Time-Regularized Blind Deconvolution Approach for Radio Interferometry

P.-A. Thouvenin, A. Repetti, A. Dabbech, and Y. Wiaux

BASP Group
Heriot-Watt University, UK

IEEE SAM Workshop
Sheffield, UK – July 11, 2018

Fourier imaging and calibration problem:

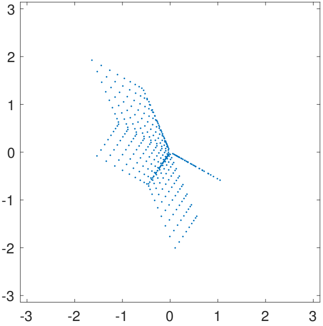
$$\mathbf{y} = \overline{\mathbf{G}}\mathcal{F}(\overline{\mathbf{x}}) + \mathbf{w}$$

OBJECTIVE: Find an estimate of the original image $\overline{\mathbf{x}}$ from the observations $\mathbf{y} \in \mathbb{C}^M$.

- $\overline{\mathbf{x}}$ is the original unknown image
- $\mathcal{F}$ is the 2D Fourier transform operator
- $\overline{\mathbf{G}}$ contains the unknown convolution kernels centred at the measured frequencies
- $\mathbf{w}$ is a realization of an additive i.i.d. Gaussian noise

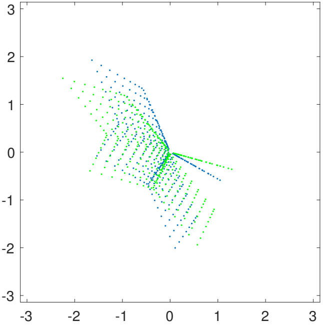
Radio interferometric imaging

VLA radio telescope (27 antennas)
Credit: NRAO



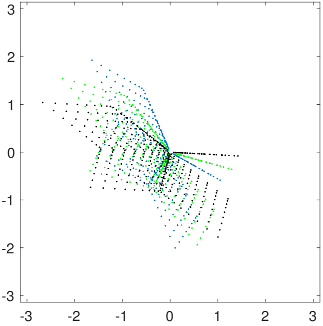
Fourier sampling at instant $t = 1$

VLA radio telescope (27 antennas)
Credit: NRAO



Fourier sampling at instant $t = 20$

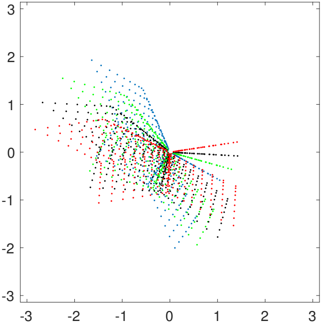
VLA radio telescope (27 antennas)
Credit: NRAO



Fourier sampling at instant $t = 40$

Radio interferometric imaging

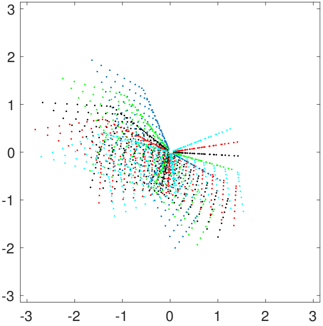
VLA radio telescope (27 antennas)
Credit: NRAO



Fourier sampling at instant $t = 60$

Radio interferometric imaging

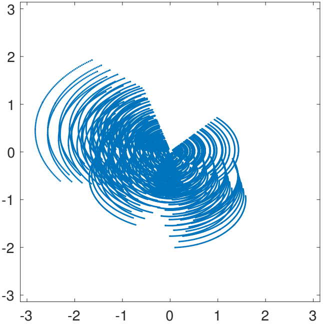
VLA radio telescope (27 antennas)
Credit: NRAO



Fourier sampling at instant $t = 80$

Radio interferometric imaging

VLA radio telescope (27 antennas)
Credit: NRAO



Fourier sampling for $T = 100$ time instant measurements

Observation model

- Interferometer with $\boxed{n_a \text{ antennas}}$ $\rightsquigarrow$ $M = n_a(n_a - 1)/2$ antenna pairs
- $T \geqslant 1$ measurements per antenna pair

$\rightsquigarrow$ TM measurements
indexed by t, α, β, with $t \in \{1, \ldots, T\}$ and $1 \leqslant \alpha < \beta \leqslant n_a$

Measurement acquired by the antenna pair (α, β) at the time instant t, at the spatial frequency $\mathbf{k}_{t,\alpha,\beta} = \mathbf{k}_{t,\alpha} - \mathbf{k}_{t,\beta}$:

$$y_{t,\alpha,\beta} = \sum_{n=-N/2}^{N/2-1} \overline{d}_{t,\alpha}(n)\overline{d}^*_{t,\beta}(n)\overline{x}(n)e^{-2i\pi(k_{t,\alpha}-k_{t,\beta})\frac{n}{N}} + w_{t,\alpha,\beta}$$

Observation model

$$\mathrm{y}_{t,\alpha,\beta} = \sum_{n=-N/2}^{N/2-1} \overline{\mathrm{d}}_{t,\alpha}(n)\overline{\mathrm{d}}^*_{t,\beta}(n)\overline{\mathrm{x}}(n)\mathrm{e}^{-2\mathrm{i}\pi(k_{t,\alpha}-k_{t,\beta})\frac{n}{N}} + \mathrm{w}_{t,\alpha,\beta}$$

- $\overline{\mathbf{x}} = (\overline{\mathrm{x}}(n))_n \in \mathbb{R}^N$
  $\rightsquigarrow$ unknown original image.

- $\overline{\mathbf{d}}_{t,\alpha} = (\overline{\mathrm{d}}_{t,\alpha}(n))_n \in \mathbb{C}^N$
  $\rightsquigarrow$ unknown direction-dependent effect (DDE) related to antenna $\alpha \in \{1, \dots, n_a\}$ and depending on the time instant $t \in \{1, \dots, T\}$

- $\mathbf{w} = \left(\mathrm{w}_{t,\alpha,\beta}\right)_{t,\alpha,\beta} \in \mathbb{C}^{TM}$
  $\rightsquigarrow$ realization of an additive i.i.d. Gaussian noise

Observation model

$$\boxed{\mathrm{y}_{t,\alpha,\beta} = \sum_{n=-N/2}^{N/2-1} \overline{\mathrm{d}}_{t,\alpha}(n)\overline{\mathrm{d}}^*_{t,\beta}(n)\overline{\mathrm{x}}(n)\mathrm{e}^{-2\mathrm{i}\pi(k_{t,\alpha}-k_{t,\beta})\frac{n}{N}} + \mathrm{w}_{t,\alpha,\beta}}$$

Particular case:

If, for every $n \in \{-N/2, \ldots, N/2-1\}$, $\overline{\mathrm{d}}_{t,\alpha}(n) = \delta_{t,\alpha} \in \mathbb{C}$, then $\overline{\mathbf{d}}_{t,\alpha}$ reduces to a direction-independent effect (DIE).

