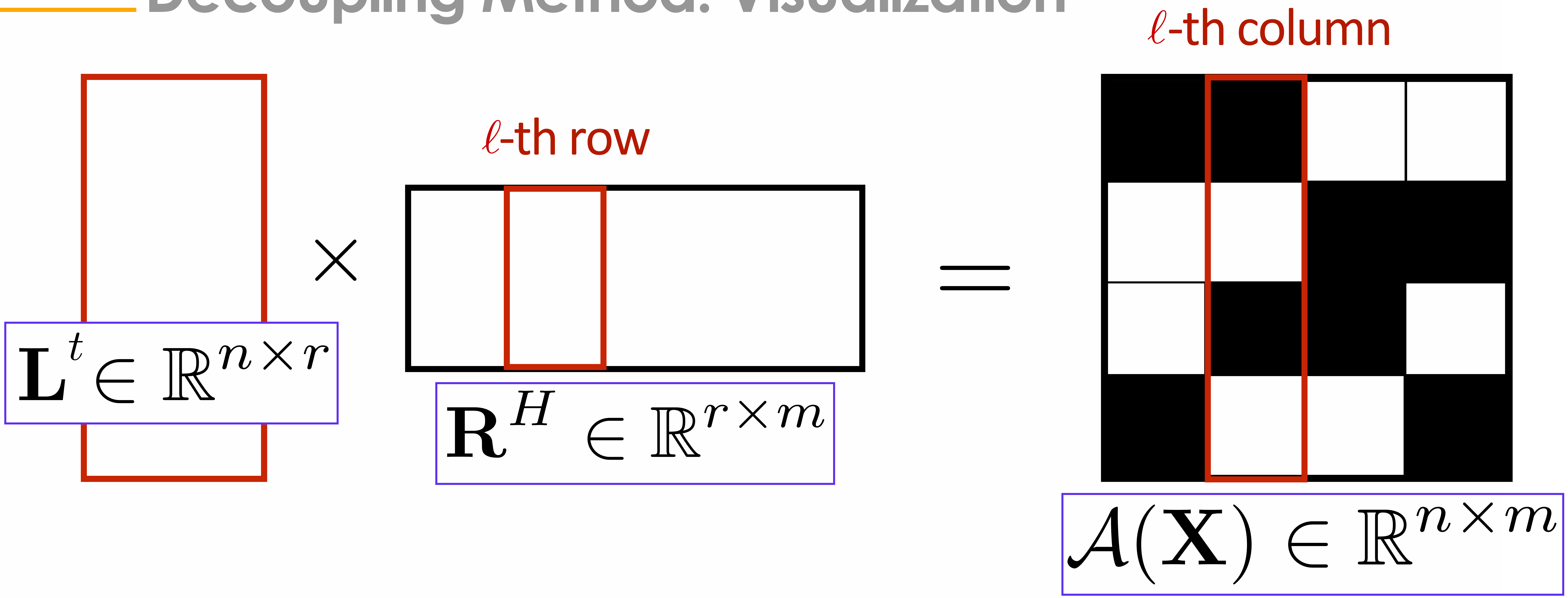
$$\mathcal{A}(\mathbf{X})_{i,j} = \begin{cases} \mathbf{X}_{i,j} & \text{if } (i,j)\in\Omega \\ 0 & \text{otherwise} \end{cases}$$

where Ω is the set of observed matrix entries.

Decoupling Method: Visualization



Decoupling Method: Visualization



$$\mathbf{R}^{t+1}(\ell, :) = \arg\min_{v \in \mathbb{R}^r} \|v\|_2 \ \text{ s.t. } \ \|\mathcal{A}_\ell(\mathbf{L}^t v) - \mathbf{b}(:, \ell)\|_2^2 \leq \frac{\sigma_t}{m}$$

SLIM

Decoupling Method: Matrix Completion

So we can solve for rows independently: $1 \leq \ell \leq m$

$$\mathbf{R}^{t+1}(\ell,:) = \arg\min_{v \in \mathbb{R}^r} \|v\|_2 \;\; \text{s.t.} \;\; \|\mathcal{A}_\ell(\mathbf{L}^t v) - \mathbf{b}(:,\ell)\|_2^2 \leq \frac{\sigma_t}{m}$$

where $\mathcal{A}_\ell$ is the action of $\mathcal{A}$ on the ℓ-th column.

Decoupling Method: Matrix Completion

$$\mathbf{R}^{t+1}(\ell,:) = \arg\min_{v\in\mathbb{R}^r} \|v\|_2 \;\; \text{s.t.} \;\; \|\mathcal{A}_\ell(\mathbf{L}^t v) - \mathbf{b}(:,\ell)\|_2^2 \leq \frac{\sigma_t}{m}$$

Many methods to solve residual constrained ℓ_2- minimization:

- SPG-ℓ_1 (Pareto curve approach)
- primal-dual splitting (for blocks of rows)
- Matlab’s backslash (QR factorization)

Decoupling Method: Matrix Completion

$$\mathbf{R}^{t+1}(\ell,:) = \arg\min_{v \in \mathbb{R}^r} \|v\|_2 \;\; \text{s.t.} \;\; \|\mathcal{A}_\ell(\mathbf{L}^t v) - \mathbf{b}(:,\ell)\|_2^2 \leq \frac{\sigma_t}{m}$$

Many methods to solve residual constrained ℓ_2- minimization:

- SPG-ℓ_1 (Pareto curve approach)
- primal-dual splitting (for blocks of rows)
- Matlab's backslash

optimized by Matlab, requires no parameters!

Pseudo Code: solve for $\mathbf{R}^{t+1}(\ell,:)$

1.Input: $\mathbf{L}^t, \Omega_\ell, \mathbf{b}(:,\ell)$

observed indices of ℓ-th column

observations of ℓ-th column

Pseudo Code: solve for $\mathbf{R}^{t+1}(\ell,:)$

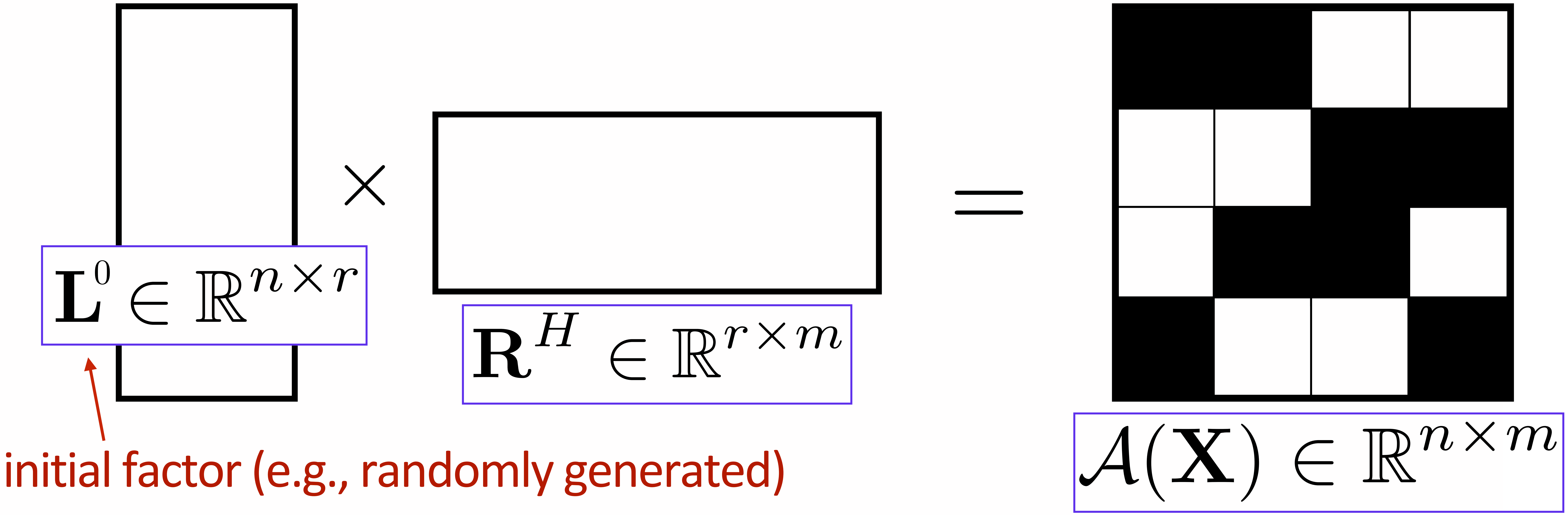
$$1.\text{Input: } \mathbf{L}^t, \Omega_\ell, \mathbf{b}(:,\ell)$$

$$2.\ \tilde{\mathbf{L}} = \mathbf{L}^t_{\Omega_\ell}$$ restrict rows of fixed factor according to Ω_ℓ

$$3.\ \tilde{\mathbf{b}} = \mathbf{b}(\Omega_\ell,\ell)$$ restrict observations according to Ω_ℓ

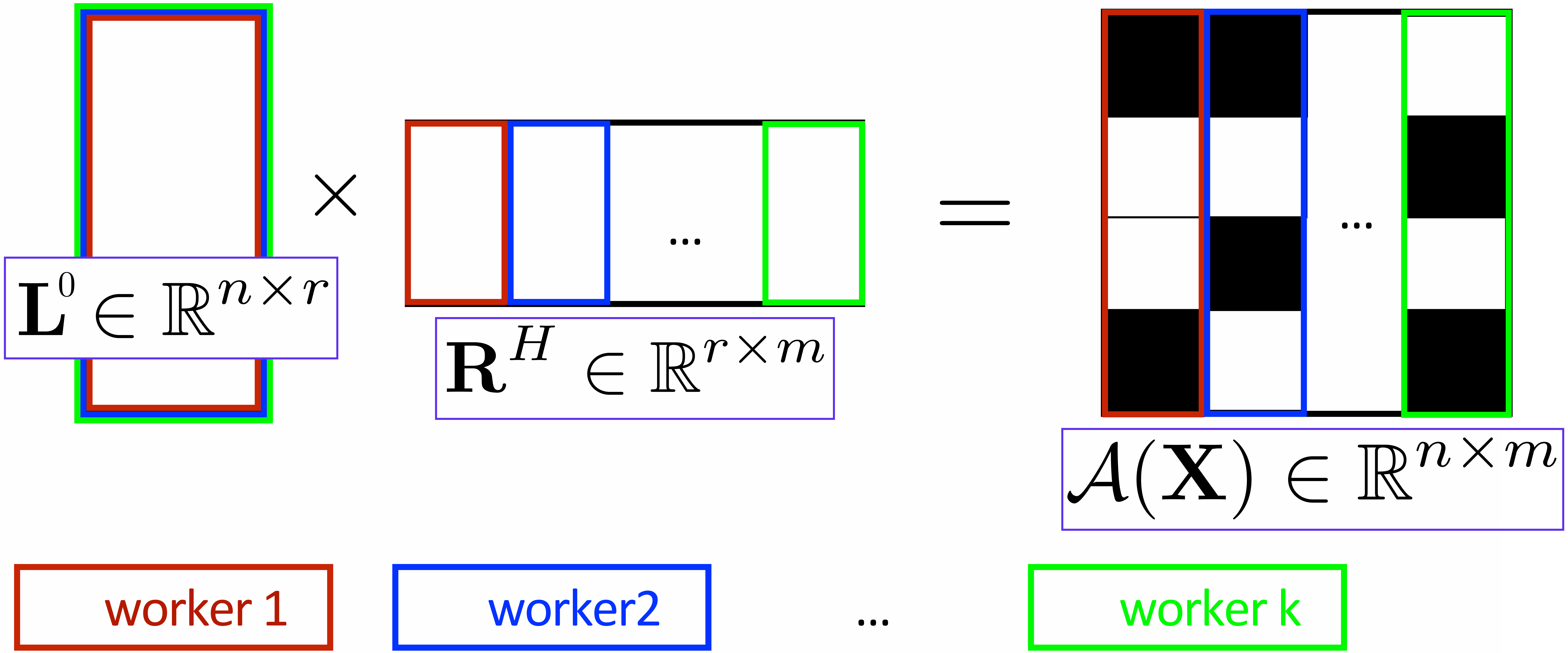
$$4.\ \mathbf{R}^{t+1}(\ell,:) = \tilde{\mathbf{L}} \backslash \tilde{\mathbf{b}}$$ QR factorization