$\rightsquigarrow$ Approximation usually considered when the DIEs and DDEs need to be calibrated.

Observation model

$$y_{t,\alpha,\beta} = \sum_{n=-N/2}^{N/2-1} \overline{\mathrm{d}}_{t,\alpha}(n)\overline{\mathrm{d}}^*_{t,\beta}(n)\overline{\mathrm{x}}(n)\mathrm{e}^{-2\mathrm{i}\pi(k_{t,\alpha}-k_{t,\beta})\frac{n}{N}} + \mathrm{w}_{t,\alpha,\beta}$$

OBJECTIVE: Find an estimate of

- $(\forall t \in \{1,\ldots,T\})(\forall \alpha \in \{1,\ldots,n_a\})$ the DDEs $\overline{\mathbf{d}}_{t,\alpha}$,
- the original image $\overline{\mathbf{x}}$.

State of the art calibration approaches

- DIEs can be approximately known by performing calibration transfer
- StEFCal: Bilinear approach to solve the least squares minimization problem associated with the DIE calibration problem when the source is approximately known [Salvini & Wijnholds (2014)]
- Faceting: Divide the field of view into facets and see each facet as an image with unknown DIEs [Tasse (2014), Smirnov & Tasse (2015), van Weeren et al. (2016)]
  - ⇝ piecewise constant DDEs
  - ⇝ method designed for images with mainly point sources

State of the art imaging approaches

Two main *families* of approaches:

- CLEAN based methods: greedy-based approaches [Högbom (1974), Schwarz (1978), Thompson et al. (2001)]

  $\rightsquigarrow$ ℓ_1 regularization on the image space

- Use of compressive sensing theory and convex optimization methods [Wiaux et al. (2009), Wenger et al. (2010), Li et al. (2011), McEwen & Wiaux (2011), Carrillo et al. (2012), Onose et al. (2017)]

  $\rightsquigarrow$ ℓ_1 regularization on a sparsity basis

Joint calibration and imaging method for RI?

- **Current methods?**

  Alternate between the estimation of the DIEs/DDEs and the estimation of the image, using the preferred methods of the user.

  $\rightsquigarrow$ No theoretical convergence guarantee

  $\rightsquigarrow$ For DDEs, only works for images with point sources

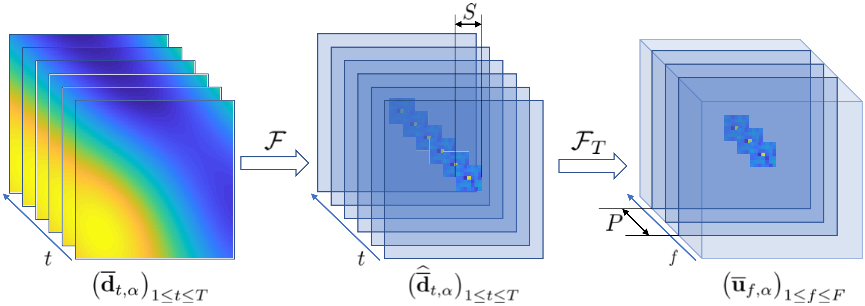
- **Proposed approach?**

  Design a global algorithm to solve the joint calibration and imaging problem, with convergence guarantees, and able to estimate sophisticated images with complex structures

  $\rightsquigarrow$ Use non-convex optimization techniques

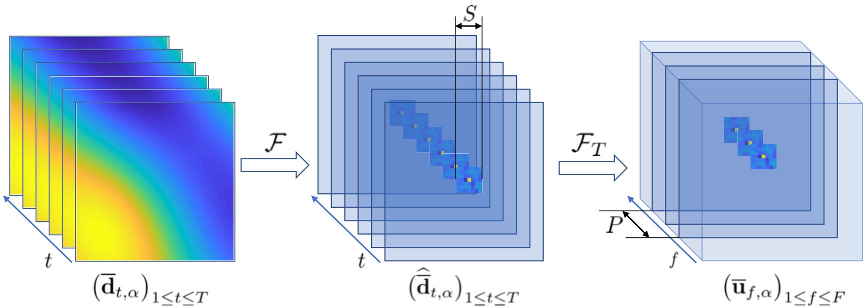
Prior information on the DDEs

The DDEs are smooth both in space and in time



Prior information on the DDEs

The DDEs are smooth both in space and in time



OBJECTIVE:

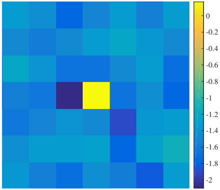
For every $\alpha \in \{1, \ldots, n_a\}$, find an estimate $\mathbf{u}_\alpha^\star \in \mathbb{C}^{S^2 \times P}$ of the non-zero Fourier coefficients $\overline{\mathbf{u}}_\alpha \in \mathbb{C}^{S^2 \times P}$ of the DDEs $(\overline{\mathbf{d}}_{t,\alpha})_{1 \leqslant t \leqslant T}$.

Assume that calibration transfer has been performed:

DIEs are normalized and are approximately known

i.e., for every $\alpha \in \{1, \ldots, n_a\}$ and $t \in \{1, \ldots, T\}$,

- the central coefficient of $\overline{\mathbf{u}}_{t,\alpha}$ belongs to a complex neighbourhood of 1.
- the other coefficients of $\overline{\mathbf{u}}_{t,\alpha}$ belong to a complex neighbourhood of 0.



Prior information of the image

Compute a preprocessing step to obtain a first approximation of the image

Solve the imaging problem considering normalized DIEs (without DDEs)

$$\mathbf{y} = \overline{\mathbf{G}}\mathcal{F}(\mathbf{x}) + \mathbf{w} \simeq \mathcal{S}\big(\mathcal{F}(\mathbf{x})\big) + \mathbf{w}$$

where

- $\mathcal{F}\colon \mathbb{C}^N \to \mathbb{C}^K$ represents the 2D Fourier transform operator
- $\overline{\mathbf{G}} \in \mathbb{C}^{TM \times K}$ contains the unknown antenna-based gains centred at the frequencies measured by the antenna pairs
- $\mathcal{S}\colon \mathbb{C}^K \to \mathbb{C}^{TM}$ linear operator selecting the frequencies measured by the antenna pairs

Prior information of the image

Compute a preprocessing step to obtain a first approximation of the image

Solve the imaging problem considering normalized DIEs (without DDEs)

$$\mathbf{y} = \overline{\mathbf{G}}\mathcal{F}(\mathbf{x}) + \mathbf{w} \simeq \mathcal{S}\big(\mathcal{F}(\mathbf{x})\big) + \mathbf{w}$$

where

- $\mathcal{F}\colon \mathbb{C}^N \to \mathbb{C}^K$ represents the 2D Fourier transform operator
- $\overline{\mathbf{G}} \in \mathbb{C}^{TM\times K}$ contains the unknown antenna-based gains centred at the frequencies measured by the antenna pairs
- $\mathcal{S}\colon \mathbb{C}^K \to \mathbb{C}^{TM}$ linear operator selecting the frequencies measured by the antenna pairs

Remark: $\mathbf{y} \simeq \mathcal{S}\big(\mathcal{F}(\mathbf{x})\big) + \mathbf{w}$ is the approximated model used by state of the art imaging methods when DDEs are unknown

Prior information of the image

Compute a preprocessing step to obtain a first approximation of the image

Solve the imaging problem considering normalized DIEs (without DDEs)

Let $\mathbf{x}_0^\star$ be a solution to $\underset{\mathbf{x}\in\mathbb{R}^N}{\text{minimize}} \ \frac{1}{2}\|\mathcal{S}\big(\mathcal{F}(\mathbf{x})\big) - \mathbf{y}\|_2^2 + \widetilde{r}(\mathbf{x})$

where $\widetilde{r}\colon \mathbb{R}^N \to]-\infty, +\infty]$ is a regularization function, e.g.,

- $\widetilde{r}(\mathbf{x}) = \nu\|\boldsymbol{\Psi}^\dagger\mathbf{x}\|_1 + \iota_{[0,+\infty[^N}(\mathbf{x})$
  where $\nu > 0$ and $\boldsymbol{\Psi}^\dagger \in \mathbb{R}^{Q\times N}$ is a given sparsity basis
- ...

$\rightsquigarrow$ Can be solved with any convex optimization algorithm, e.g. forward-backward [Combettes & Wajs (2005)], primal-dual algorithms [Condat (2013), Vu (2013), Combettes & Pesquet (2012)], ...

Compute a preprocessing step to obtain a first approximation of the image

Solve the imaging problem considering normalized DIEs (without DDEs)

Let $\mathbf{x}_0^\star$ be a solution to $\underset{\mathbf{x}\in\mathbb{R}^N}{\text{minimize}} \;\; \frac{1}{2}\|\mathcal{S}\big(\mathcal{F}(\mathbf{x})\big) - \mathbf{y}\|_2^2 + \widetilde{r}(\mathbf{x})$

$\rightsquigarrow$ $\mathbf{x}_0^\star$ can be used as prior information

Prior information of the image

Compute a preprocessing step to obtain a first approximation of the image

Solve the imaging problem considering normalized DIEs (without DDEs)

Let $\mathbf{x}_0^\star$ be a solution to $\underset{\mathbf{x}\in\mathbb{R}^N}{\text{minimize}} \ \frac{1}{2}\|\mathcal{S}\big(\mathcal{F}(\mathbf{x})\big) - \mathbf{y}\|_2^2 + \widetilde{r}(\mathbf{x})$

- ~~$\mathbf{x}_0^\star$ can be used as prior information~~
- $\mathbf{x}_0^\star$ will contain artefacts (e.g. noisy background, wrong amplitude, wrong source detection, etc.)
- Use a thresholded version of $\mathbf{x}_0^\star$, denoted by $\mathbf{x}_0$, where low amplitude coefficients have been removed:

The original image can be decomposed as $\overline{\mathbf{x}} = \mathbf{x}_0 + \overline{\boldsymbol{\epsilon}}$
where $\overline{\boldsymbol{\epsilon}} \in \mathbb{R}^N$ is the error image to be estimated

Non-linear inverse problem

OBJECTIVE: Find an estimate of $\overline{\epsilon}$ and $\overline{\mathbf{U}}$, where $\overline{\mathbf{U}} = \left(\overline{\mathbf{u}_\alpha}\right)_{1 \leqslant \alpha \leqslant n_a}$, from

$$\boxed{\mathbf{y} = \Phi(\overline{\mathbf{U}}, \overline{\epsilon}) + \mathbf{w}}$$

with Φ the measurement operator associated with the RI inverse problem.

Non-linear inverse problem

OBJECTIVE: Find an estimate of $\overline{\boldsymbol{\epsilon}}$ and $\overline{\mathbf{U}}$, where $\overline{\mathbf{U}} = \left(\overline{\mathbf{u}_\alpha}\right)_{1 \leqslant \alpha \leqslant n_a}$, from

$$\mathbf{y} = \boldsymbol{\Phi}(\overline{\mathbf{U}}, \overline{\boldsymbol{\epsilon}}) + \mathbf{w}$$

with $\boldsymbol{\Phi}$ the measurement operator associated with the RI inverse problem.

- ⋆ Linear problem w.r.t. $\overline{\boldsymbol{\epsilon}}$

Non-linear inverse problem

OBJECTIVE: Find an estimate of $\overline{\epsilon}$ and $\overline{\mathbf{U}}$, where $\overline{\mathbf{U}} = \left(\overline{\mathbf{u}_\alpha}\right)_{1 \leqslant \alpha \leqslant n_a}$, from

$$\mathbf{y} = \mathbf{\Phi}(\overline{\mathbf{U}}, \overline{\epsilon}) + \mathbf{w}$$

with $\mathbf{\Phi}$ the measurement operator associated with the RI inverse problem.

- ⋆ Linear problem w.r.t. $\overline{\epsilon}$
- ⋆ Non-linear problem w.r.t. $\overline{\mathbf{U}}$

Non-linear inverse problem

OBJECTIVE: Find an estimate of $\overline{\boldsymbol{\epsilon}}$ and $\overline{\mathbf{U}}$, where $\overline{\mathbf{U}} = \left(\overline{\mathbf{u}_\alpha}\right)_{1 \leqslant \alpha \leqslant n_a}$, from

$$\mathbf{y} = \boldsymbol{\Phi}(\overline{\mathbf{U}}, \overline{\boldsymbol{\epsilon}}) + \mathbf{w}$$

with $\boldsymbol{\Phi}$ the measurement operator associated with the RI inverse problem.