Decoupling Method in Action

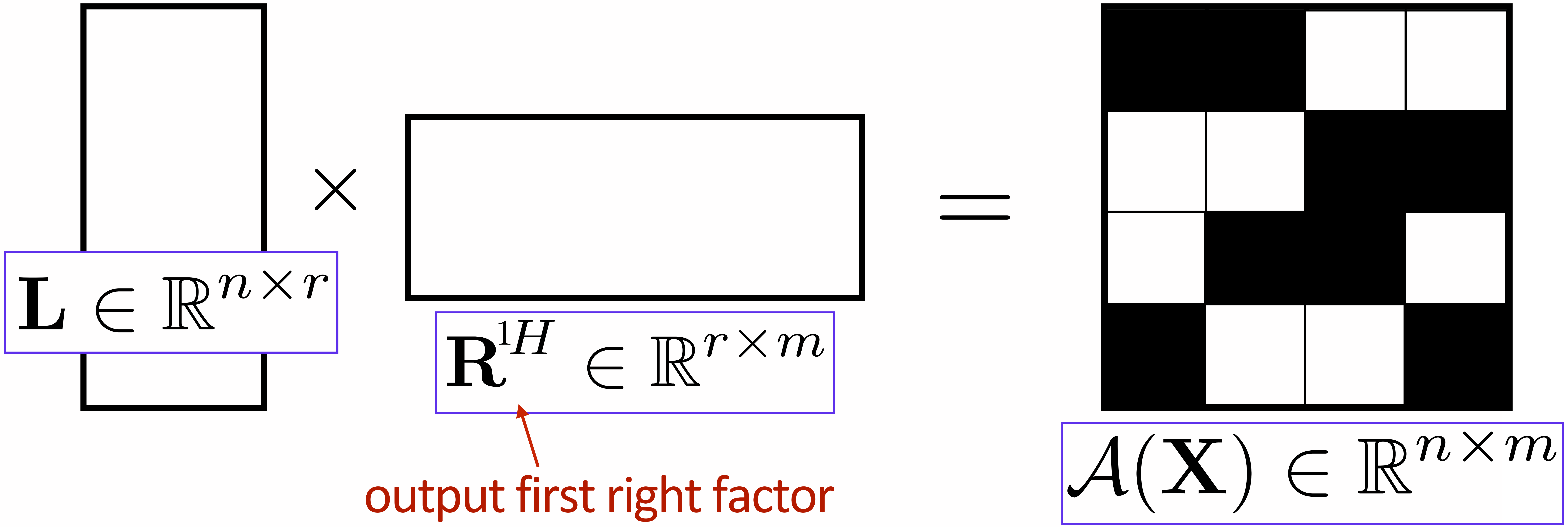


SLIM

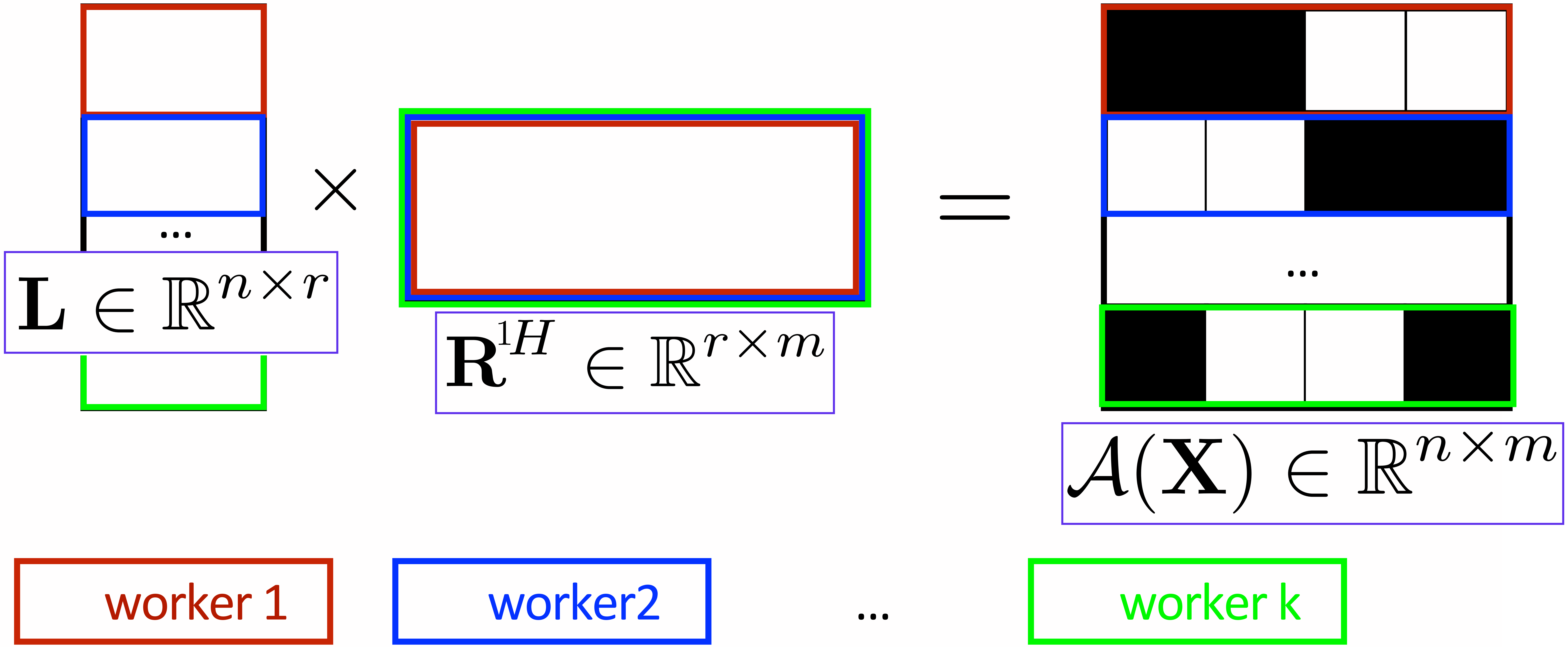
Decoupling Method in Action



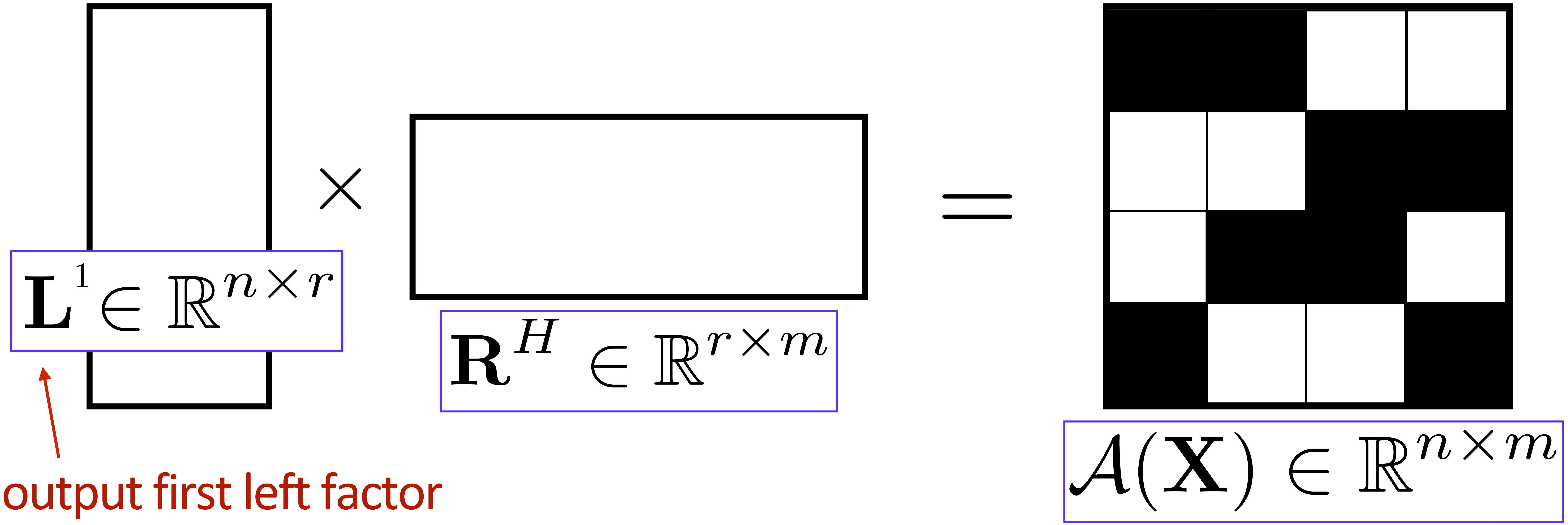
Decoupling Method in Action



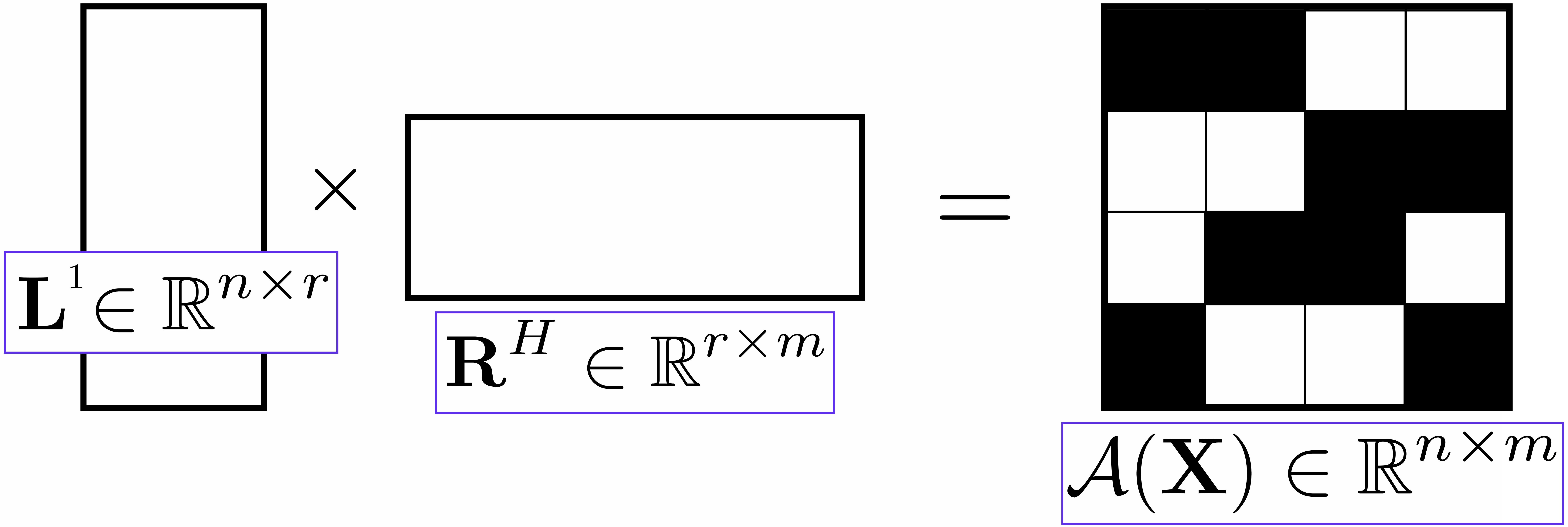
Decoupling Method in Action



Decoupling Method in Action



Decoupling Method in Action



...and so on solve for $(\mathbf{L}^2, \mathbf{R}^2),\ (\mathbf{L}^3, \mathbf{R}^3),\ \dots\ ,\ (\mathbf{L}^T, \mathbf{R}^T)$

Decoupling Method

- Parallelizable: assign set of rows to each worker.
- Similar approaches can be designed for other $\mathcal{A}$ (see Rajiv's talk).
- Can efficiently solve each sub problem.

Outline

- Methodology
  - decoupling method

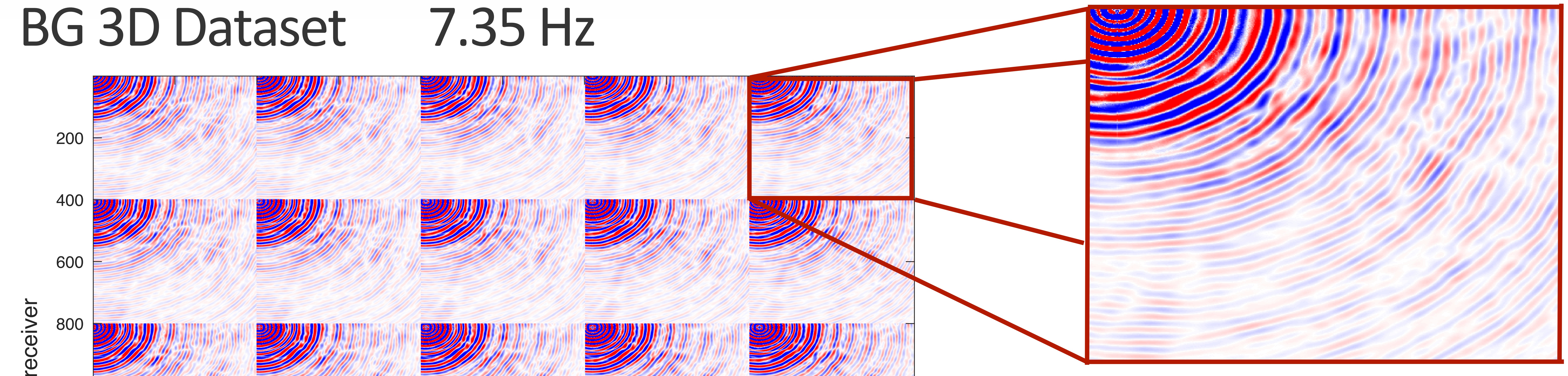
- Numerical Experiments

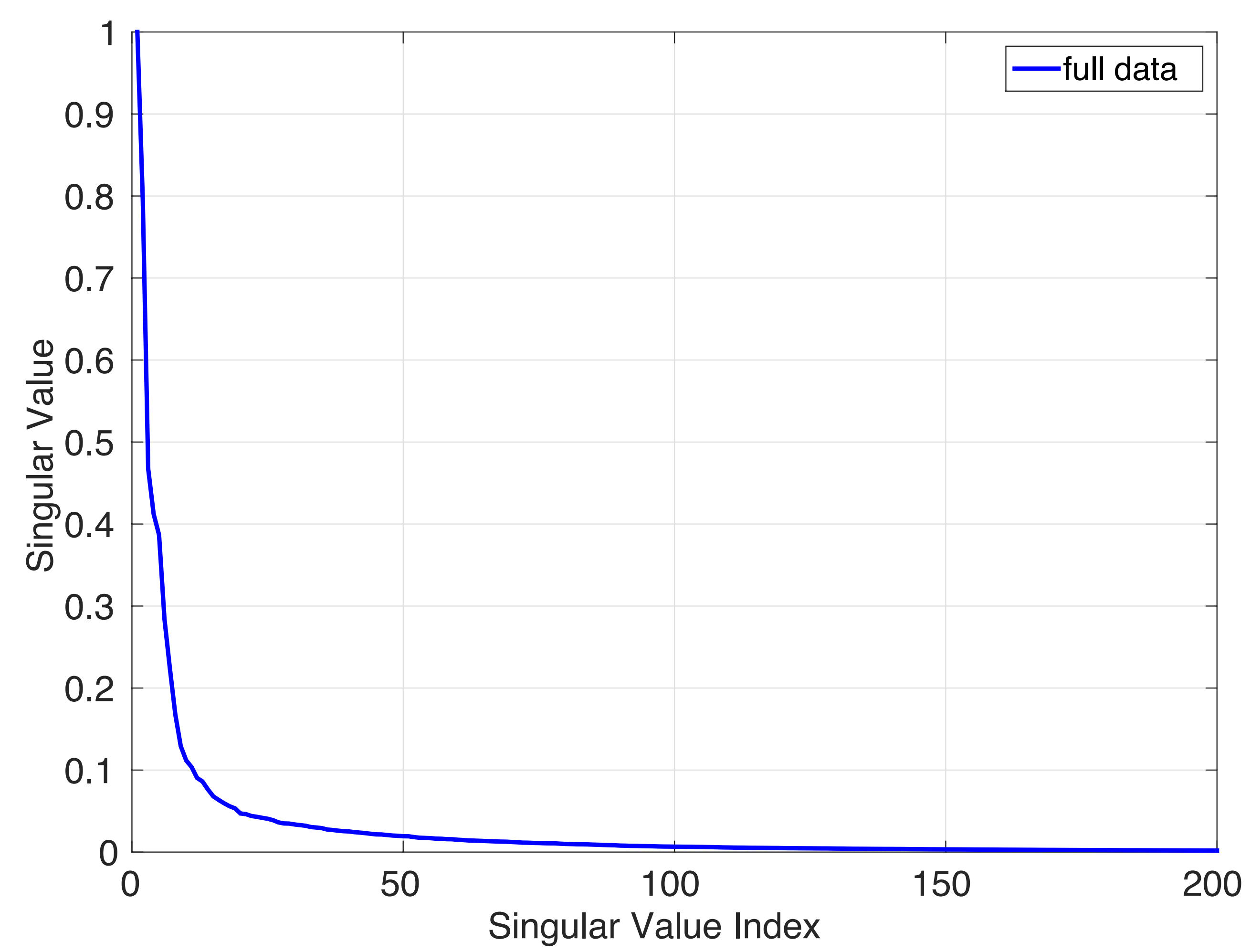
Interpolation: Synthetic BG 3D Model

- 67 x 67 sources with 401 x 401 receivers
- Data at 7.34 Hz and 12.3 Hz.
- Matricize in “(rec,rec)”-form

Data Matricized - (rec,rec) form

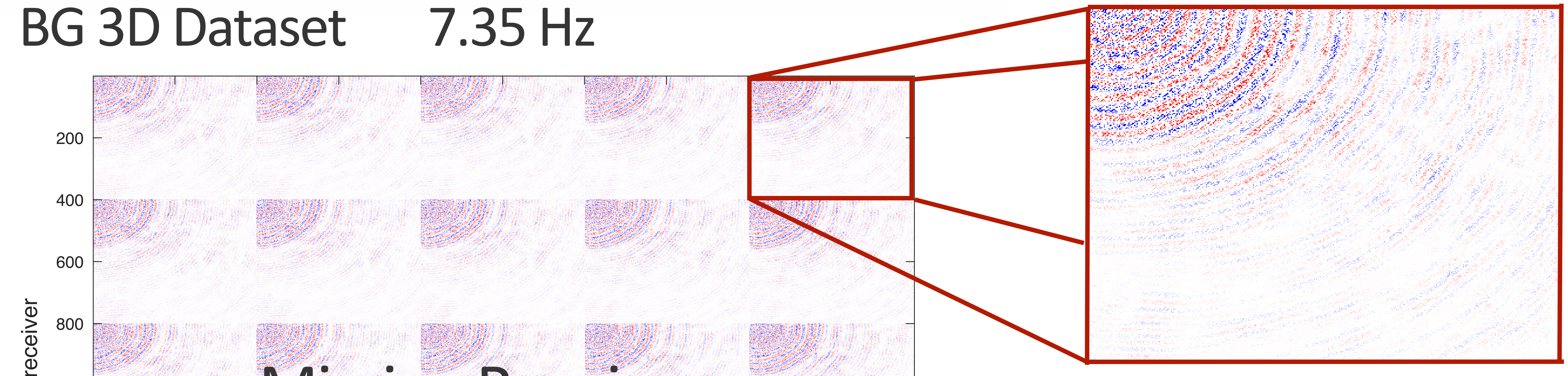
BG 3D Dataset   7.35 Hz

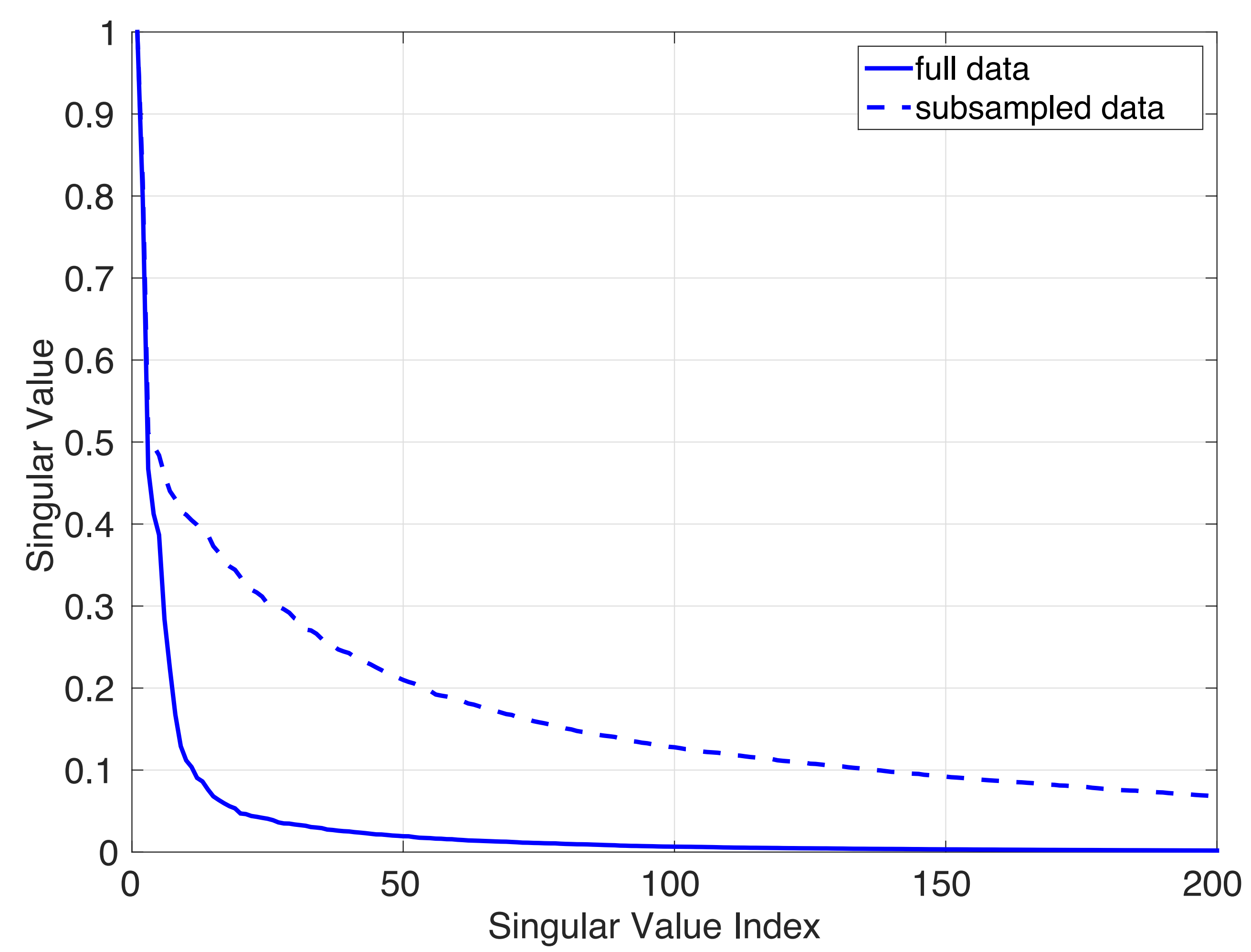




Data Matricized - (rec,rec) form

BG 3D Dataset 7.35 Hz

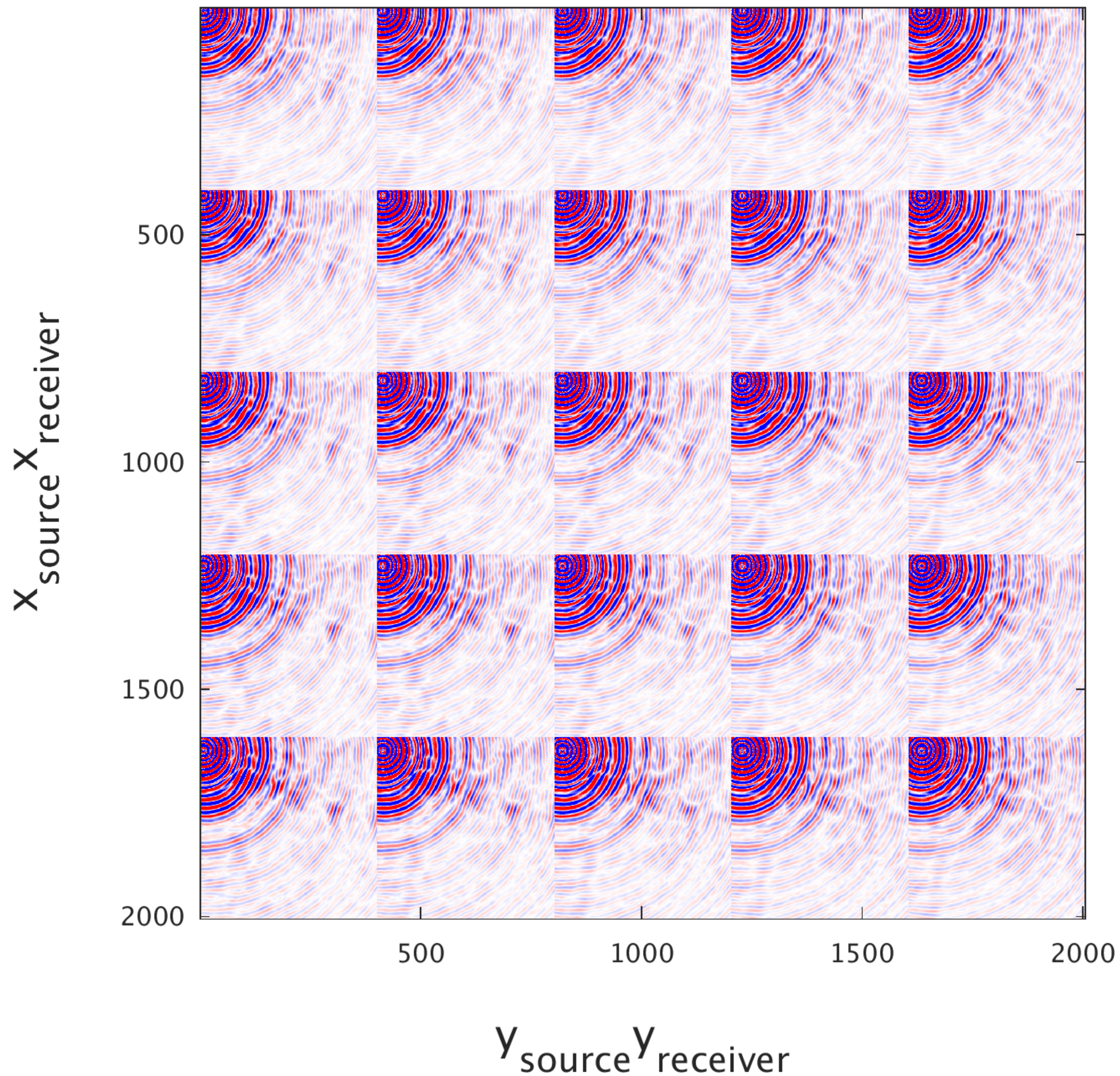




3D Interpolation Experiment

BG 3D Dataset

7.35 Hz



Size: 26,867 x 26,867

(full slice, no windowing)

Remove 80 % of Receivers randomly

Compare Interpolation via:

- SPG-LR
- Decoupling Method

How to choose the rank parameter?

Typical abridged result from low-rank matrix recovery theory:

If $\mathcal{A} : \mathbb{C}^{n\times m} \mapsto \mathbb{C}^k$ is a random linear operator (e.g., Ω chosen randomly, subgaussian), then can recover a rank-r matrix via nuclear norm minimization if

$$k \geq Cr\max(n,m)\log(\max(n,m))$$

with high probability.