- Linear problem w.r.t. $\overline{\boldsymbol{\epsilon}}$
- Non-linear problem w.r.t. $\overline{\mathbf{U}}$
  - ⇝ Use a bi-linear approach by introducing $\overline{\mathbf{U}}_1 = \overline{\mathbf{U}}_2 = \overline{\mathbf{U}}$

Problem reformulation: $\mathbf{y} = \boldsymbol{\Phi}(\overline{\mathbf{U}}_1, \overline{\mathbf{U}}_2, \overline{\boldsymbol{\epsilon}}) + \mathbf{w}$

Minimization problem

$$\boxed{\underset{\boldsymbol{\epsilon}\in\mathbb{R}^N,\,(\mathbf{U}_1,\mathbf{U}_2)\in(\mathbb{C}^{n_a\times S})^{2T}}{\operatorname{minimize}} \quad h\big(\boldsymbol{\epsilon},\mathbf{U}_1,\mathbf{U}_2\big) + r(\boldsymbol{\epsilon}) + p\big(\mathbf{U}_1,\mathbf{U}_2\big)}$$

- h is the least squares data fidelity term associated with the data model

$$h\big(\boldsymbol{\epsilon},\mathbf{U}_1,\mathbf{U}_2\big) = \frac{1}{2}\|\boldsymbol{\Phi}(\mathbf{U}_1,\mathbf{U}_2,\boldsymbol{\epsilon}) - \mathbf{y}\|^2$$

- r is the regularization term for the image

- p is the regularization term for the DDEs

Minimization problem

$$\underset{\epsilon \in \mathbb{R}^N,\, (\mathbf{U}_1, \mathbf{U}_2) \in (\mathbb{C}^{n_a \times S})^{2T}}{\text{minimize}} \quad h\big(\epsilon, \mathbf{U}_1, \mathbf{U}_2\big) + r(\epsilon) + p\big(\mathbf{U}_1, \mathbf{U}_2\big)$$

- $r(\epsilon) = \lambda \|\boldsymbol{\Psi}^\dagger(\mathbf{x}_0 + \epsilon)\|_1 + \iota_{\mathbb{E}}(\epsilon)$

  $\boldsymbol{\Psi} \in \mathbb{R}^{Q \times N}$ is a given sparsity basis

  $\lambda > 0$ is a regularization parameter

  $\mathbb{E}$ is a closed, convex and non-empty subset of $\mathbb{R}^N$ defined as
  $\mathbb{E} = \Big\{\epsilon \in \mathbb{R}^N \,\Big|\, (\forall n \in \mathbb{S}_0) \; -\vartheta x_0(n) \leqslant \epsilon(n) \leqslant \vartheta x_0(n),$
  and $(\forall n \in \mathbb{S}_0^c)\; 0 \leqslant \epsilon(n)\Big\}$
  with $\mathbb{S}_0$ the support of $\mathbf{x}_0$, $\mathbb{S}_0^c$ its complementary set, and $\vartheta \in [0, 1]$ representing the percentage error we assume on $\mathbf{x}_0$

Minimization problem

$$\underset{\epsilon \in \mathbb{R}^N,\, (\mathbf{U}_1, \mathbf{U}_2) \in (\mathbb{C}^{n_a \times S})^{2T}}{\text{minimize}} \quad h(\epsilon, \mathbf{U}_1, \mathbf{U}_2) + r(\epsilon) + p(\mathbf{U}_1, \mathbf{U}_2)$$

- $p(\mathbf{U}_1, \mathbf{U}_2) = \eta \|\mathbf{U}_1 - \mathbf{U}_2\|_2^2 + \iota_{\mathbb{D}}(\mathbf{U}_1) + \iota_{\mathbb{D}}(\mathbf{U}_2)$

  $\eta > 0$ is a regularization parameter

  $$\mathbb{D} = \big\{ \mathbf{U} \in \mathbb{C}^{T n_a \times S} \,|\, (\forall t \in \{1, \ldots, T\})(\forall \alpha \in \{1, \ldots, n_a\}) \\ \mathrm{u}_{t,\alpha}(0) \in B_\infty(1; \upsilon) \quad \text{and} \quad (\forall s \neq 0) \quad \mathrm{u}_{t,\alpha}(s) \in B_\infty(0; \upsilon) \big\}$$

Minimization problem

$$\boxed{\underset{\epsilon \in \mathbb{R}^N,\, (\mathbf{U}_1, \mathbf{U}_2) \in (\mathbb{C}^{n_a \times S})^{2T}}{\text{minimize}} \quad h\big(\epsilon, \mathbf{U}_1, \mathbf{U}_2\big) + r(\epsilon) + p\big(\mathbf{U}_1, \mathbf{U}_2\big)}$$

- Use a block coordinate forward-backward algorithm, alternating between the estimation of $\overline{\epsilon}$, $\overline{\mathbf{U}}_1$, and $\overline{\mathbf{U}}_2$
  [Bolte *et al.* (2014), Frankel *et al.* (2015), Chouzenoux *et al.* (2016)]

Block coordinate forward-backward algorithm

For $i = 0, 1, \ldots$

Choose to update either the DDEs $\big(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\big)$, or the image $\boldsymbol{\epsilon}^{(i)}$.

If the DDEs are updated:

$\mathbf{U}_1^{(i,0)} = \mathbf{U}_1^{(i)}$, $\mathbf{U}_2^{(i,0)} = \mathbf{U}_2^{(i)}$.

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_1^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\Big(\mathbf{U}_1^{(i,\ell)} - \Gamma_1^{(i)} \cdot \Big(\nabla_{\mathbf{U}_1} h\big(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i,\ell)}, \mathbf{U}_2^{(i)}\big) - \eta\big(\mathbf{U}_1^{(i,\ell)} - \mathbf{U}_2^{(i)}\big)\Big)\Big)$$

$\mathbf{U}_1^{(i+1)} = \mathbf{U}_1^{(i,L^{(i)})}$.

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_2^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\Big(\mathbf{U}_2^{(i,\ell)} - \Gamma_2^{(i)} \cdot \Big(\nabla_{\mathbf{U}_2} h\big(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i,\ell)}\big) - \eta\big(\mathbf{U}_2^{(i,\ell)} - \mathbf{U}_1^{(i+1)}\big)\Big)\Big)$$

$\mathbf{U}_2^{(i+1)} = \mathbf{U}_2^{(i,L^{(i)})}$.

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i)}$.

If the image is updated:

$\boldsymbol{\epsilon}^{(i,0)} = \boldsymbol{\epsilon}^{(i)}$.

For $j = 0, \ldots, J^{(i)} - 1$

$$\boldsymbol{\epsilon}^{(i,j+1)} = \operatorname{prox}_{\tau^{(i)} r}\Big(\boldsymbol{\epsilon}^{(i,j)} - \tau^{(i)} \nabla_{\boldsymbol{\epsilon}} h\big(\boldsymbol{\epsilon}^{(i,j)}, \mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\big)\Big)$$

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i,J^{(i)})}$.

$\big(\mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i+1)}\big) = \big(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\big)$.

Block coordinate forward-backward algorithm

For $i = 0, 1, \ldots$

Choose to update either the DDEs $(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)})$, or the image $\boldsymbol{\epsilon}^{(i)}$. (essentially cyclic updating rule)

If the DDEs are updated:

$\mathbf{U}_1^{(i,0)} = \mathbf{U}_1^{(i)}, \ \mathbf{U}_2^{(i,0)} = \mathbf{U}_2^{(i)}$.

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_1^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\left(\mathbf{U}_1^{(i,\ell)} - \Gamma_1^{(i)} \cdot \left(\nabla_{\mathbf{U}_1} h(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i,\ell)}, \mathbf{U}_2^{(i)}) - \eta(\mathbf{U}_1^{(i,\ell)} - \mathbf{U}_2^{(i)})\right)\right)$$

$\mathbf{U}_1^{(i+1)} = \mathbf{U}_1^{(i,L^{(i)})}$.

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_2^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\left(\mathbf{U}_2^{(i,\ell)} - \Gamma_2^{(i)} \cdot \left(\nabla_{\mathbf{U}_2} h(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i,\ell)}) - \eta(\mathbf{U}_2^{(i,\ell)} - \mathbf{U}_1^{(i+1)})\right)\right)$$

$\mathbf{U}_2^{(i+1)} = \mathbf{U}_2^{(i,L^{(i)})}$.

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i)}$.

If the image is updated:

$\boldsymbol{\epsilon}^{(i,0)} = \boldsymbol{\epsilon}^{(i)}$.

For $j = 0, \ldots, J^{(i)} - 1$

$$\boldsymbol{\epsilon}^{(i,j+1)} = \operatorname{prox}_{\tau^{(i)} r}\left(\boldsymbol{\epsilon}^{(i,j)} - \tau^{(i)} \nabla_{\boldsymbol{\epsilon}} h(\boldsymbol{\epsilon}^{(i,j)}, \mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)})\right)$$

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i,J^{(i)})}$.

$(\mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i+1)}) = (\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)})$.

Block coordinate forward-backward algorithm

For $i = 0, 1, \ldots$

Choose to update either the DDEs $\left(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\right)$, or the image $\boldsymbol{\epsilon}^{(i)}$.

If the DDEs are updated:

$\mathbf{U}_1^{(i,0)} = \mathbf{U}_1^{(i)}, \ \mathbf{U}_2^{(i,0)} = \mathbf{U}_2^{(i)}.$

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_1^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}} \left(\mathbf{U}_1^{(i,\ell)} - \Gamma_1^{(i)} \cdot \left(\nabla_{\mathbf{U}_1} h(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i,\ell)}, \mathbf{U}_2^{(i)}) - \eta (\mathbf{U}_1^{(i,\ell)} - \mathbf{U}_2^{(i)}) \right) \right)$$

$\mathbf{U}_1^{(i+1)} = \mathbf{U}_1^{(i,L^{(i)})}.$

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_2^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}} \left(\mathbf{U}_2^{(i,\ell)} - \Gamma_2^{(i)} \cdot \left(\nabla_{\mathbf{U}_2} h(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i,\ell)}) - \eta (\mathbf{U}_2^{(i,\ell)} - \mathbf{U}_1^{(i+1)}) \right) \right)$$

$\mathbf{U}_2^{(i+1)} = \mathbf{U}_2^{(i,L^{(i)})}.$

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i)}.$

If the image is updated:

$\boldsymbol{\epsilon}^{(i,0)} = \boldsymbol{\epsilon}^{(i)}.$

For $j = 0, \ldots, J^{(i)} - 1$

$$\boldsymbol{\epsilon}^{(i,j+1)} = \operatorname{prox}_{\tau^{(i)} r} \left(\boldsymbol{\epsilon}^{(i,j)} - \tau^{(i)} \nabla_{\boldsymbol{\epsilon}} h(\boldsymbol{\epsilon}^{(i,j)}, \mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}) \right)$$

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i,J^{(i)})}.$

$(\mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i+1)}) = (\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}).$

Block coordinate forward-backward algorithm

For $i = 0, 1, \ldots$

Choose to update either the DDEs $\left(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\right)$, or the image $\boldsymbol{\epsilon}^{(i)}$.

If the DDEs are updated:

$\mathbf{U}_1^{(i,0)} = \mathbf{U}_1^{(i)}, \; \mathbf{U}_2^{(i,0)} = \mathbf{U}_2^{(i)}.$

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_1^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\left(\mathbf{U}_1^{(i,\ell)} - \Gamma_1^{(i)} \cdot \left(\nabla_{\mathbf{U}_1} h(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i,\ell)}, \mathbf{U}_2^{(i)}) - \eta(\mathbf{U}_1^{(i,\ell)} - \mathbf{U}_2^{(i)})\right)\right)$$

$\mathbf{U}_1^{(i+1)} = \mathbf{U}_1^{(i,L^{(i)})}.$

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_2^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\left(\mathbf{U}_2^{(i,\ell)} - \Gamma_2^{(i)} \cdot \left(\nabla_{\mathbf{U}_2} h(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i,\ell)}) - \eta(\mathbf{U}_2^{(i,\ell)} - \mathbf{U}_1^{(i+1)})\right)\right)$$

$\mathbf{U}_2^{(i+1)} = \mathbf{U}_2^{(i,L^{(i)})}.$

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i)}.$

If the image is updated:

$\boldsymbol{\epsilon}^{(i,0)} = \boldsymbol{\epsilon}^{(i)}.$

For $j = 0, \ldots, J^{(i)} - 1$

$$\boldsymbol{\epsilon}^{(i,j+1)} = \operatorname{prox}_{\tau^{(i)} r}\left(\boldsymbol{\epsilon}^{(i,j)} - \tau^{(i)} \nabla_{\boldsymbol{\epsilon}} h(\boldsymbol{\epsilon}^{(i,j)}, \mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)})\right)$$

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i,J^{(i)})}.$

$\left(\mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i+1)}\right) = \left(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\right).$

Block coordinate forward-backward algorithm

For $i = 0, 1, \ldots$

Choose to update either the DDEs $\big(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\big)$, or the image $\boldsymbol{\epsilon}^{(i)}$.