SLIM

How to choose the rank parameter?

$$k \geq Cr\max(n,m)\log(\max(n,m))$$

In our case: $k = .2 \cdot nm$

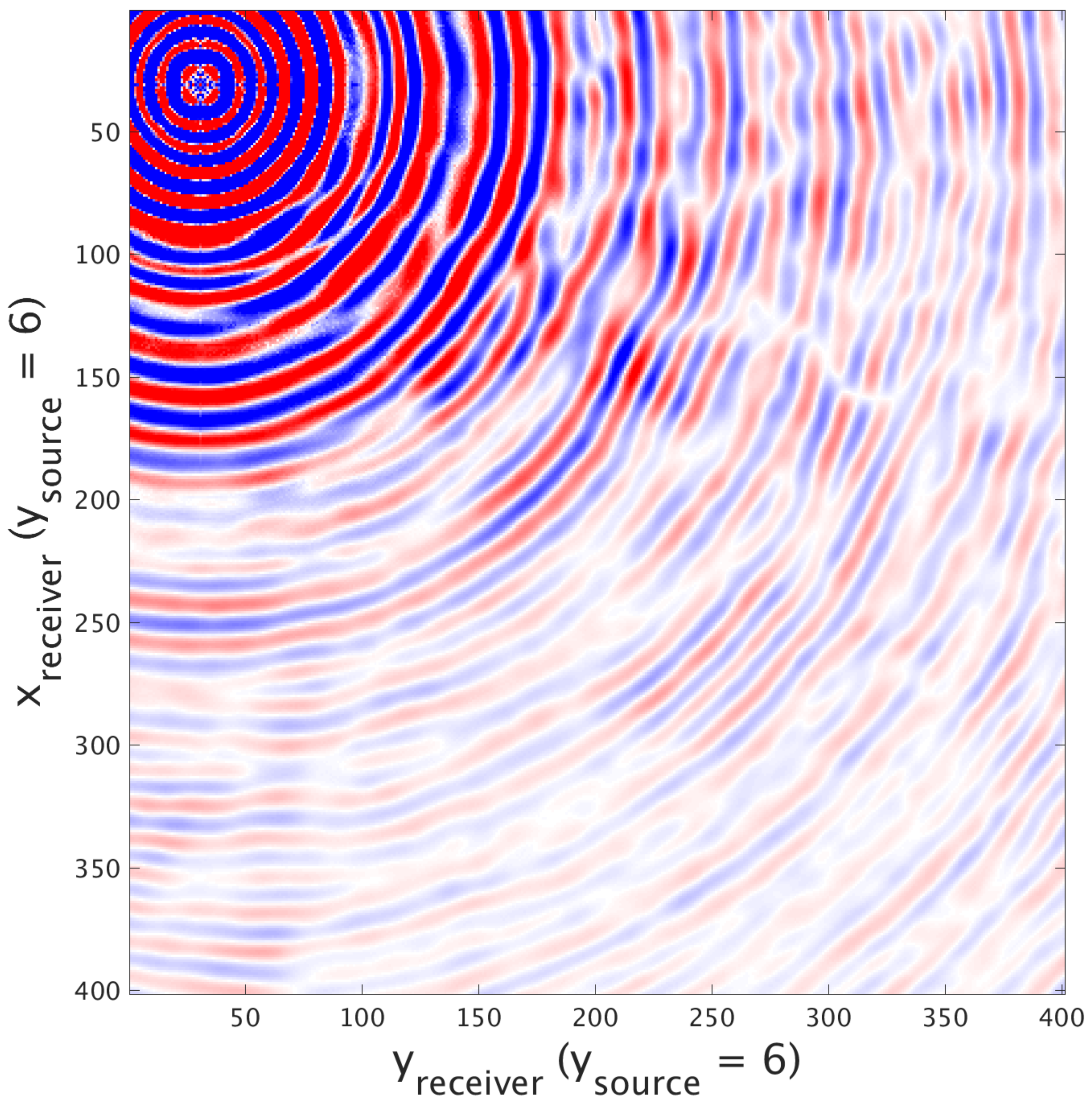
$n = m = 26,867$

(with $C = 1$ and rounding) $\Longrightarrow$ $r \leq 527$

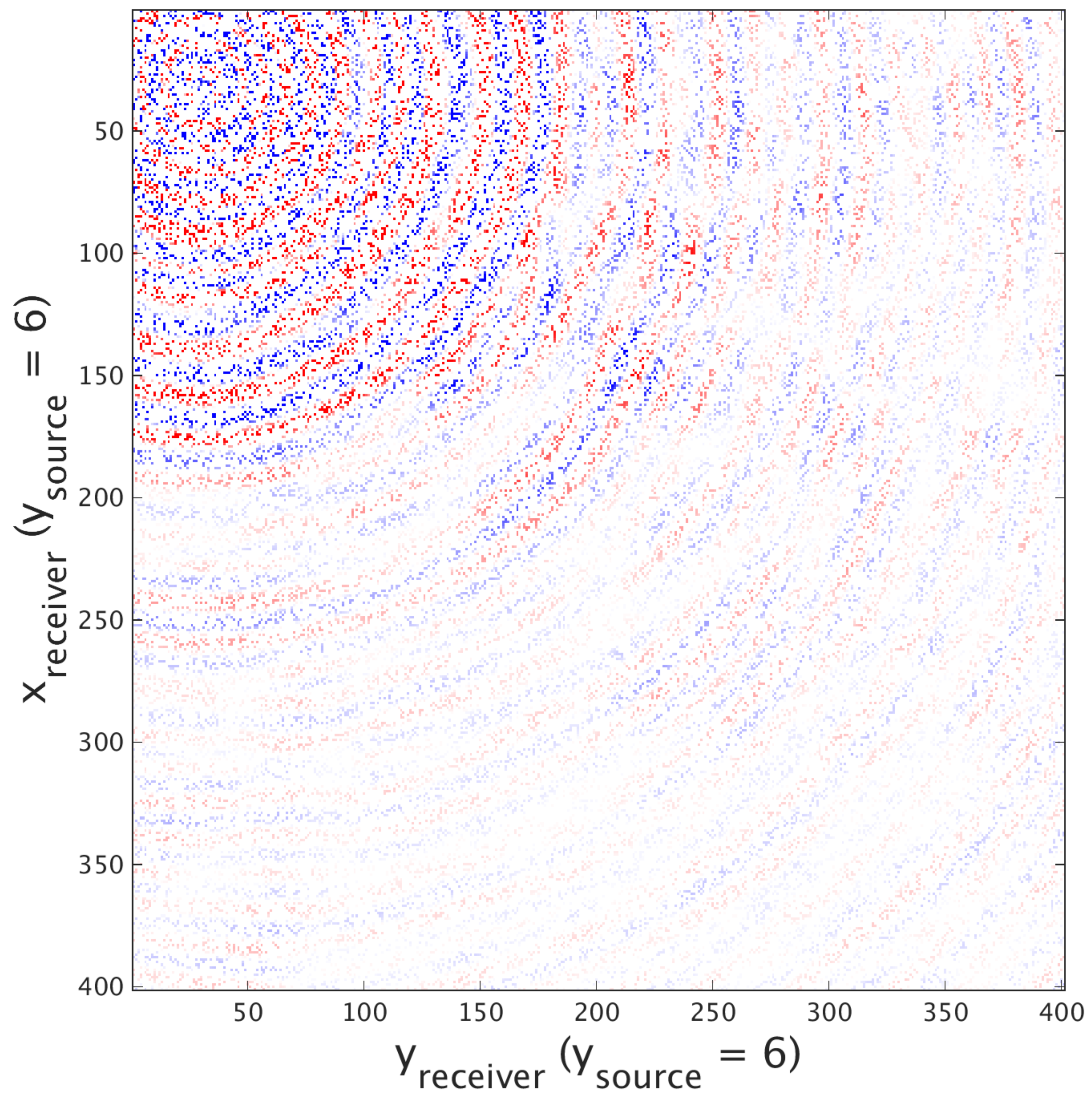
choose upper bound as rank.