If the DDEs are updated:

$\mathbf{U}_1^{(i,0)} = \mathbf{U}_1^{(i)}$, $\mathbf{U}_2^{(i,0)} = \mathbf{U}_2^{(i)}$.

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_1^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\Big(\mathbf{U}_1^{(i,\ell)} - \Gamma_1^{(i)} \cdot \Big(\nabla_{\mathbf{U}_1} h\big(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i,\ell)}, \mathbf{U}_2^{(i)}\big) - \eta\big(\mathbf{U}_1^{(i,\ell)} - \mathbf{U}_2^{(i)}\big)\Big)\Big)$$

$\mathbf{U}_1^{(i+1)} = \mathbf{U}_1^{(i,L^{(i)})}$.

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_2^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\Big(\mathbf{U}_2^{(i,\ell)} - \Gamma_2^{(i)} \cdot \Big(\nabla_{\mathbf{U}_2} h\big(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i,\ell)}\big) - \eta\big(\mathbf{U}_2^{(i,\ell)} - \mathbf{U}_1^{(i+1)}\big)\Big)\Big)$$

$\mathbf{U}_2^{(i+1)} = \mathbf{U}_2^{(i,L^{(i)})}$.

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i)}$.

If the image is updated:

$\boldsymbol{\epsilon}^{(i,0)} = \boldsymbol{\epsilon}^{(i)}$.

For $j = 0, \ldots, J^{(i)} - 1$

$$\boldsymbol{\epsilon}^{(i,j+1)} = \mathrm{prox}_{\tau^{(i)} r}\Big(\boldsymbol{\epsilon}^{(i,j)} - \tau^{(i)} \nabla_{\boldsymbol{\epsilon}} h\big(\boldsymbol{\epsilon}^{(i,j)}, \mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\big)\Big)$$

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i,J^{(i)})}$.

$\big(\mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i+1)}\big) = \big(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\big)$.

$$(\forall \widetilde{\mathbf{U}} \in \mathbb{C}^{n_a \times S}) \quad \mathcal{P}_{\mathbb{D}}(\widetilde{\mathbf{U}}) = \underset{\mathbf{U} \in \mathbb{D}}{\operatorname{argmin}} \; \|\mathbf{U} - \widetilde{\mathbf{U}}\|_2^2$$

Block coordinate forward-backward algorithm

For $i = 0, 1, \dots$

Choose to update either the DDEs $\big(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\big)$, or the image $\boldsymbol{\epsilon}^{(i)}$.

If the DDEs are updated:

$\mathbf{U}_1^{(i,0)} = \mathbf{U}_1^{(i)}, \ \mathbf{U}_2^{(i,0)} = \mathbf{U}_2^{(i)}.$

For $\ell = 0, \dots, L^{(i)} - 1$

$$\mathbf{U}_1^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\left(\mathbf{U}_1^{(i,\ell)} - \Gamma_1^{(i)} \cdot \left(\nabla_{\mathbf{U}_1} h\big(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i,\ell)}, \mathbf{U}_2^{(i)}\big) - \eta\big(\mathbf{U}_1^{(i,\ell)} - \mathbf{U}_2^{(i)}\big)\right)\right)$$

$\mathbf{U}_1^{(i+1)} = \mathbf{U}_1^{(i,L^{(i)})}.$

For $\ell = 0, \dots, L^{(i)} - 1$

$$\mathbf{U}_2^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\left(\mathbf{U}_2^{(i,\ell)} - \Gamma_2^{(i)} \cdot \left(\nabla_{\mathbf{U}_2} h\big(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i,\ell)}\big) - \eta\big(\mathbf{U}_2^{(i,\ell)} - \mathbf{U}_1^{(i+1)}\big)\right)\right)$$

$\mathbf{U}_2^{(i+1)} = \mathbf{U}_2^{(i,L^{(i)})}.$

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i)}.$

If the image is updated:

$\boldsymbol{\epsilon}^{(i,0)} = \boldsymbol{\epsilon}^{(i)}.$

For $j = 0, \dots, J^{(i)} - 1$

$$\boldsymbol{\epsilon}^{(i,j+1)} = \mathrm{prox}_{\tau^{(i)} r}\left(\boldsymbol{\epsilon}^{(i,j)} - \tau^{(i)} \nabla_{\boldsymbol{\epsilon}} h\big(\boldsymbol{\epsilon}^{(i,j)}, \mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\big)\right)$$

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i,J^{(i)})}.$

$\big(\mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i+1)}\big) = \big(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\big).$

Block coordinate forward-backward algorithm

For $i = 0, 1, \ldots$

Choose to update either the DDEs $\big(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\big)$, or the image $\boldsymbol{\epsilon}^{(i)}$.

If the DDEs are updated:

$\mathbf{U}_1^{(i,0)} = \mathbf{U}_1^{(i)}$, $\mathbf{U}_2^{(i,0)} = \mathbf{U}_2^{(i)}$.

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_1^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\left(\mathbf{U}_1^{(i,\ell)} - \Gamma_1^{(i)} \cdot \left(\nabla_{\mathbf{U}_1} h(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i,\ell)}, \mathbf{U}_2^{(i)}) - \eta(\mathbf{U}_1^{(i,\ell)} - \mathbf{U}_2^{(i)})\right)\right)$$

$\mathbf{U}_1^{(i+1)} = \mathbf{U}_1^{(i,L^{(i)})}$.

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_2^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\left(\mathbf{U}_2^{(i,\ell)} - \Gamma_2^{(i)} \cdot \left(\nabla_{\mathbf{U}_2} h(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i,\ell)}) - \eta(\mathbf{U}_2^{(i,\ell)} - \mathbf{U}_1^{(i+1)})\right)\right)$$

$\mathbf{U}_2^{(i+1)} = \mathbf{U}_2^{(i,L^{(i)})}$.

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i)}$.

If the image is updated:

$\boldsymbol{\epsilon}^{(i,0)} = \boldsymbol{\epsilon}^{(i)}$.

For $j = 0, \ldots, J^{(i)} - 1$

$$\boldsymbol{\epsilon}^{(i,j+1)} = \mathrm{prox}_{\tau^{(i)} r}\left(\boldsymbol{\epsilon}^{(i,j)} - \tau^{(i)} \nabla_{\boldsymbol{\epsilon}} h(\boldsymbol{\epsilon}^{(i,j)}, \mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)})\right)$$

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i,J^{(i)})}$.

$\big(\mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i+1)}\big) = \big(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\big)$.

$$(\forall \widetilde{\boldsymbol{\epsilon}} \in \mathbb{R}^N) \quad \mathrm{prox}_r(\widetilde{\boldsymbol{\epsilon}}) = \underset{\boldsymbol{\epsilon} \in \mathbb{R}^N}{\operatorname{argmin}} \quad r(\boldsymbol{\epsilon}) + \frac{1}{2}\|\boldsymbol{\epsilon} - \widetilde{\boldsymbol{\epsilon}}\|_2^2$$

Block coordinate forward-backward algorithm

For $i = 0, 1, \ldots$

Choose to update either the DDEs $(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)})$, or the image $\boldsymbol{\epsilon}^{(i)}$.

If the DDEs are updated:

$\mathbf{U}_1^{(i,0)} = \mathbf{U}_1^{(i)}, \ \mathbf{U}_2^{(i,0)} = \mathbf{U}_2^{(i)}$.

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_1^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\Big(\mathbf{U}_1^{(i,\ell)} - \boxed{\Gamma_1^{(i)}} \cdot \big(\nabla_{\mathbf{U}_1} h(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i,\ell)}, \mathbf{U}_2^{(i)}) - \eta(\mathbf{U}_1^{(i,\ell)} - \mathbf{U}_2^{(i)})\big)\Big)$$

$\mathbf{U}_1^{(i+1)} = \mathbf{U}_1^{(i,L^{(i)})}$.

For $\ell = 0, \ldots, L^{(i)} - 1$

$$\mathbf{U}_2^{(i,\ell+1)} = \mathcal{P}_{\mathbb{D}}\Big(\mathbf{U}_2^{(i,\ell)} - \boxed{\Gamma_2^{(i)}} \cdot \big(\nabla_{\mathbf{U}_2} h(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i,\ell)}) - \eta(\mathbf{U}_2^{(i,\ell)} - \mathbf{U}_1^{(i+1)})\big)\Big)$$

$\mathbf{U}_2^{(i+1)} = \mathbf{U}_2^{(i,L^{(i)})}$.

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i)}$.

If the image is updated:

$\boldsymbol{\epsilon}^{(i,0)} = \boldsymbol{\epsilon}^{(i)}$.

For $j = 0, \ldots, J^{(i)} - 1$

$$\boldsymbol{\epsilon}^{(i,j+1)} = \mathrm{prox}_{\tau^{(i)} r}\Big(\boldsymbol{\epsilon}^{(i,j)} - \boxed{\tau^{(i)}} \nabla_{\boldsymbol{\epsilon}} h(\boldsymbol{\epsilon}^{(i,j)}, \mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)})\Big)$$

$\boldsymbol{\epsilon}^{(i+1)} = \boldsymbol{\epsilon}^{(i,J^{(i)})}$.

$(\mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i+1)}) = (\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)})$.

- $(\forall i \in \mathbb{N})$ $\Gamma_1^{(i)}$ and $\Gamma_2^{(i)}$ $\rightsquigarrow$ preconditioning matrices
- $(\forall i \in \mathbb{N})$ $\tau^{(i)}$ $\rightsquigarrow$ step-size

Synthetic data

Simulation settings:

$n_a = 27$ antennas of the VLA telescope

each antenna pair acquires $T = 200$ snapshots

time interval of 12 hours

images of dimension $N = 256 \times 256$

DDEs: spatial Fourier support of size $S = 5 \times 5$

temporal Fourier support of size $P = 3$

Comparison between the following methods:

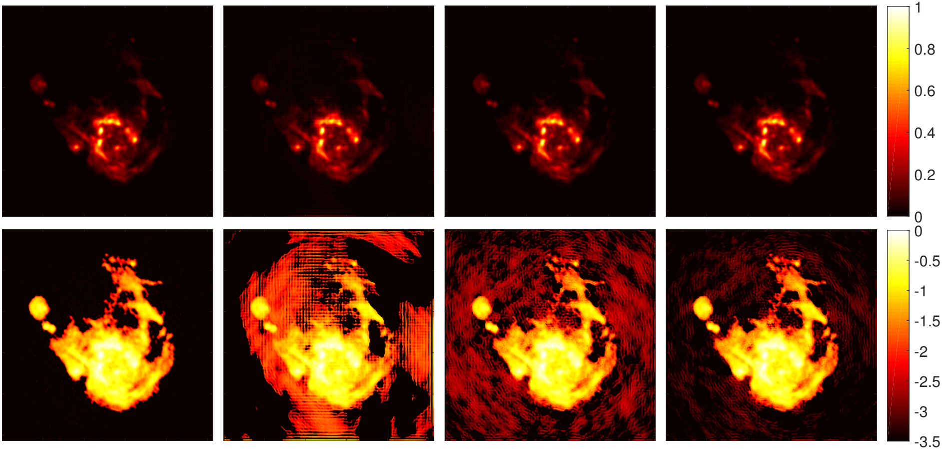
Imaging with the ground truth DDEs

Imaging with normalized DIEs

Joint imaging and DDE calibration without time regularization

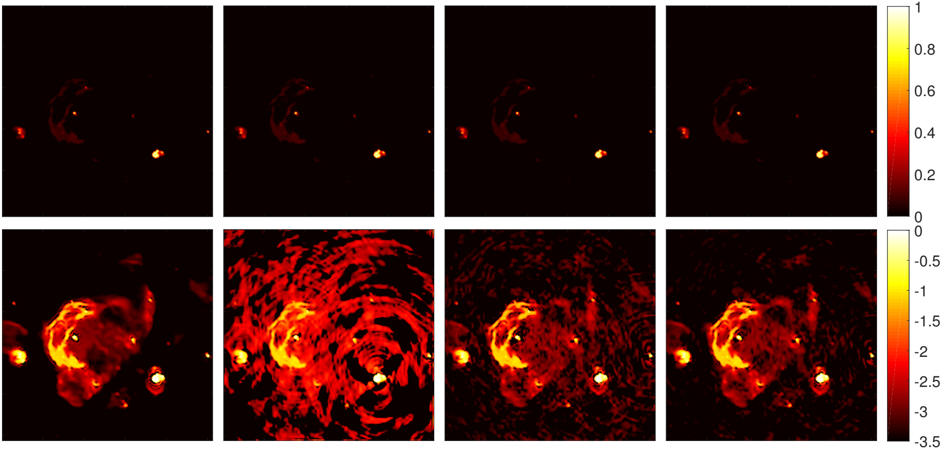
Proposed approach

Image of M31



SNR = 38.26 dB | SNR = 17.32 dB | SNR = 26.77 dB | SNR = 28.51 dB

Image of W28



SNR = 20.12 dB   SNR = 17.20 dB   SNR = 20.73 dB   SNR = 20.91 dB

Conclusions

- Joint DDE calibration and imaging model
  - DDEs are modelled as smooth images in space and time
    $\rightsquigarrow$ Estimation of the non-zero Fourier coefficients
  - Images with sophisticated structures can be considered
    $\rightsquigarrow$ Adapted choice of the regularization
- Flexible block coordinate forward backward approach
  - Essentially cyclic updating rule
  - Variable metric strategy
  - Global convergence to a critical point
  - Adapted initialization using calibration transfer
- Good performances on realistic synthetic data
- Results on real datasets from VLA telescope (ongoing work)

Thank you!

References:

- Y. Wiaux, L. Jacques, G. Puy, A. M. M. Scaife and P. Vandergheynst. **Compressed sensing imaging techniques for radio interferometry**. MNRAS, 2009.
- S. Salvini and S. J. Wijnholds. **Fast gain calibration in radio astronomy using alternating direction implicit methods: Analysis and applications**. A&A, 2014.
- O.M. Smirnov. **Revisiting the radio interferometer measurement equation. II. Calibration and direction-dependent effects**. A & A, 2015.
- E. Chouzenoux, J.-C. Pesquet and A. Repetti. **A Block Coordinate Variable Metric Forward-Backward Algorithm**. J. Global Optim., 2016.
- A. Repetti, J. Birdi, A. Dabbech and Y. Wiaux. **Non-convex optimization for self-calibration of direction-dependent effects in radio interferometric imaging**. MNRAS, 2017.

Backup slide: convergence guarantees

$$\boxed{0 < \tau^{(i)} < 1/\|\boldsymbol{\Phi}(\mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}, .)\|^2}$$

$\Gamma_1^{(i)} = (\gamma_{1,\alpha}^{(i)} \mathbf{1}_{S^2 \times P})_{1 \leqslant \alpha \leqslant n_a} \in \mathbb{R}^{S^2 \times P \times n_a}$ with $\boxed{0 < \gamma_{1,\alpha}^{(i)} < 1/(\eta + \zeta_{1,\alpha}^{(i)})}$

$\zeta_{1,\alpha}^{(i)} \rightsquigarrow$ Lipschitz constant of the partial derivative of $h(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)})$ w.r.t. $\mathbf{u}_{1,\alpha}^{(i)}$

$\Gamma_2^{(i)} = (\gamma_{2,\alpha}^{(i)} \mathbf{1}_{S^2 \times P})_{1 \leqslant \alpha \leqslant n_a} \in \mathbb{R}^{S^2 \times P \times n_a}$ with $\boxed{0 < \gamma_{2,\alpha}^{(i)} < 1/(\eta + \zeta_{2,\alpha}^{(i)})}$

$\zeta_{2,\alpha}^{(i)} \rightsquigarrow$ Lipschitz constant of the partial derivative of $h(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i+1)}, \mathbf{U}_2^{(i)})$ w.r.t. $\mathbf{u}_{2,\alpha}^{(i)}$

- ⋆ The sequence of iterates $\left(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\right)_{i \in \mathbb{N}}$ generated by the BCFB algorithm converges to a critical point $\left(\boldsymbol{\epsilon}^\star, \mathbf{U}_1^\star, \mathbf{U}_2^\star\right)$ of f
- ⋆ $\left(f\left(\boldsymbol{\epsilon}^{(i)}, \mathbf{U}_1^{(i)}, \mathbf{U}_2^{(i)}\right)\right)_{i \in \mathbb{N}}$ is a non-increasing sequence converging to $f\left(\boldsymbol{\epsilon}^\star, \mathbf{U}_1^\star, \mathbf{U}_2^\star\right)$

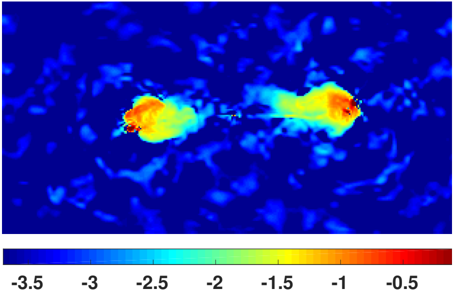
Backup slide: Observation of Cygnus A (real data)

- VLA data at 8.4 GHz with $n_a = 26$ active antennas
- $T = 625$ snapshots
- $M = 185648$ measurements

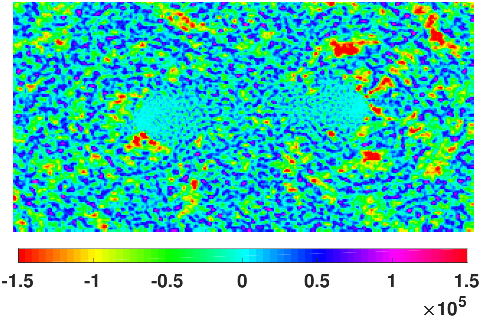
Reconstruction with normalized DIEs

$\text{SNR}_{dirty} = 44.45$ dB

$\boldsymbol{x}^\star$



Residual image: $\boldsymbol{\Phi}^\dagger(\boldsymbol{\Phi}\boldsymbol{x}^\star - \boldsymbol{y})$



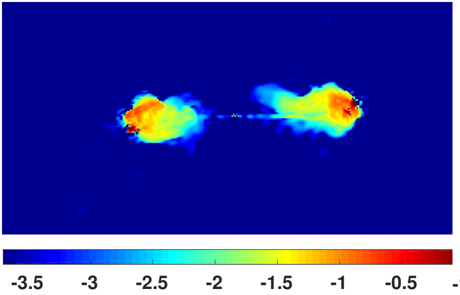
Backup slide: Observation of Cygnus A (real data)

- VLA data at 8.4 GHz with $n_a = 26$ active antennas
- $T = 625$ snapshots
- $M = 185648$ measurements

Joint imaging and DDE calibration

$\mathrm{SNR}_{dirty} = 50.21$ dB

$\boldsymbol{x}^\star$



Residual image: $\boldsymbol{\Phi}^\dagger(\boldsymbol{\Phi}\boldsymbol{x}^\star - \boldsymbol{y})$