Common Source Gather

True Source Gather



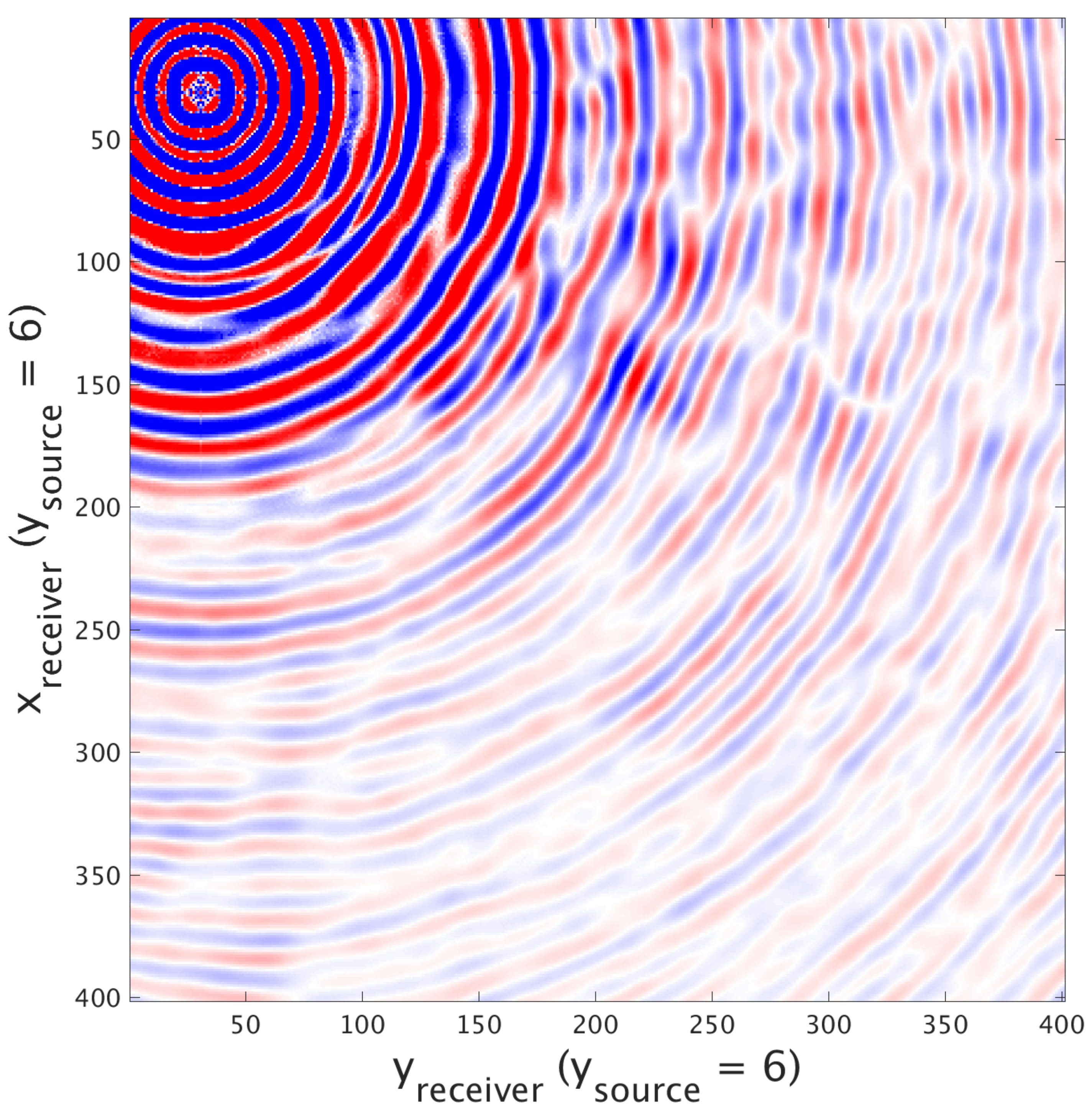
Subsampled Source Gather



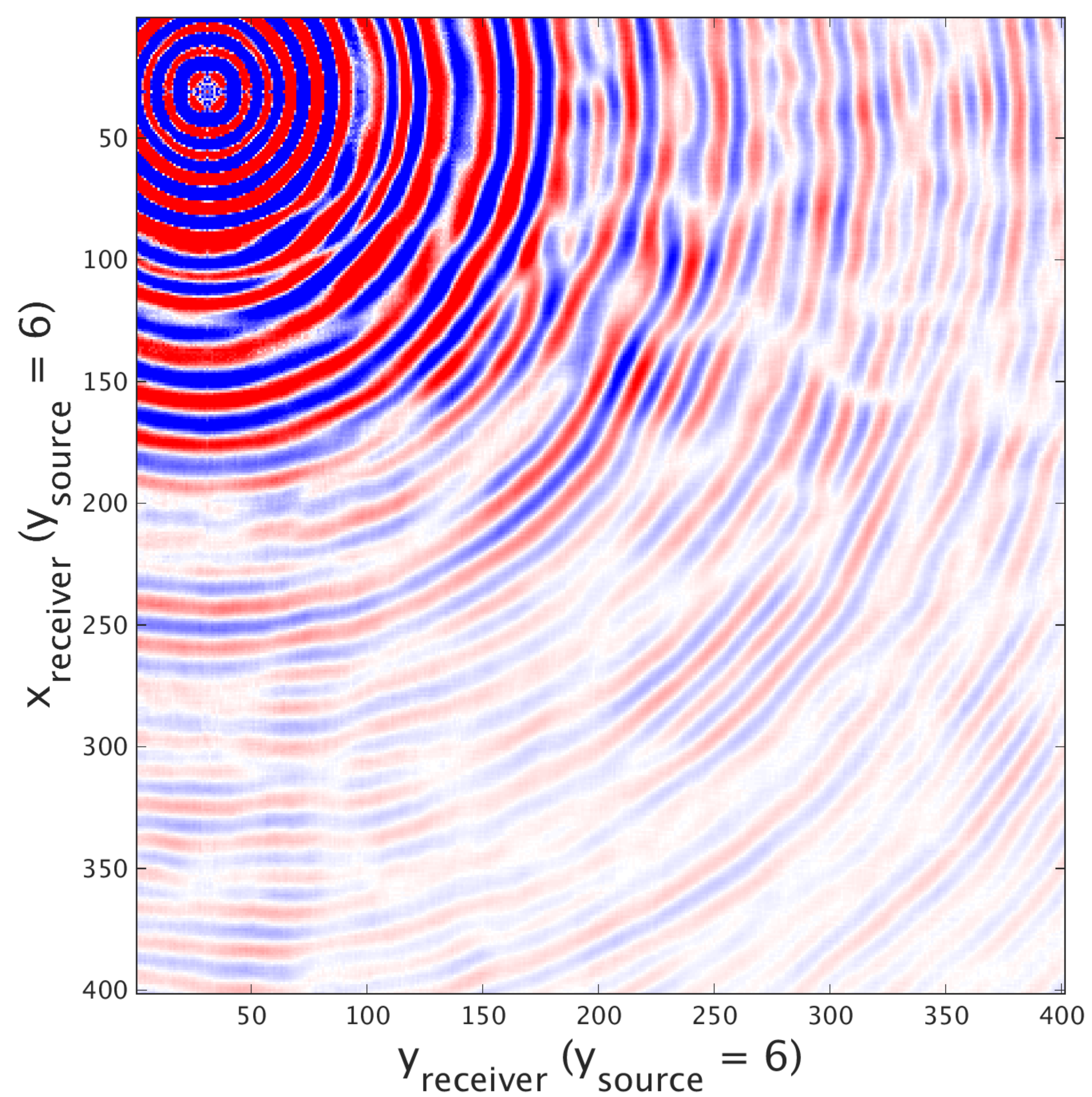
Remove 80 % of Receivers randomly

Results: SPG-LR

True Source Gather



Recovered Source Gather



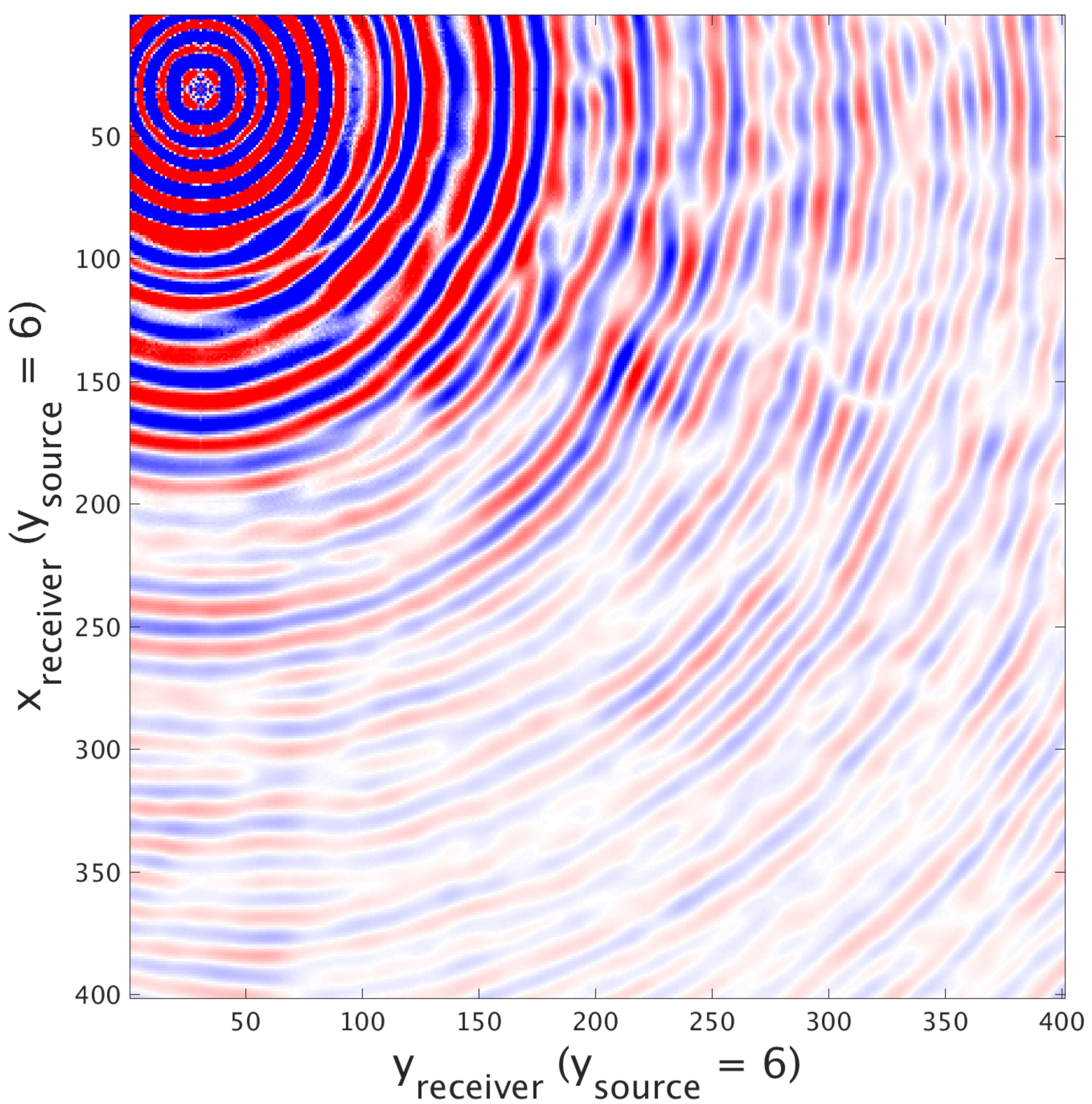
SPG-ℓ_1 iterations: 400

SNR = 26.1 dB

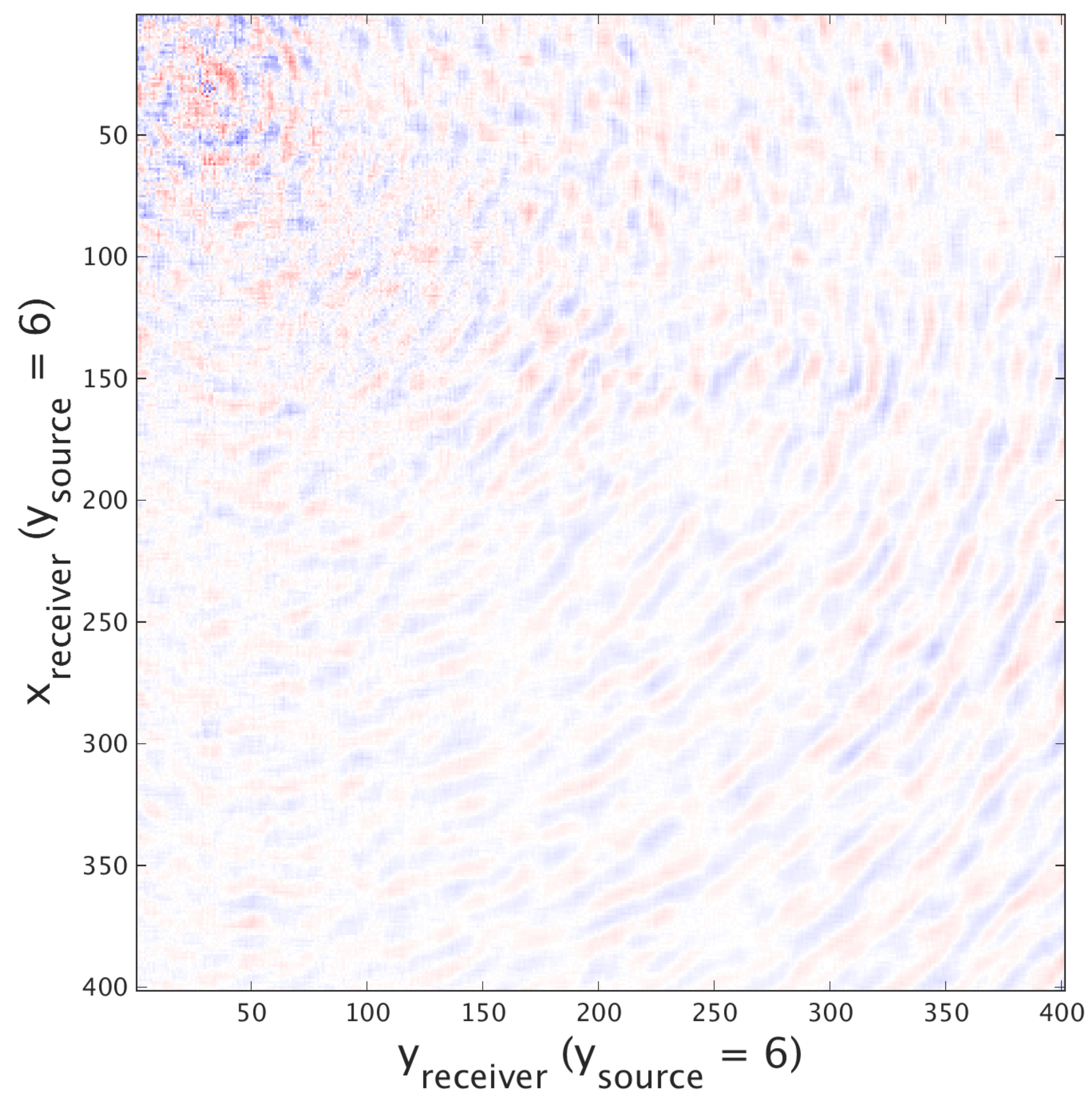
Time = 82 hrs and 40 min

Results: SPG-LR

True Source Gather



Difference Plot



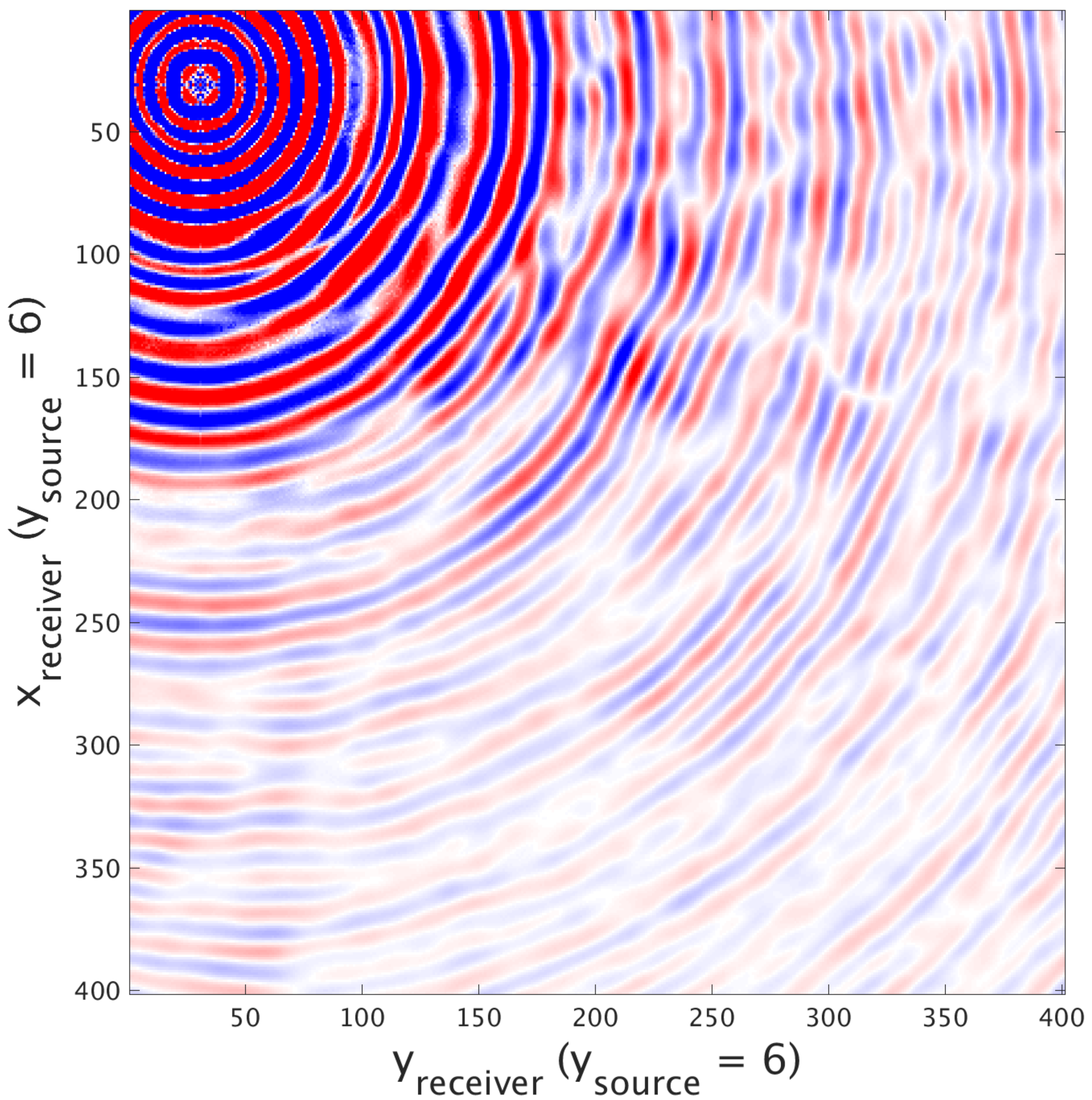
SPG-ℓ_1 iterations: 400

SNR = 26.1 dB

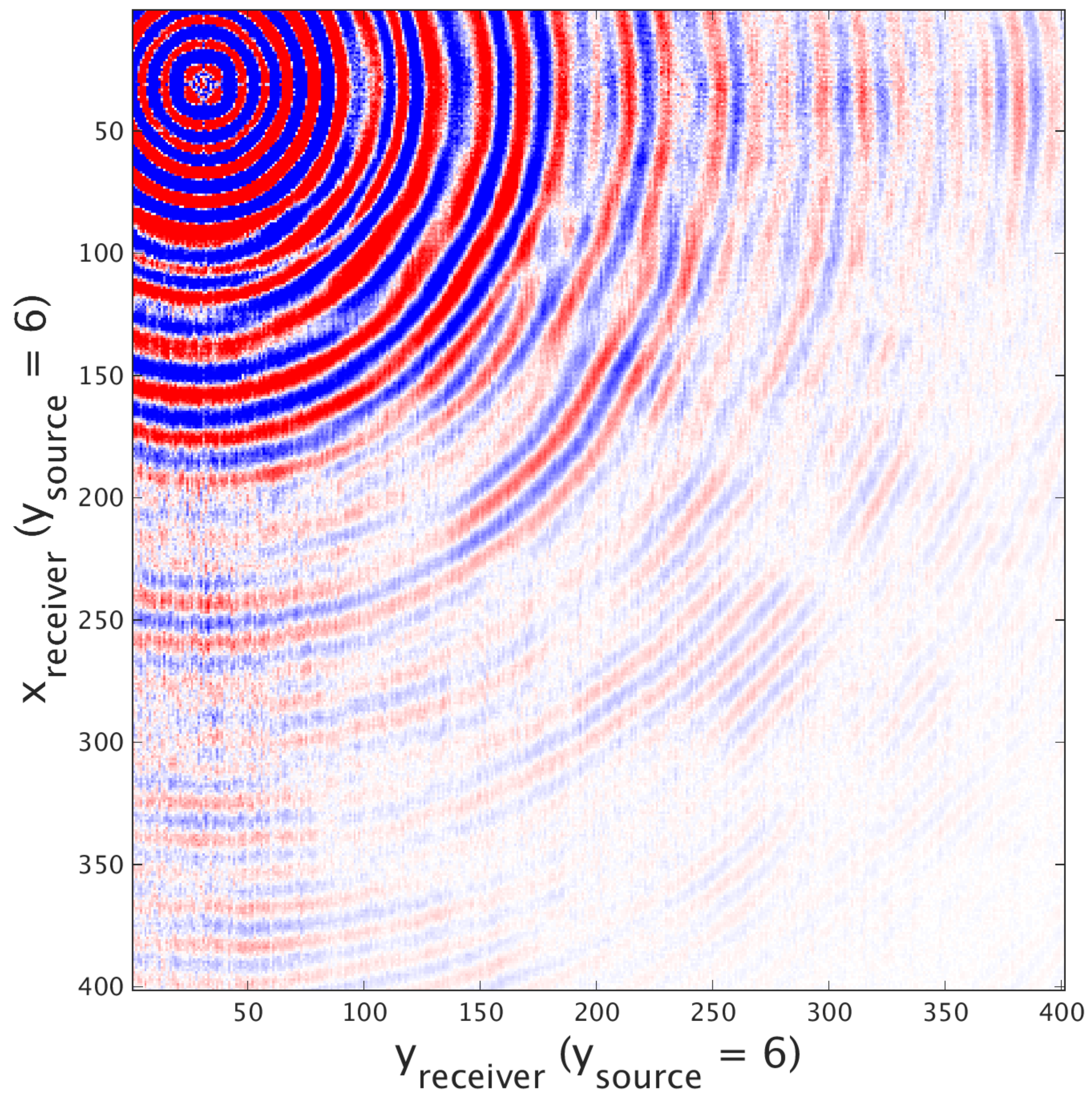
Time = 82 hrs and 40 min

Results: Decoupling Method

True Source Gather



Recovered Source Gather



40 processors

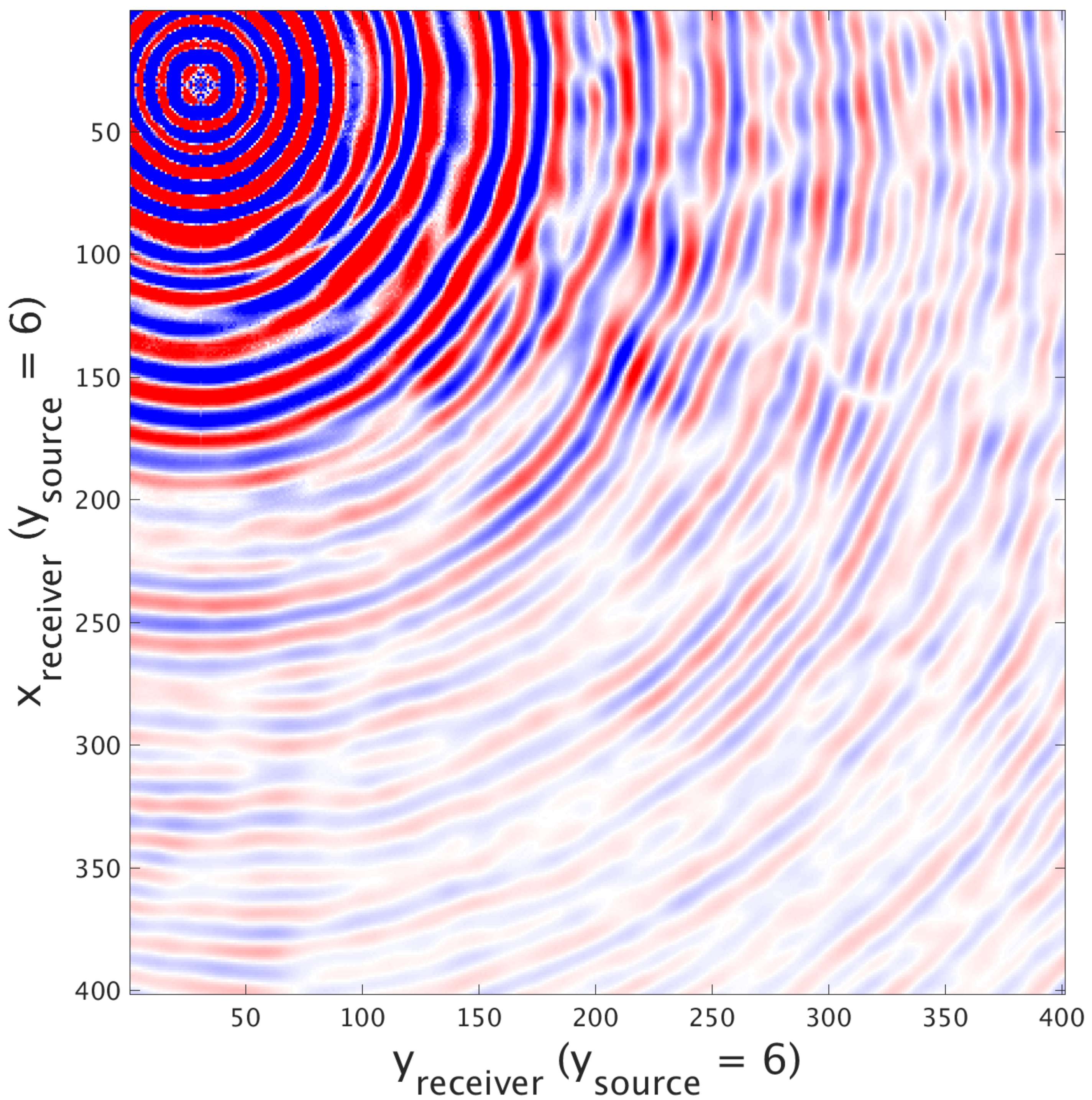
Alternations: 2

SNR = 16 dB

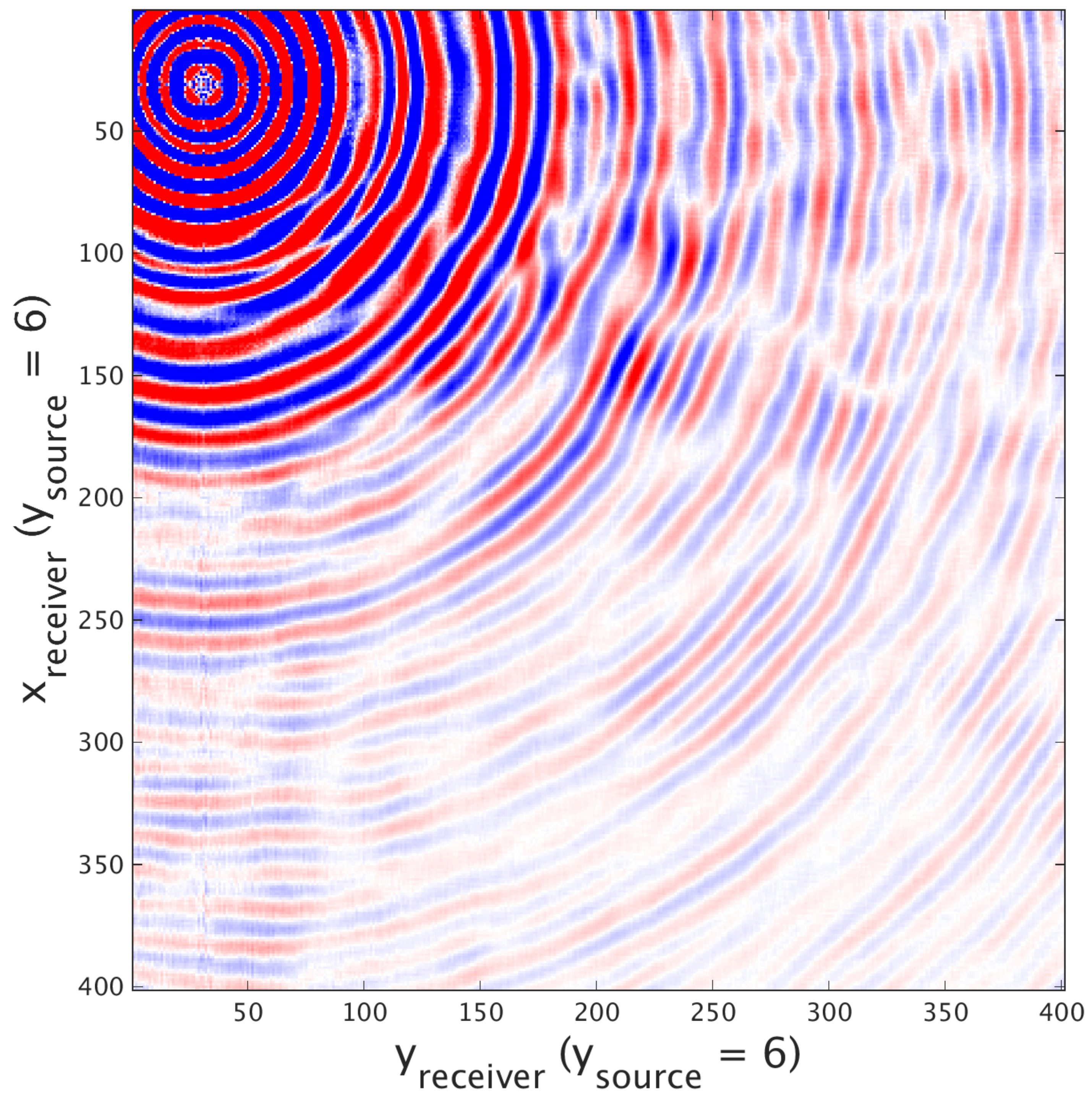
Time = 1 hrs and 28 mins

Results: Decoupling Method

True Source Gather



Recovered Source Gather



40 processors

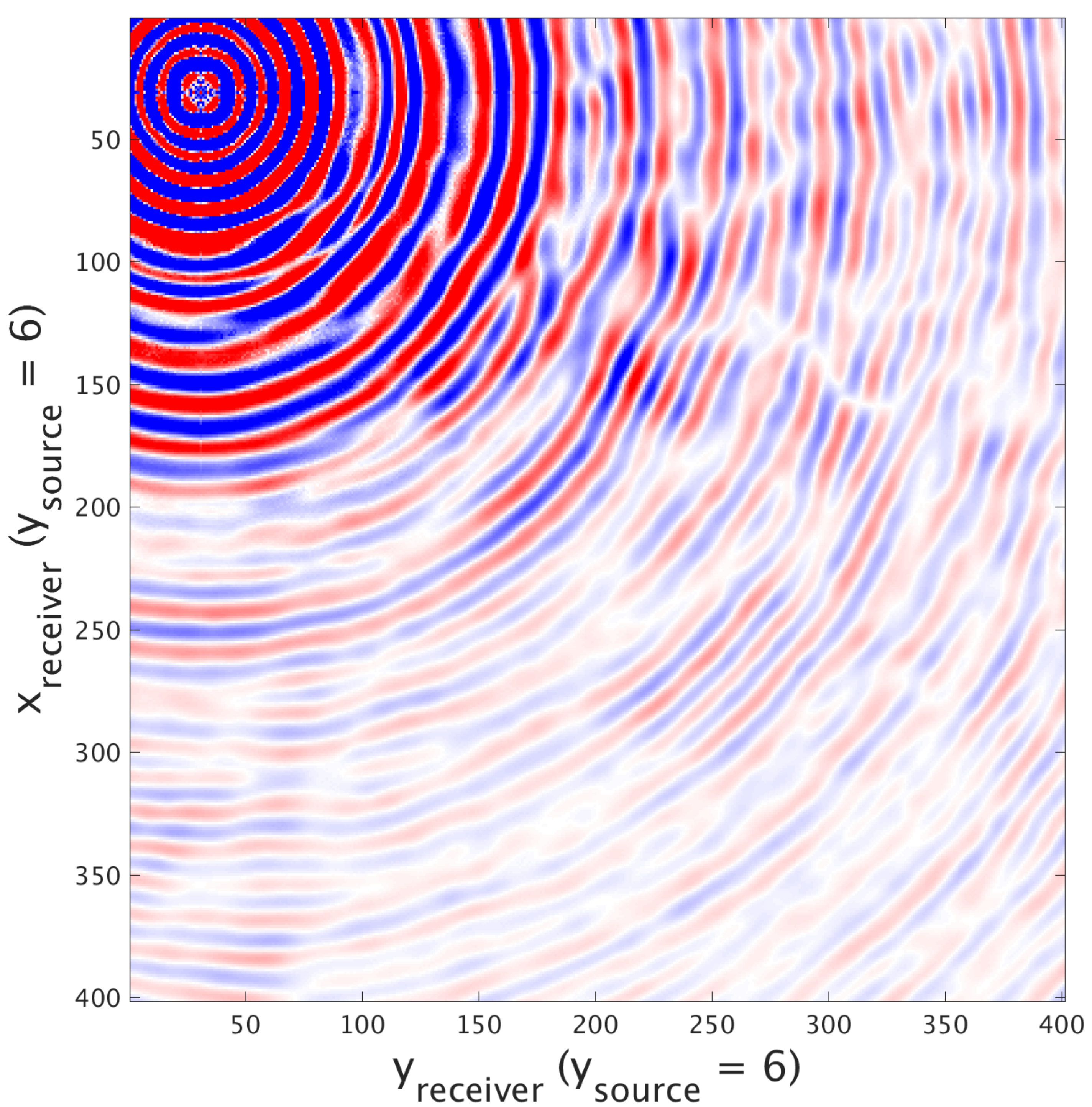
Alternations: 5

SNR = 24.3 dB

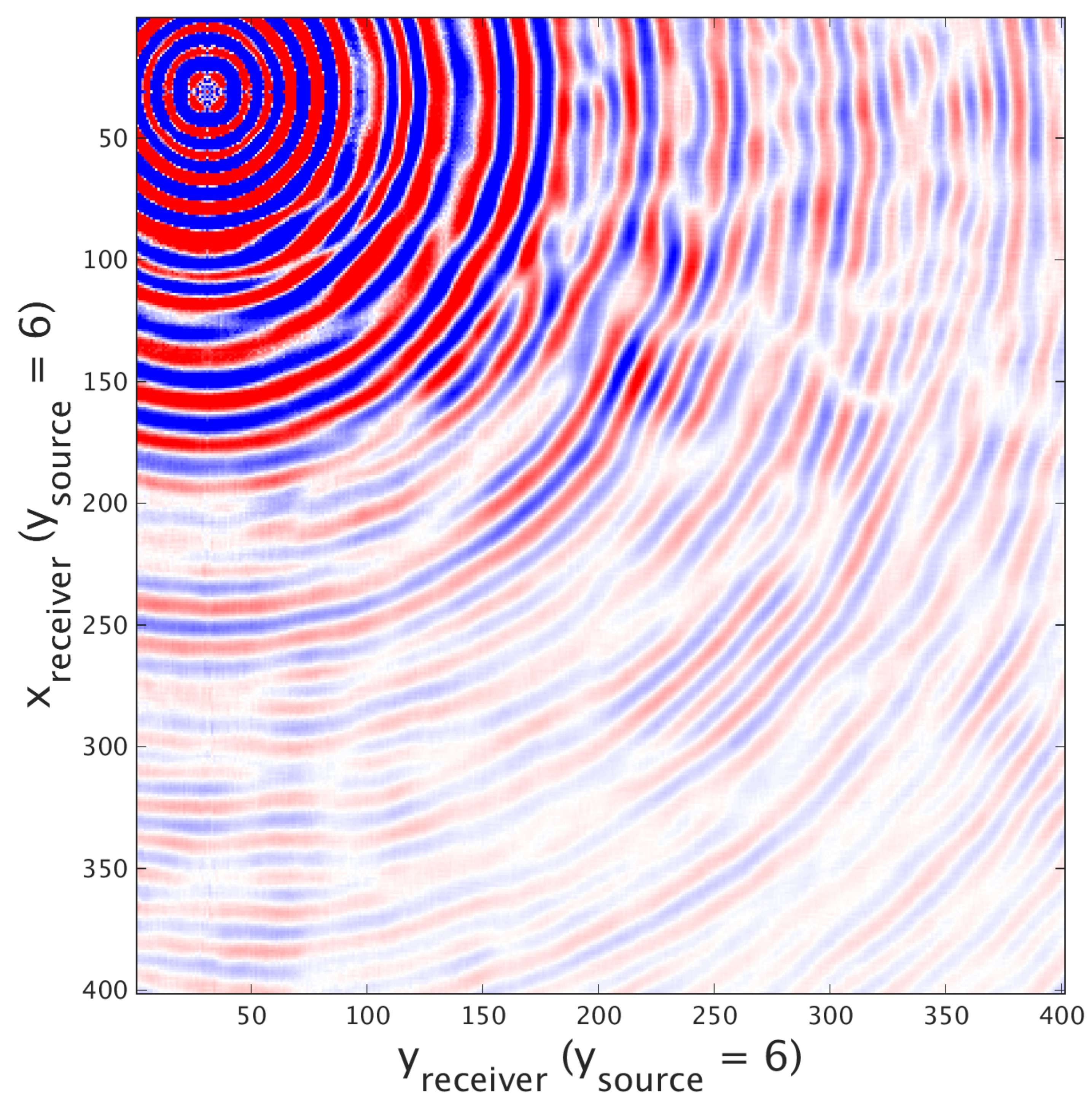
Time = 3 hrs and 47 mins

Results: Decoupling Method

True Source Gather



Recovered Source Gather



40 processors

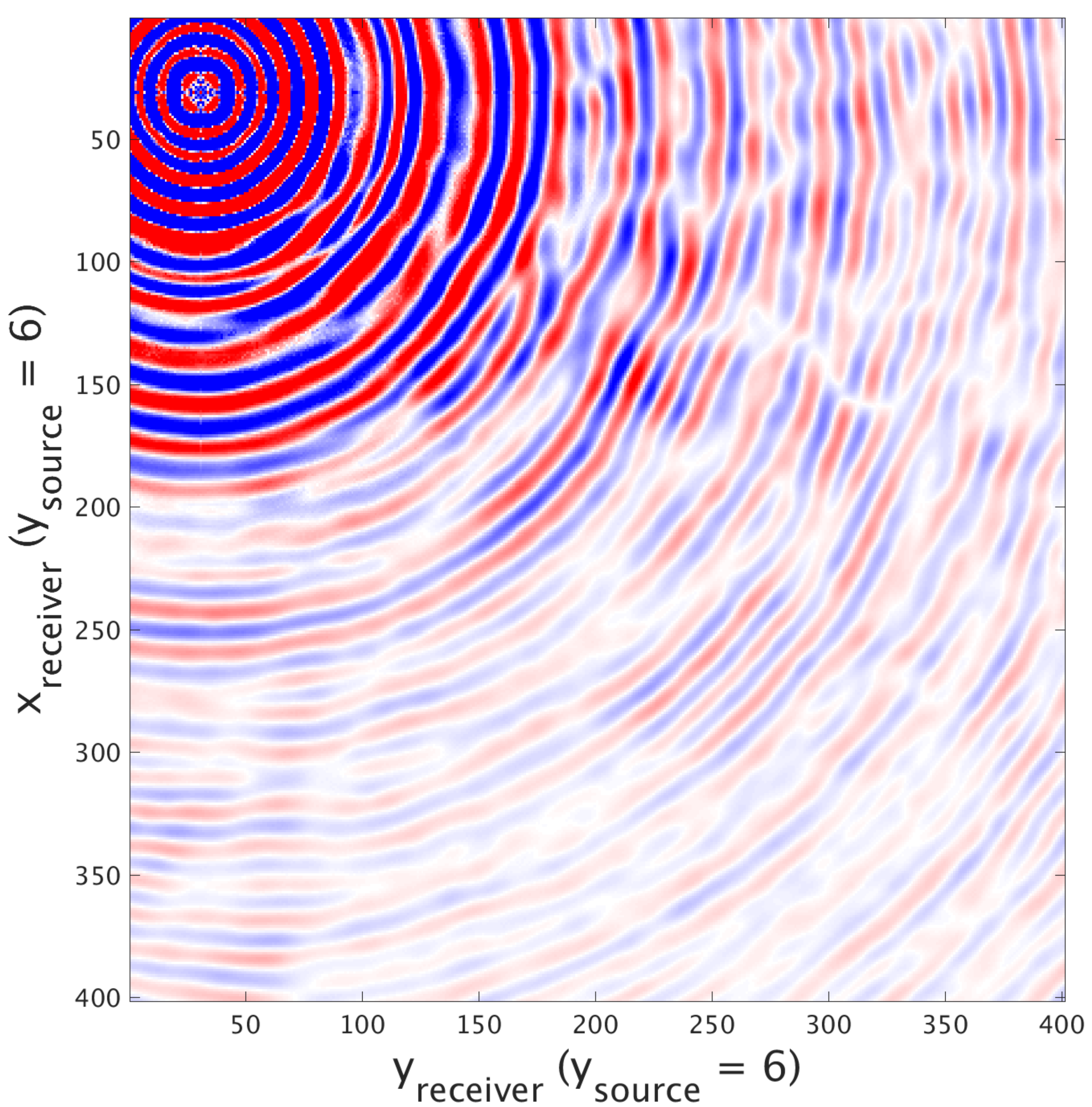
Alternations: 7

SNR = 25.3 dB

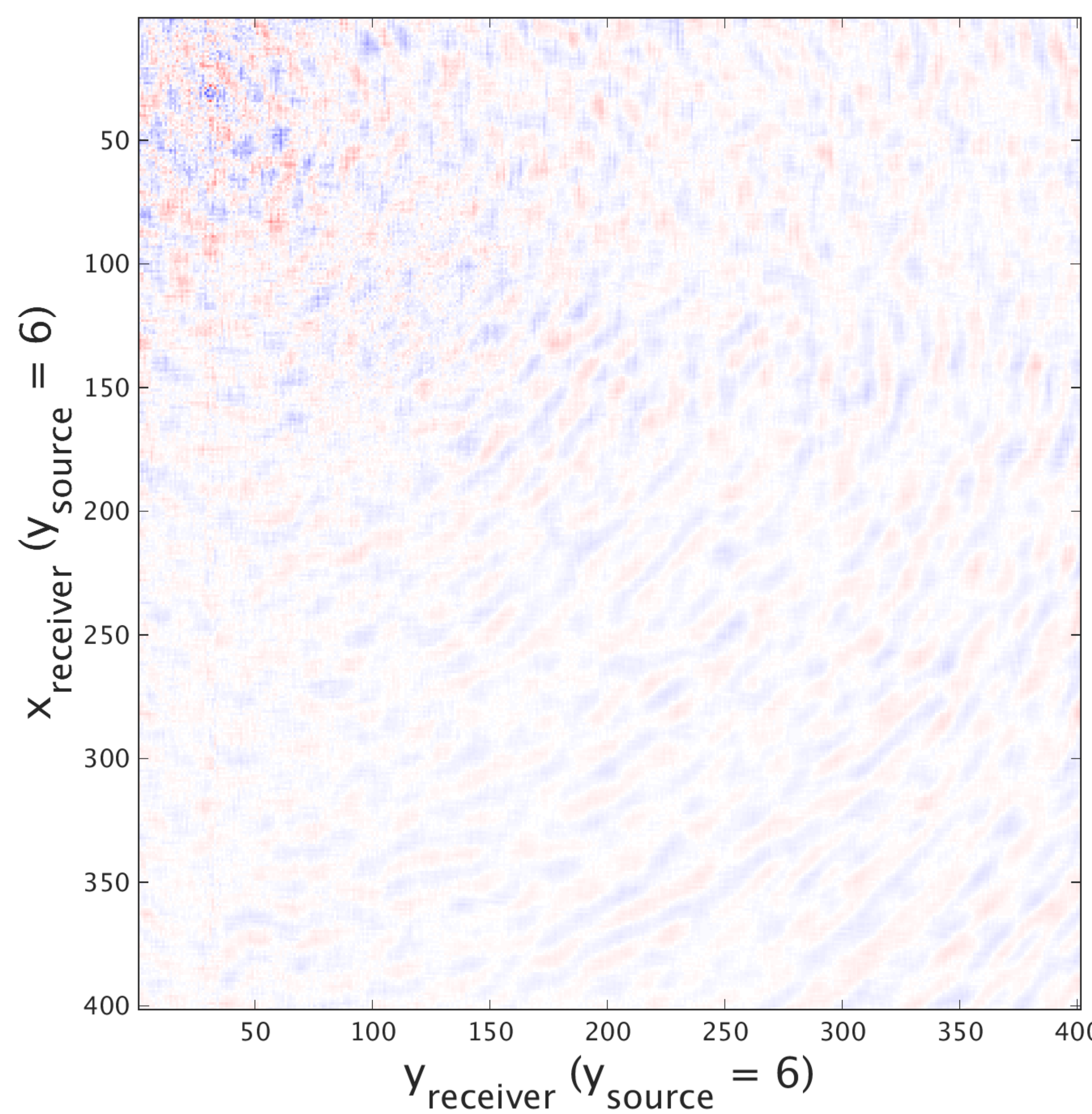
Time = 5 hrs and 20 mins

Results: Decoupling Method

True Source Gather



Difference Plot



40 processors

Alternations: 7

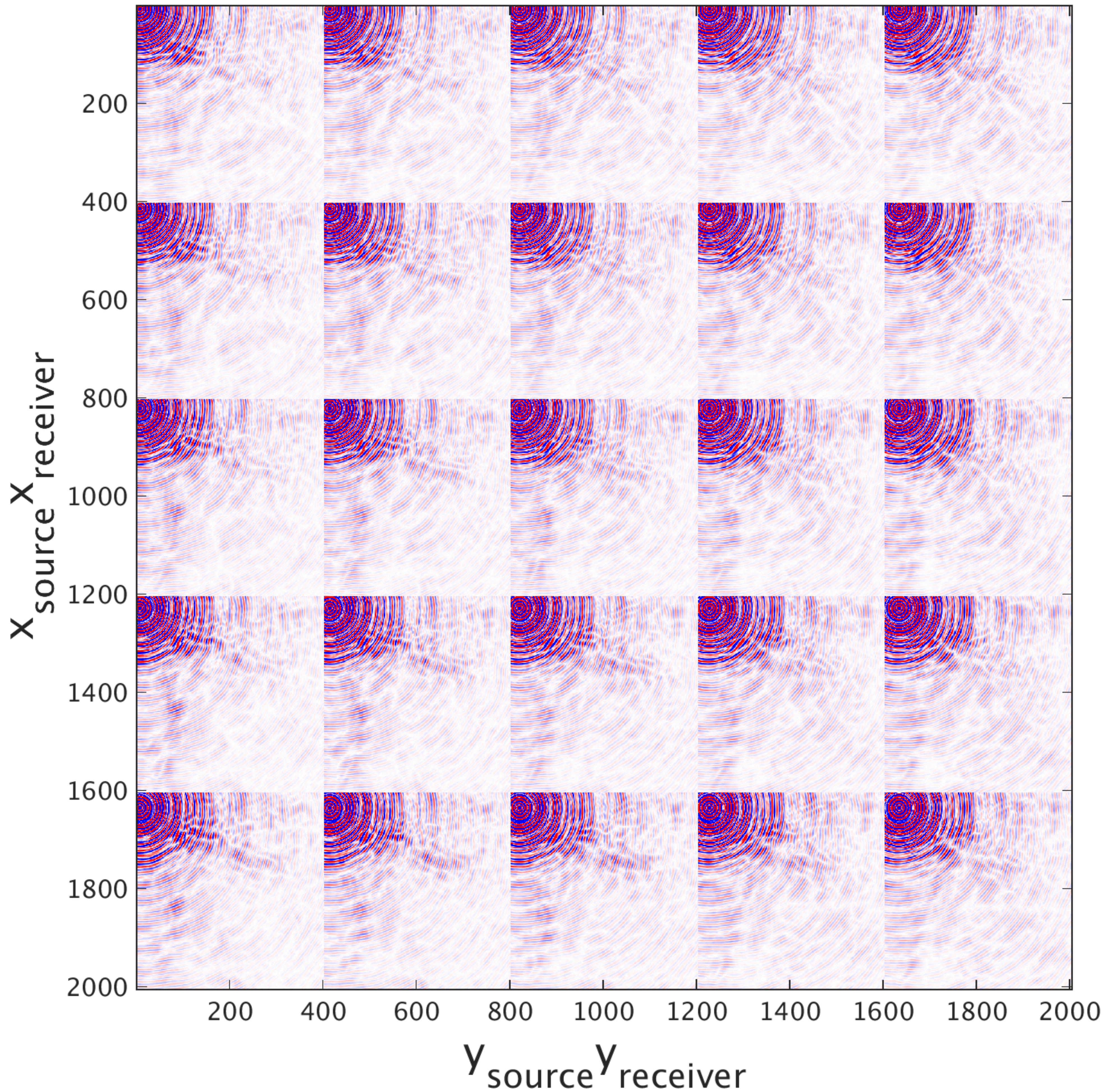
SNR = 25.3 dB

Time = 5 hrs and 20 mins

3D Interpolation Experiment

BG 3D Dataset

12.3 Hz



Size: 26,867 x 26,867

(full slice, no windowing)

Remove 80 % of Receivers randomly

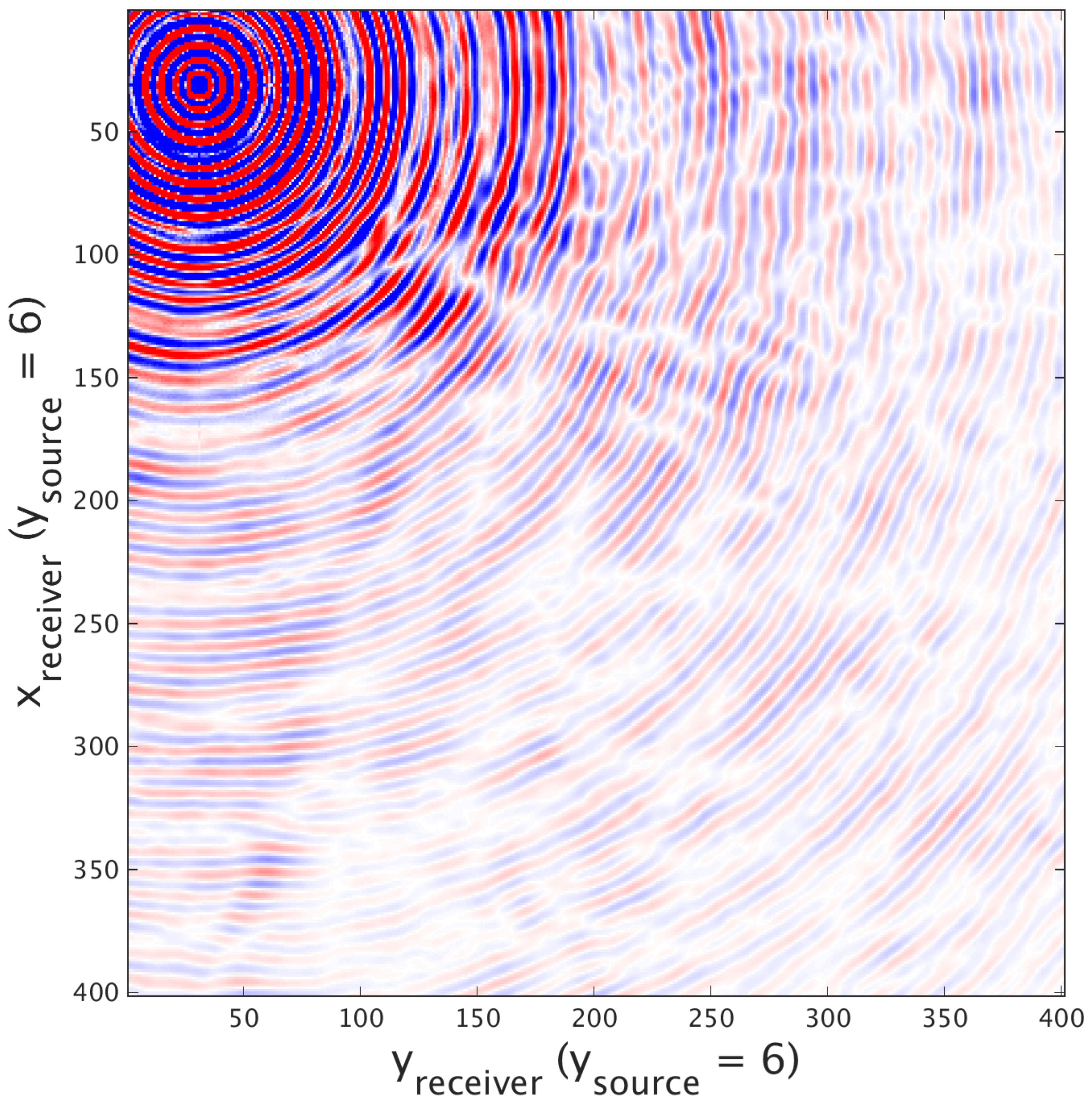
Compare Interpolation via:

- SPG-LR

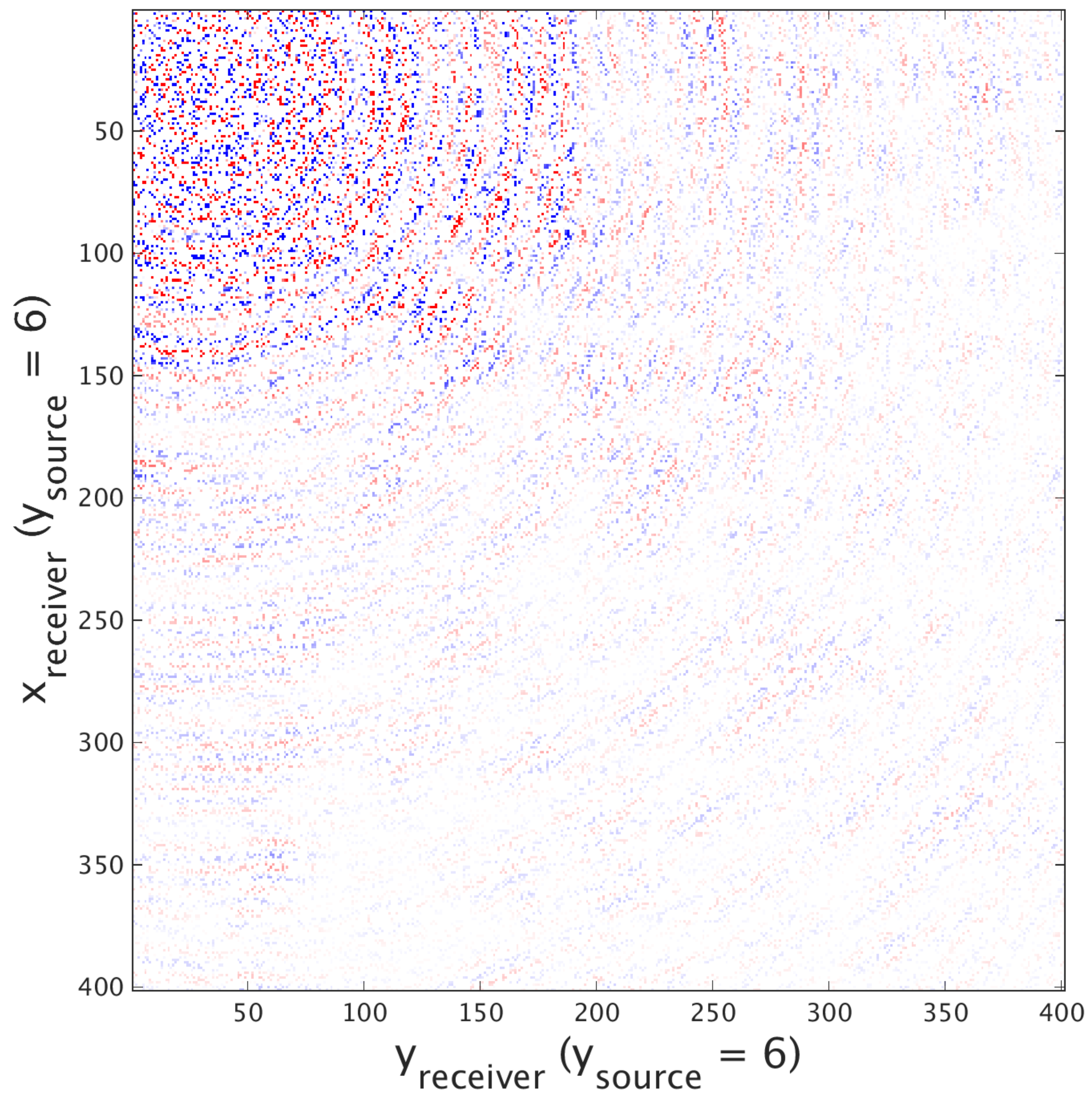
- Decoupling Method

Common Source Gather

True Source Gather



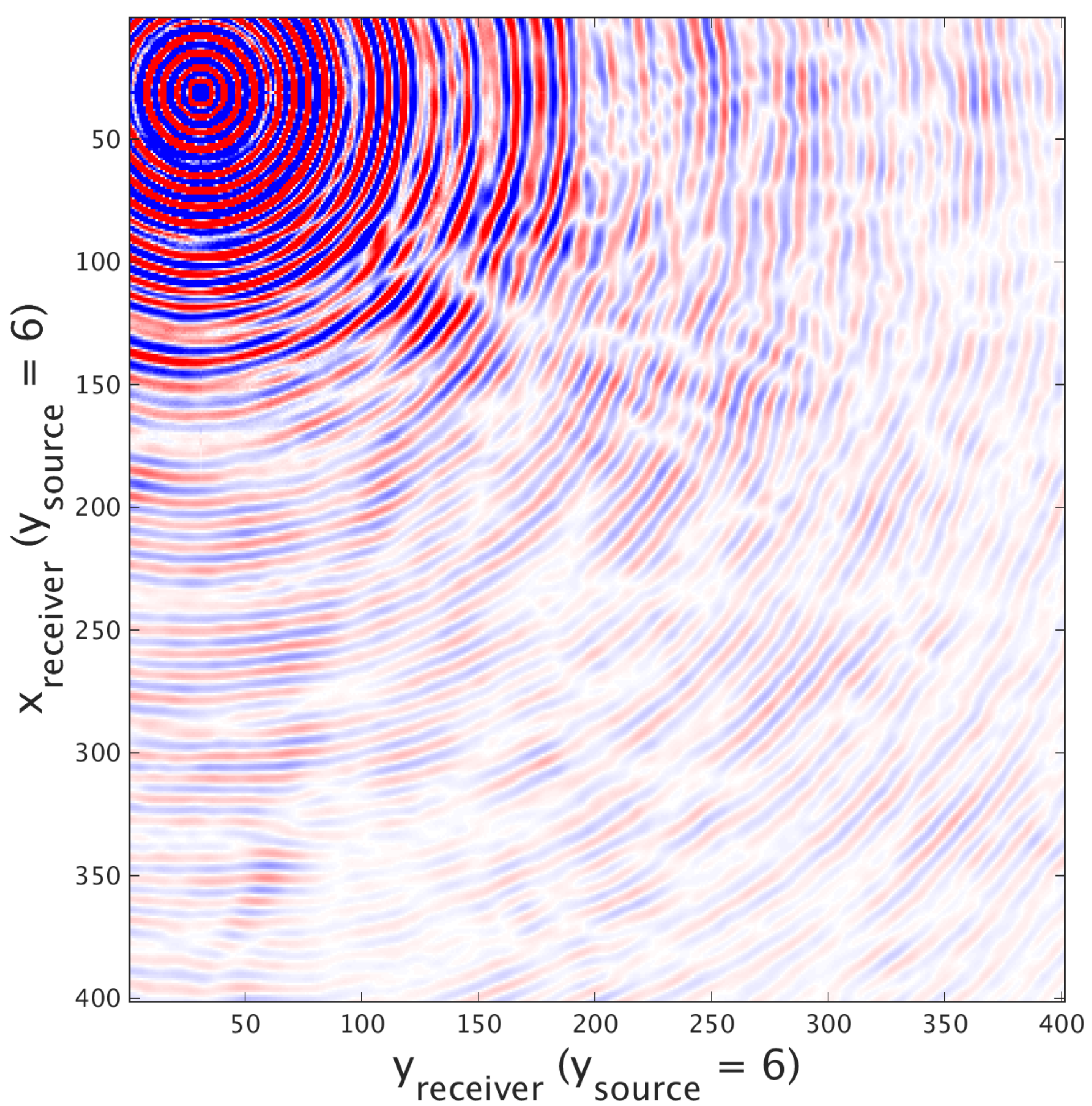
Subsampled Source Gather



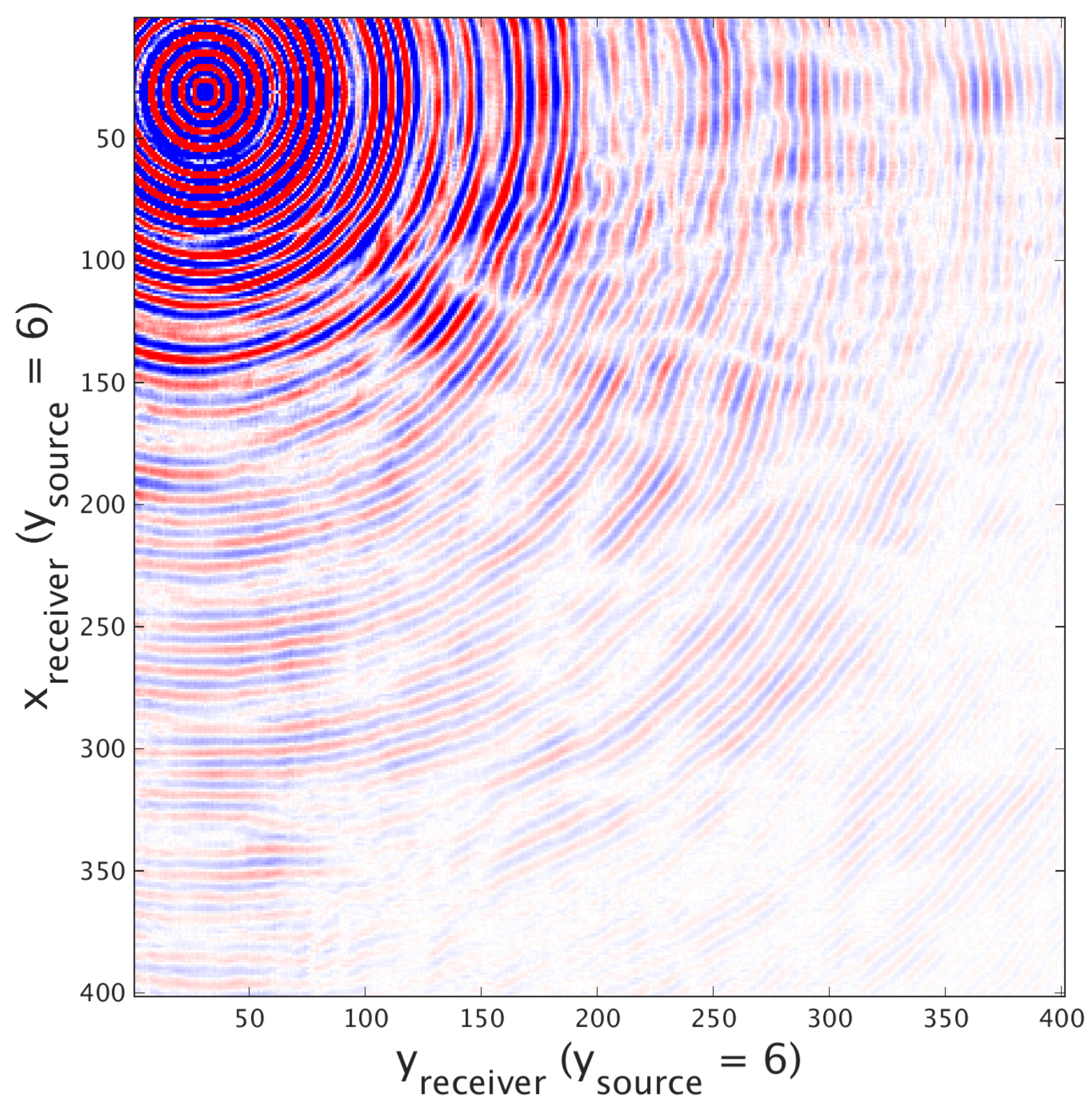
Remove 80 % of Receivers randomly

Results: SPG-LR

True Source Gather



Recovered Source Gather



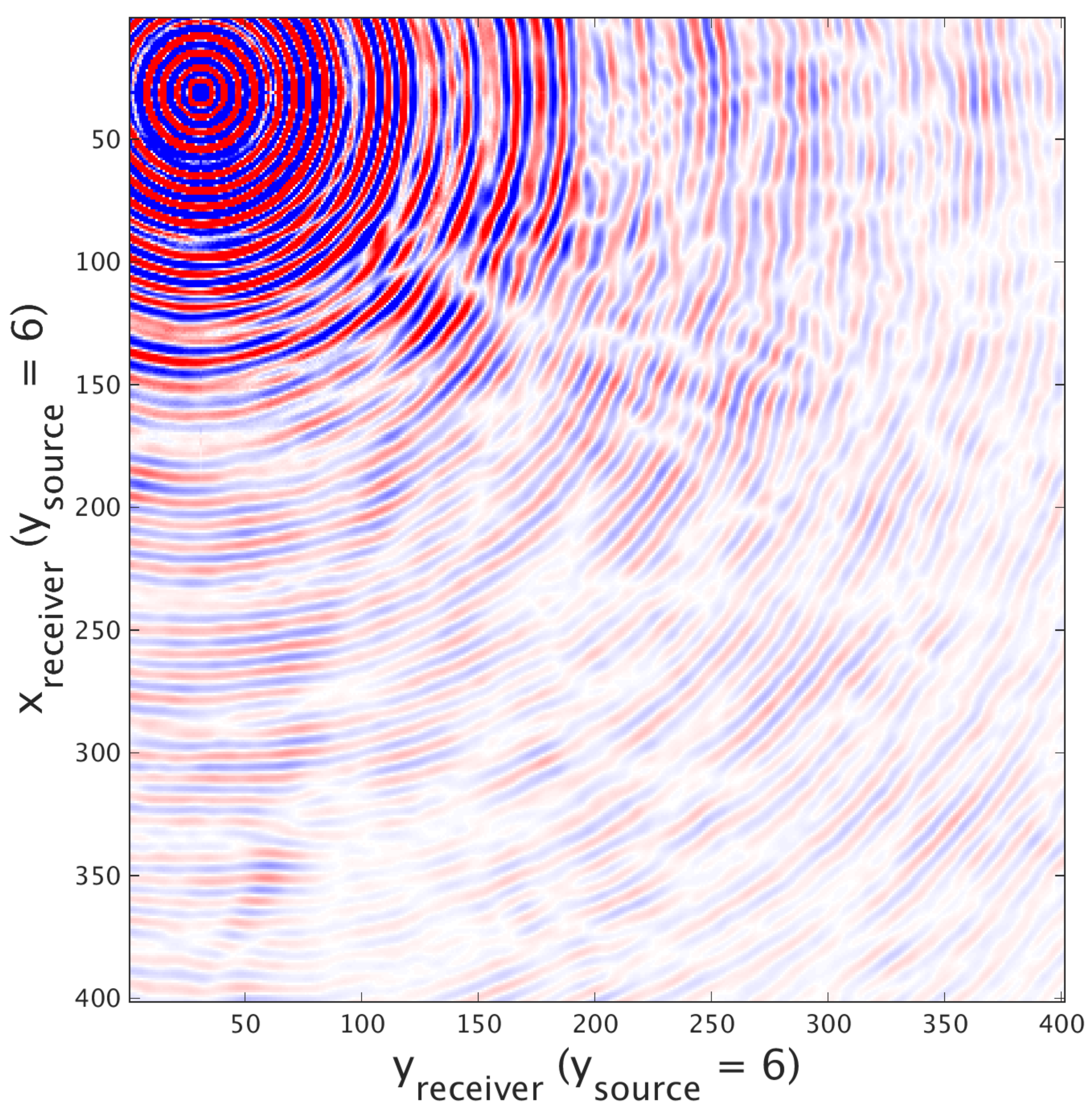
SPG-ℓ_1 iterations: 400

SNR = 20.5 dB

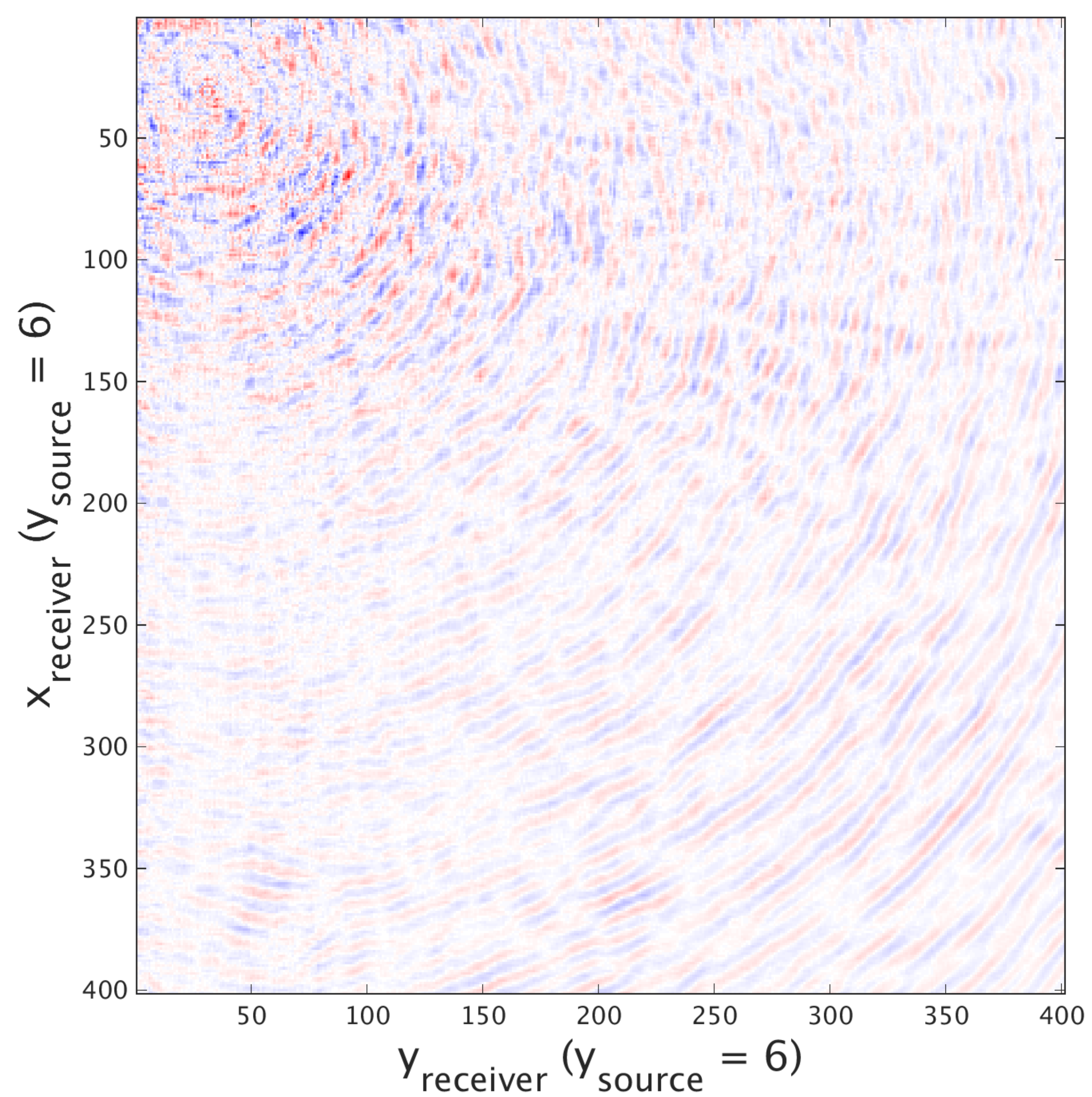
Time = 137 hrs and 20 min

Results: SPG-LR

True Source Gather



Difference Plot



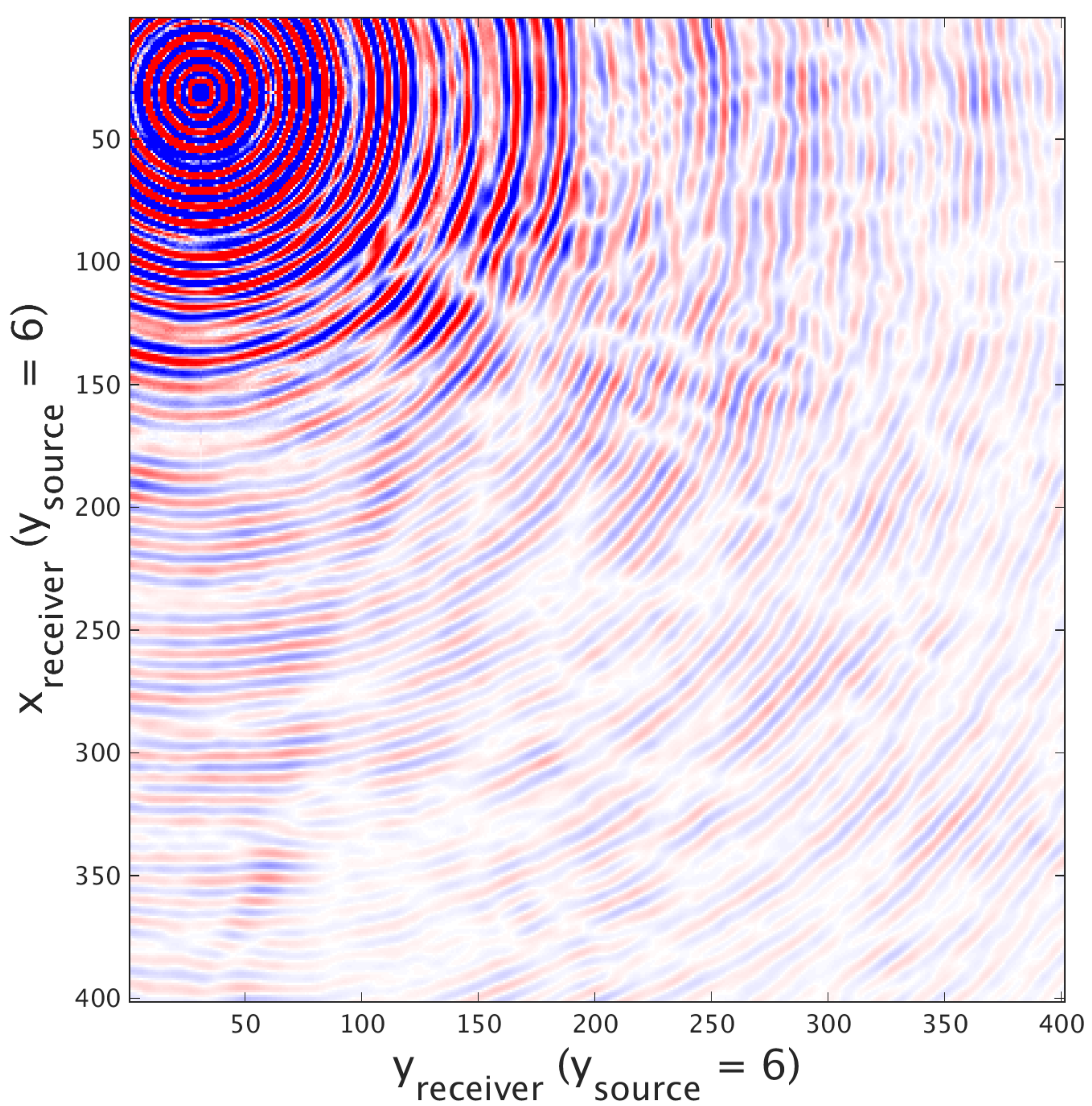
SPG-ℓ_1 iterations: 400

SNR = 20.5 dB

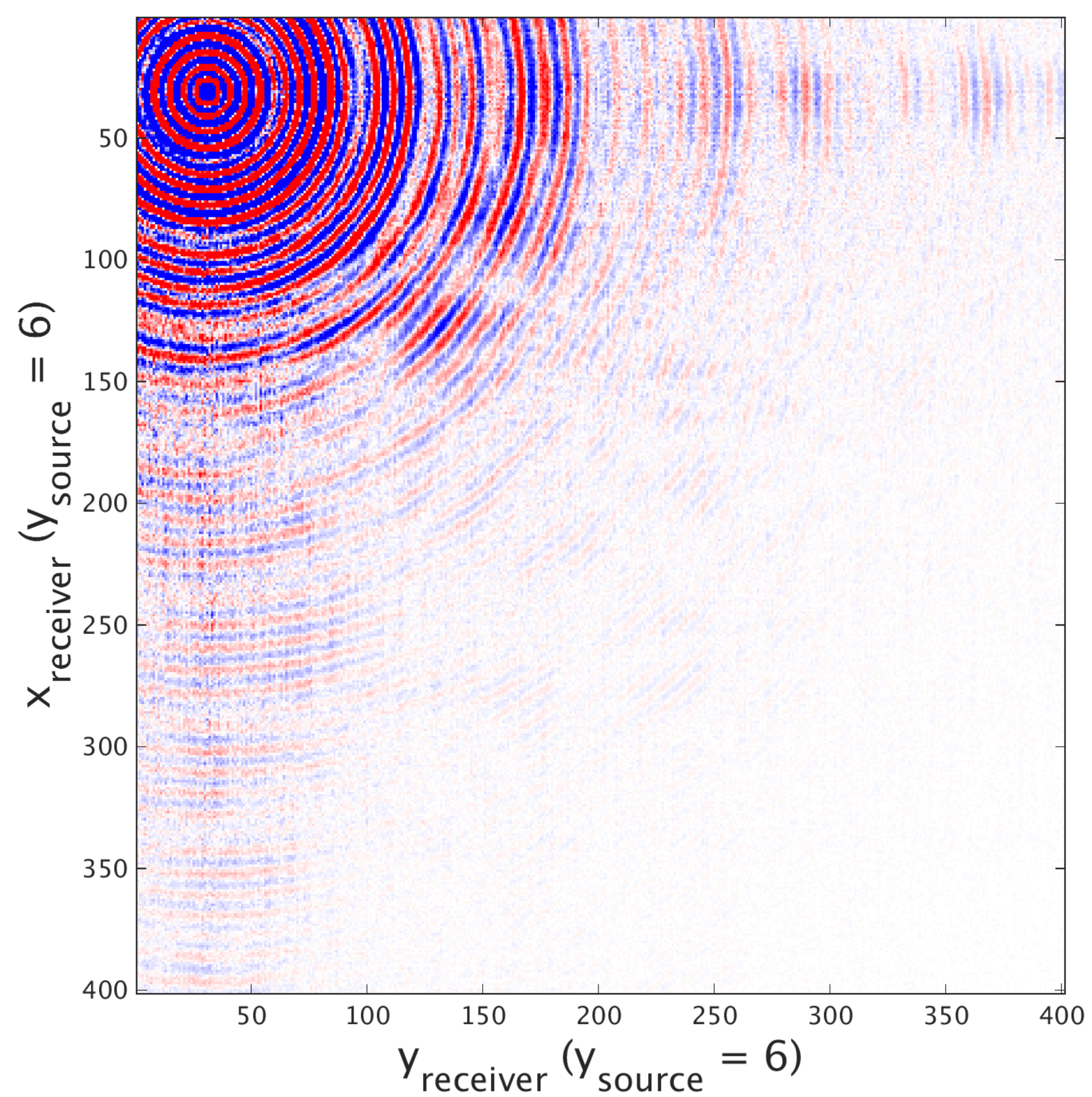
Time = 137 hrs and 20 min

Results: Decoupling Method

True Source Gather



Recovered Source Gather



40 processors

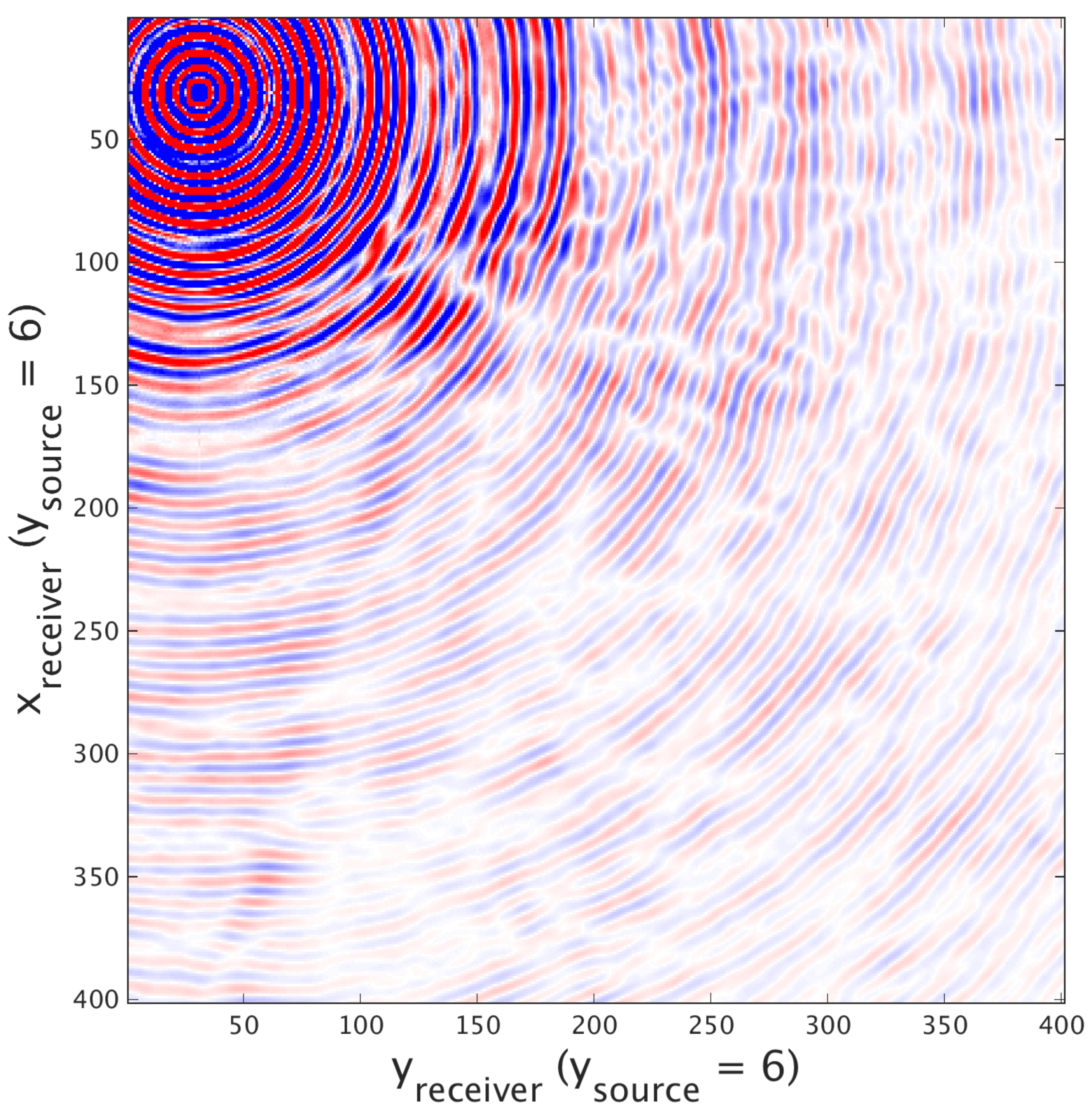
Alternations: 2

SNR = 12.4 dB

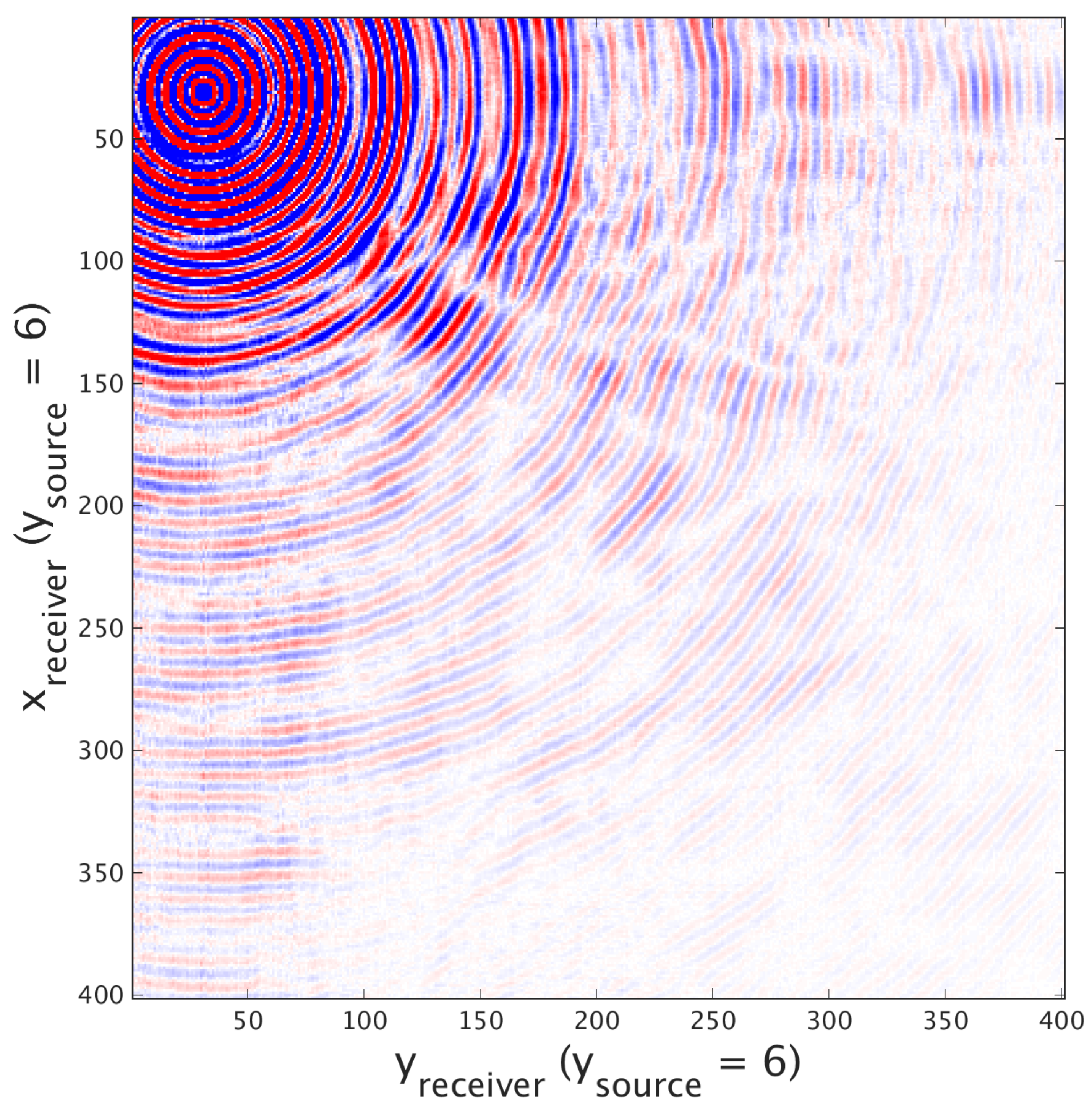
Time = 1 hrs and 43 mins

Results: Decoupling Method

True Source Gather



Recovered Source Gather



40 processors

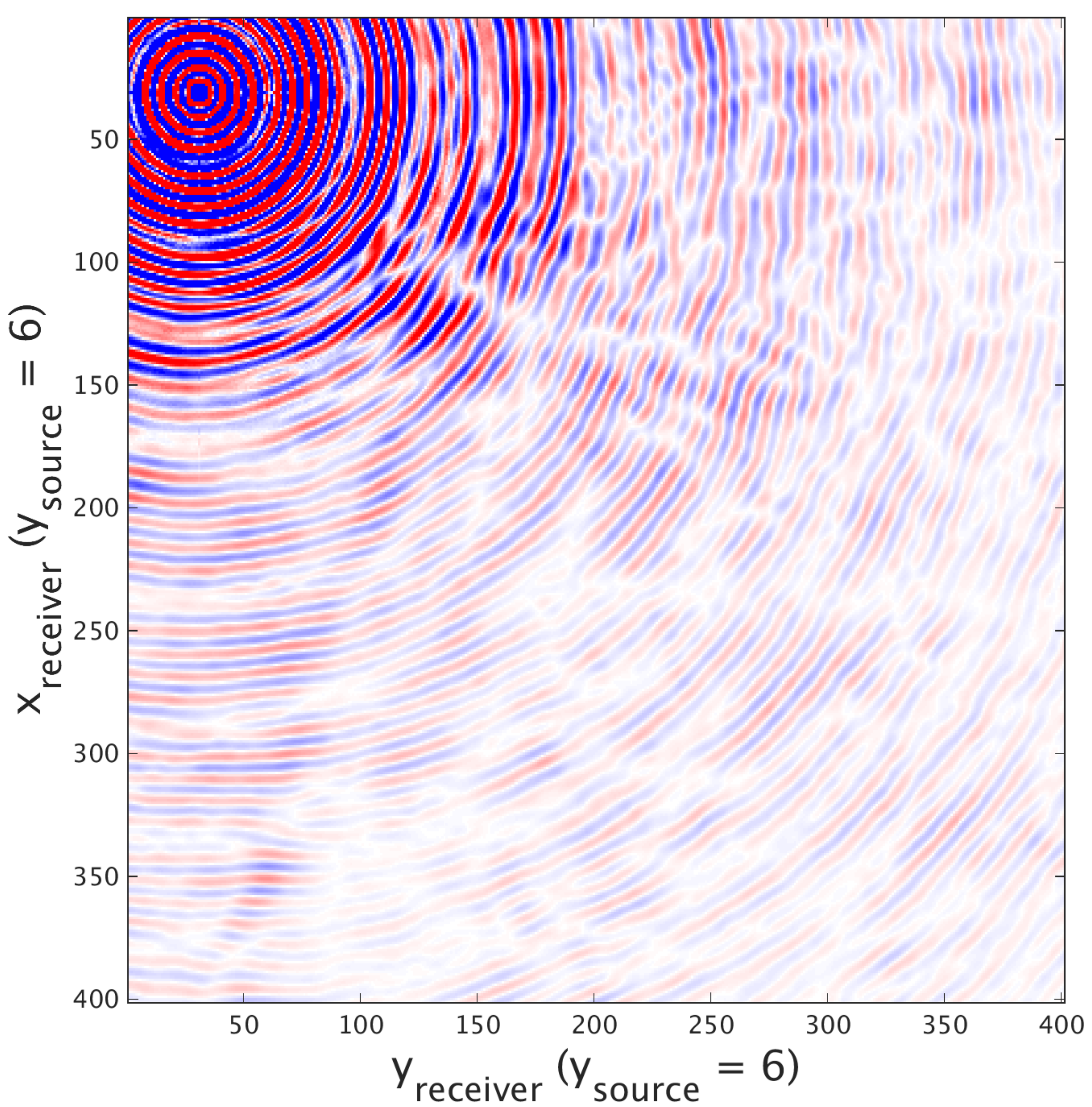
Alternations: 5

SNR = 19 dB

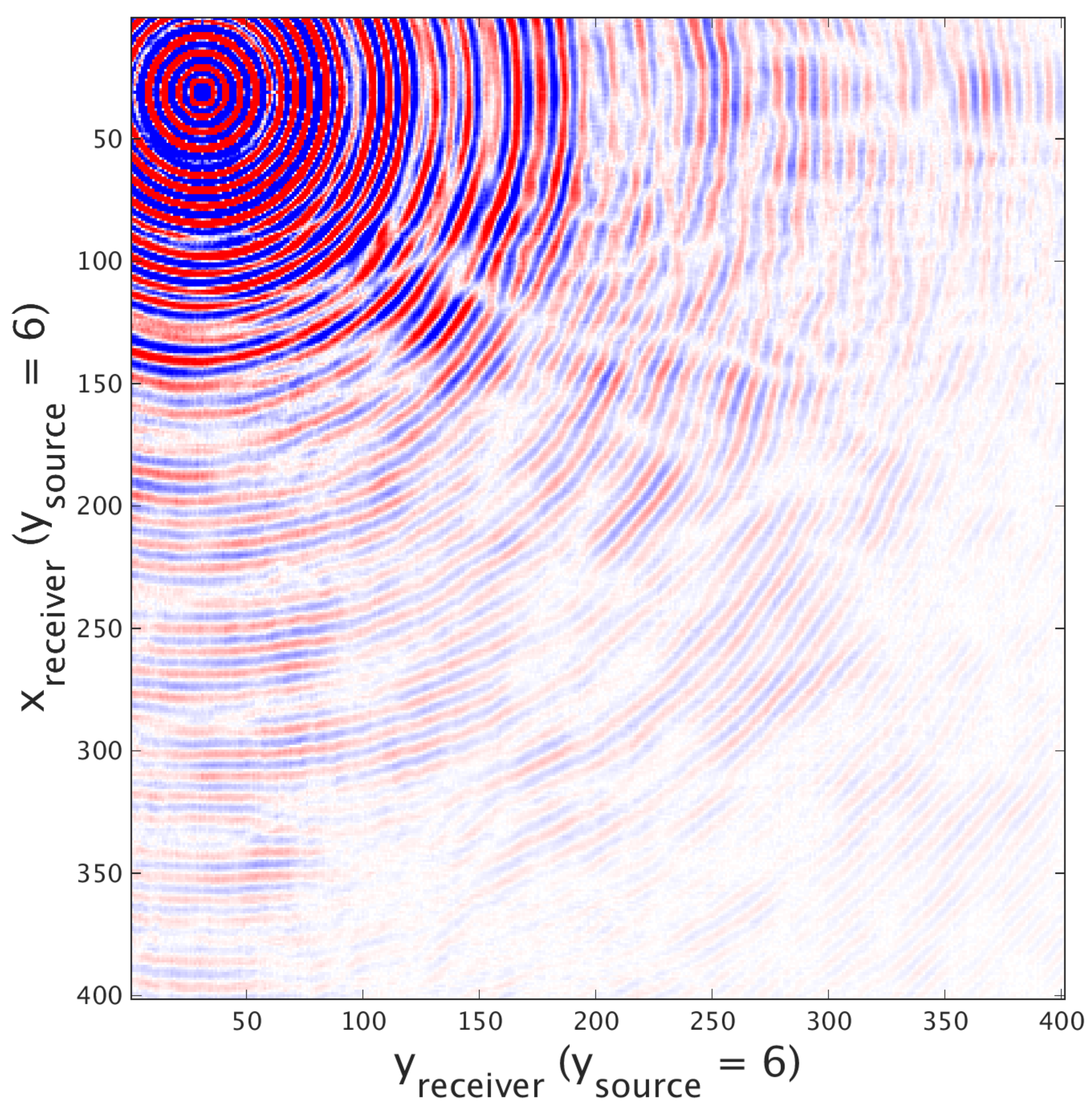
Time = 4 hrs and 43 mins

Results: Decoupling Method

True Source Gather



Recovered Source Gather



40 processors

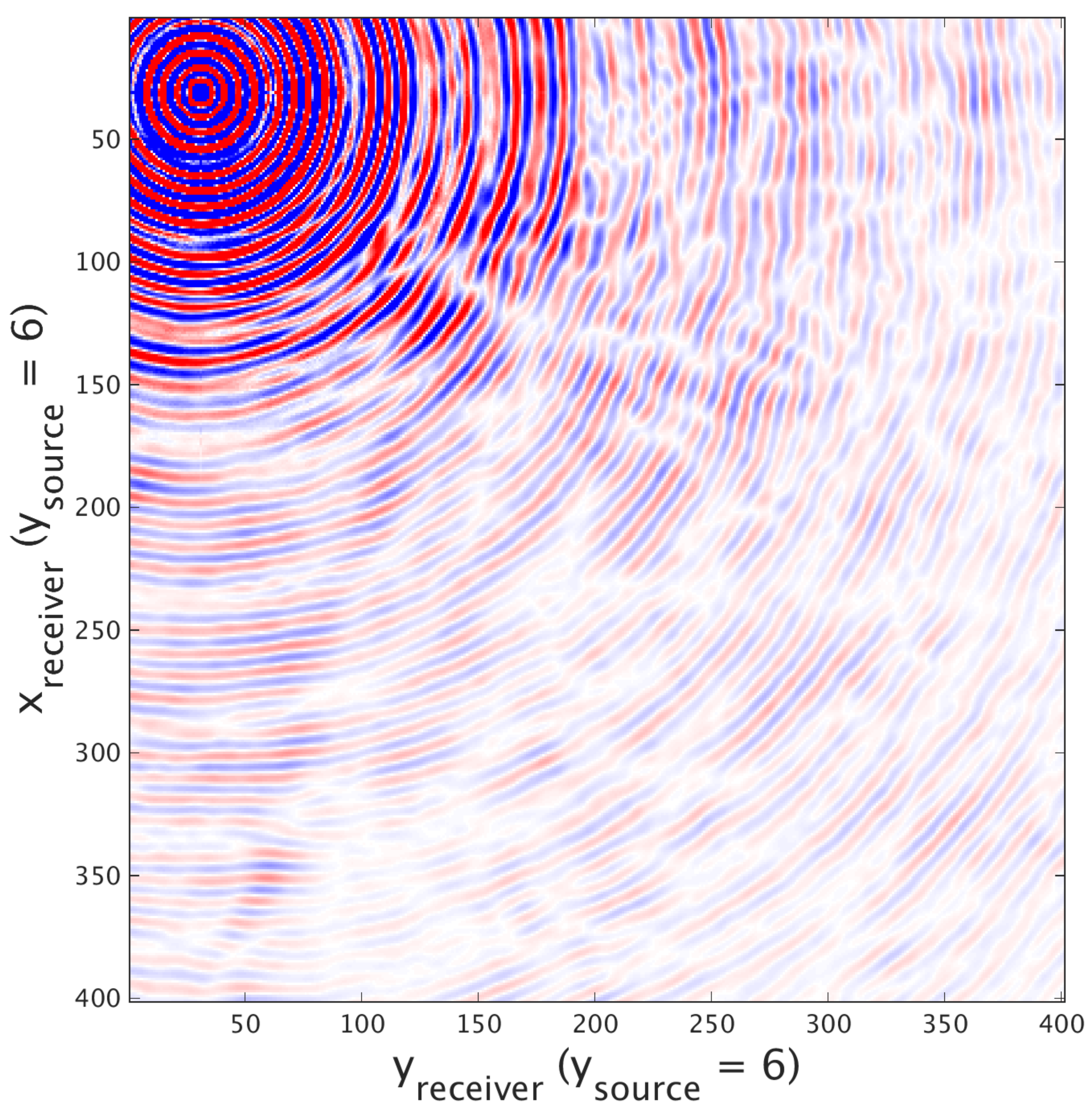
Alternations: 7

SNR = 20 dB

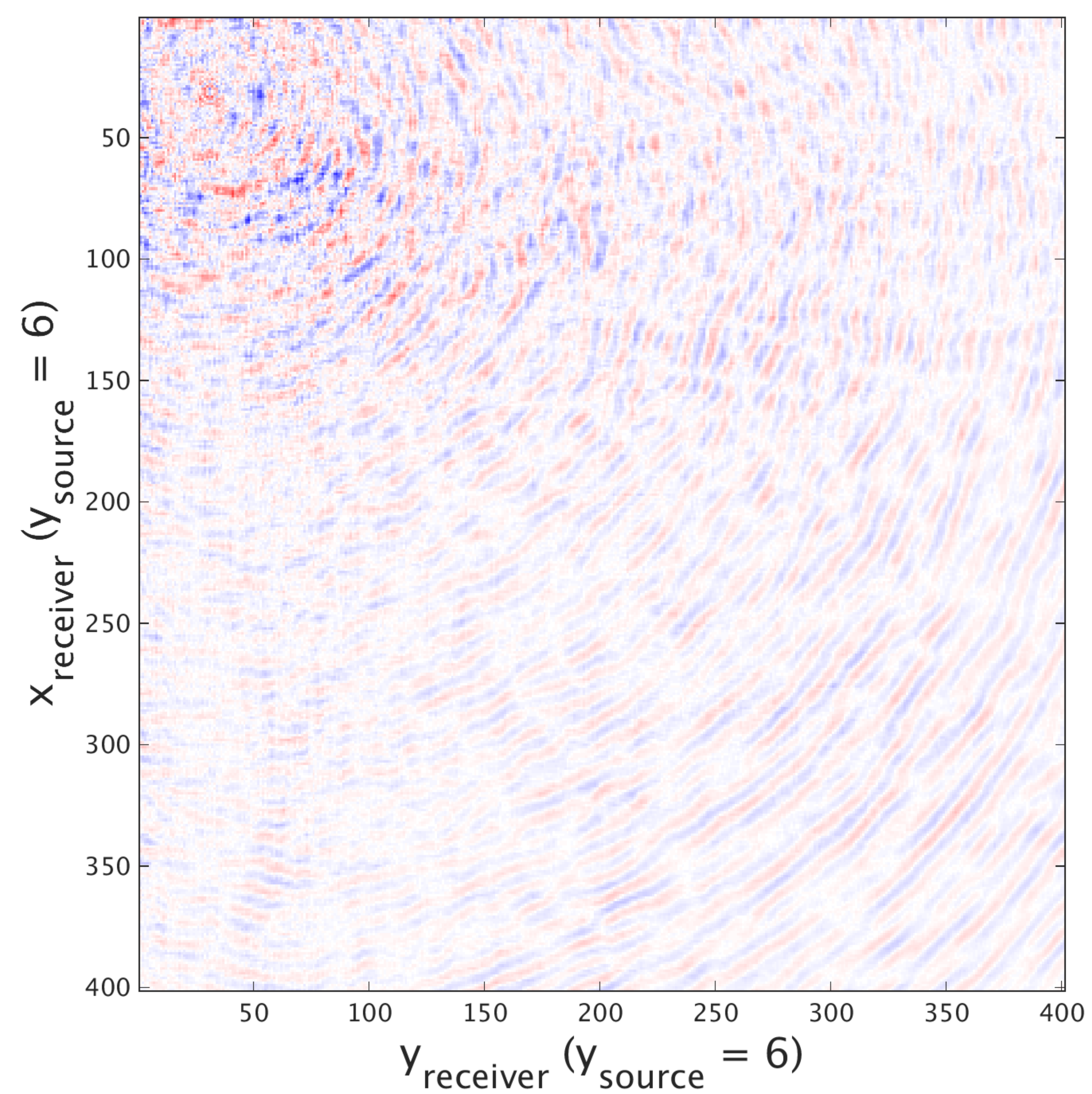
Time = 6 hrs and 19 mins

Results: Decoupling Method

True Source Gather



Difference Plot



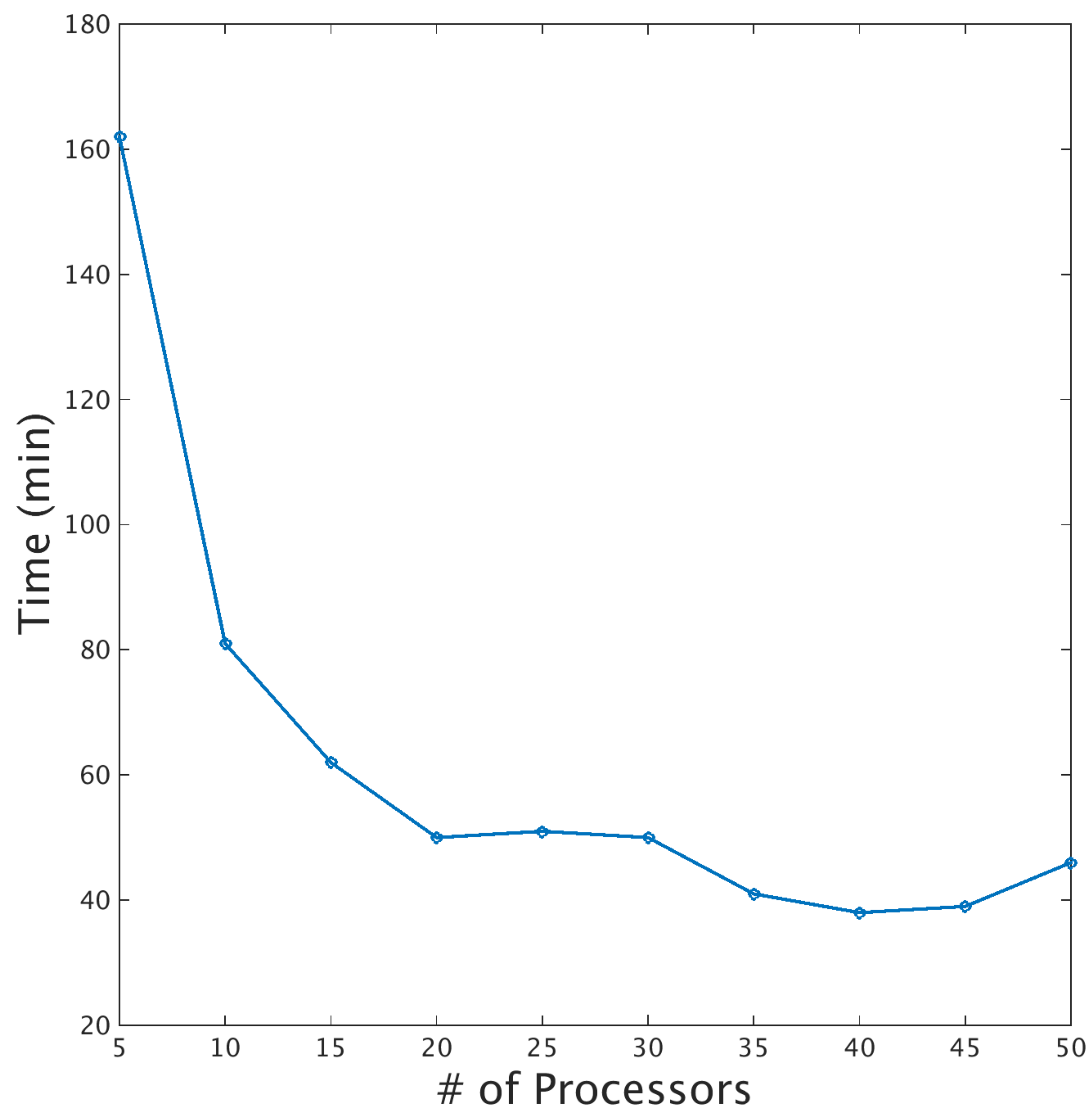
40 processors

Alternations: 7

SNR = 20 dB

Time = 6 hrs and 19 mins

Computation Time vs. # of Processors



Matrix Size: 26,867 x 26,867

(full slice, no windowing)

Computation time of 1 alternation

Conclusions

- Significant improvement in computation time
- Equivalent SNR output
- No need to form full matrices
- Parameter free

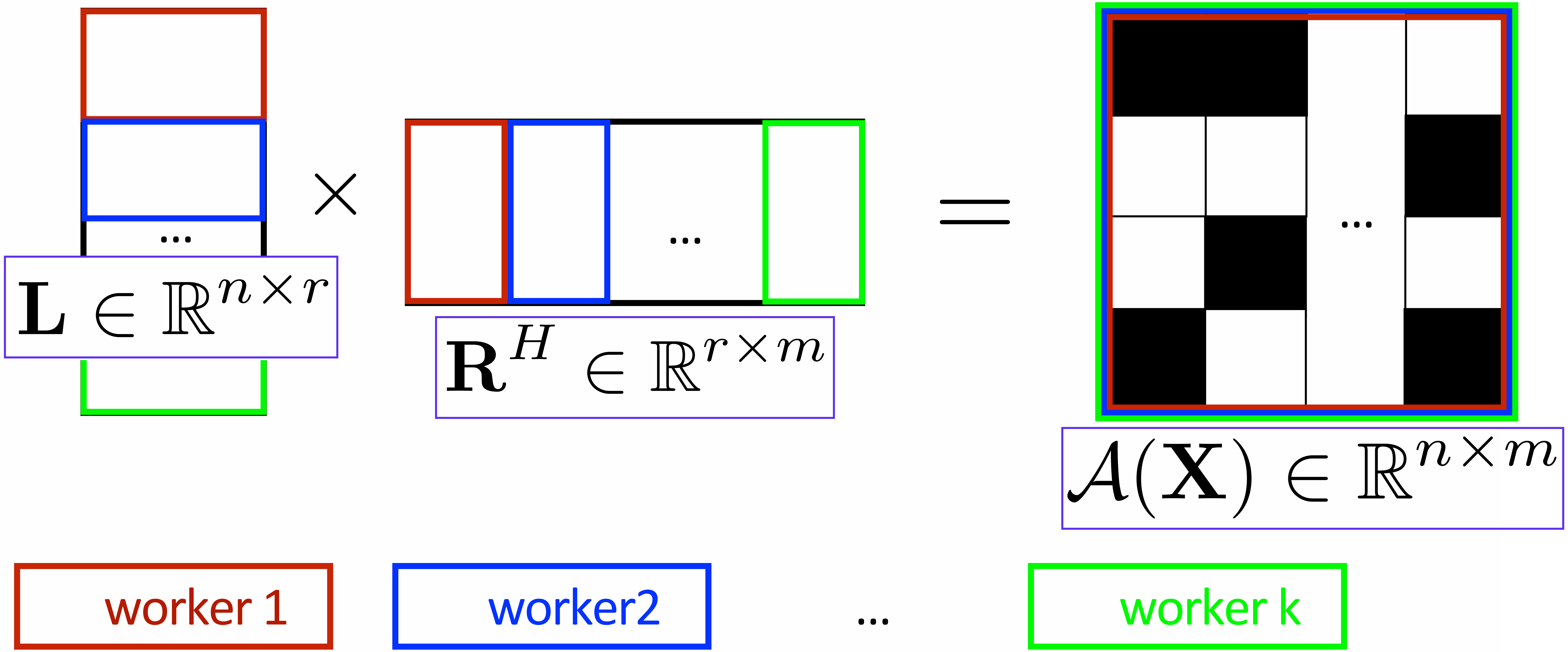
Future Work

- Design for other measurement operators, $\mathcal{A}$ (see Rajiv's talk).

- Fully parallel version
  – design for distributed arrays, e.g., distributed QR factorization.
  – no need to store full factor in each worker.

- Julia implementation.

Fully Parallel Version



Acknowledgements

This research was carried out as part of the SINBAD project with the support of the member organizations of the SINBAD Consortium.

Software release coming soon

SINBAD

Thank you for your attention!