- View through the visibilities -

Beyond CLEAN: scalable algorithms for interferometric imaging in the SKA era

Prof. Yves Wiaux[1,2]

[1]Inst. Sensors, Signals and Systems, Heriot-Watt Edinburgh, UK
[2]*Host @ Signal Processing Lab., EPFL Lausanne, Switzerland*

Interferometric imaging in the SKA era

Our previous work highlighted...

- *Convex optimisation and ℓ_1 minimisation* - Y. Wiaux, L. Jacques, G. Puy, A.M.M. Scaife, *"Compressed sensing imaging techniques for radio interferometry"*, MNRAS, 2009
- *Non-Fourier acquisition* - Y. Wiaux, G. Puy, Y. Boursier, P. Vandergheynst, *"Spread spectrum for imaging techniques in radio interferometry"*, MNRAS, 2009

- *Software* - R.E. Carrillo, J.D. McEwen, Y. Wiaux, *"PURIFY: a new approach to radio-interferometric imaging"*, MNRAS, 2014
- *Scalability* - A. Onose, R.E. Carrillo, A. Repetti, J.D. McEwen, J.-P. Thiran, J.-C. Pesquet, Y. Wiaux, *"Scalable splitting algorithms for big-data interferometric imaging in the SKA era"*, MNRAS, submitted
- *Wide-band wide-field polarised imaging, and joint calibration represent tremendous challenges...*

Interferometric imaging in the SKA era

Challenges

- Increase the resolution and sensitivity up to two orders of magnitude over current instruments

gigapixel images   huge dynamic range

Interferometric imaging in the SKA era

Challenges

- Increase the resolution and sensitivity up to two orders of magnitude over current instruments

  gigapixel images   huge dynamic range

- Unprecedented amount of data to be processed
- Good reconstruction quality with scalable algorithms employing parallel and distributed processing

Measurement model

Inverse problem

- Measurement equation

$$y(\boldsymbol{u}) = \int D(\boldsymbol{l}, \boldsymbol{u}) x(\boldsymbol{l}) e^{-2i\pi \boldsymbol{u}\cdot\boldsymbol{l}} \mathrm{d}^2\boldsymbol{l}$$

- Discretised version of the ill-posed inverse problem

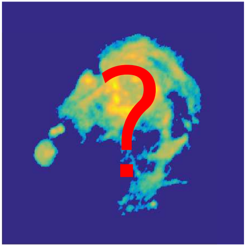
$$\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{n} \quad \text{with} \quad \boldsymbol{\Phi} = \mathbf{GF}$$

  - $\boldsymbol{x} \in \mathbb{R}_+^N$ the intensity image of interest
  - $\boldsymbol{\Phi} \in \mathbb{C}^{M\times N}$ a linear map; image domain to visibility space
  - $\boldsymbol{y} \in \mathbb{C}^M$ the measured visibilities
  - $\mathbf{G} \in \mathbb{C}^{M\times kN}$ gridding matrix modelling DDEs
  - $\mathbf{F} \in \mathbb{C}^{kN\times N}$ Fourier matrix with zero padding

Ill-posed inverse problem

SKA u–v coverage

$= \ \boldsymbol{\Phi}$



$+ \ \boldsymbol{n}$

CLEAN

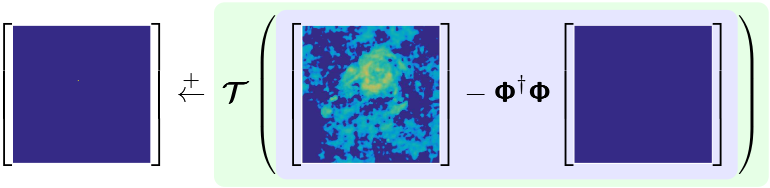
- Greedy iterative deconvolution algorithm
  - Select atoms associated with brightest pixel of residual image
  - Build the solution implicitly imposing sparsity in image space

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \boldsymbol{\mathcal{T}}\left(\boldsymbol{\Phi}^{\dagger}\left(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}^{(t-1)}\right)\right)$$

- Forward - backward *like* structure
  - Forward step on the gradient direction of the ℓ_2 norm of the residual image (major cycle)
  - Backward step with non-linear sparsity enforcing operator $\boldsymbol{\mathcal{T}}$ (minor cycle)

Traditional imaging techniques

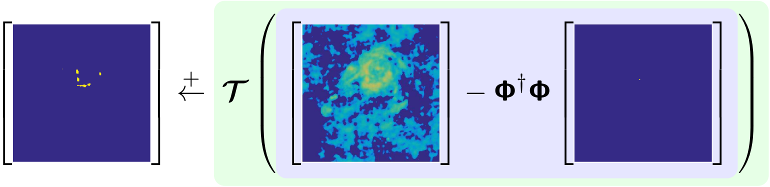
CLEAN



- Forward - backward *like* structure
  - Forward step on the gradient direction of the ℓ_2 norm of the residual image (major cycle)
  - Backward step with non-linear sparsity enforcing operator $\mathcal{T}$ (minor cycle)

Traditional imaging techniques

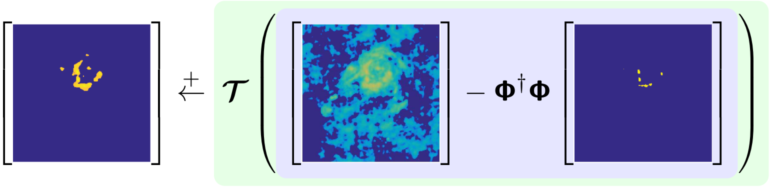
CLEAN



- Forward - backward *like* structure
  - Forward step on the gradient direction of the ℓ_2 norm of the residual image (major cycle)
  - Backward step with non-linear sparsity enforcing operator $\mathcal{T}$ (minor cycle)

Traditional imaging techniques

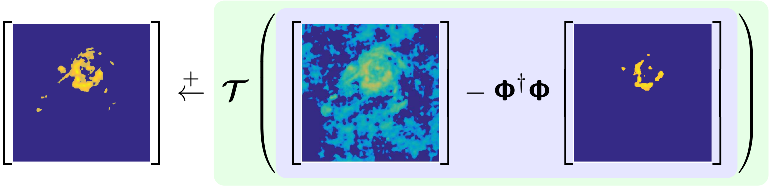
CLEAN



- Forward - backward *like* structure
  - Forward step on the gradient direction of the ℓ_2 norm of the residual image (major cycle)
  - Backward step with non-linear sparsity enforcing operator $\mathcal{T}$ (minor cycle)

Traditional imaging techniques

CLEAN



- Forward - backward *like* structure
  - Forward step on the gradient direction of the ℓ_2 norm of the residual image (major cycle)
  - Backward step with non-linear sparsity enforcing operator $\mathcal{T}$ (minor cycle)

Measurement model

Extreme data size

- Measurement equation

$$y(\boldsymbol{u}) = \int D(\boldsymbol{l}, \boldsymbol{u}) x(\boldsymbol{l}) e^{-2i\pi \boldsymbol{u}\cdot\boldsymbol{l}} \mathrm{d}^2\boldsymbol{l}$$

- Discretised version of the ill-posed inverse problem

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{n} \text{ with } \boldsymbol{\Phi} = \mathbf{GF}$$

extremely large M and N

... even more for wide-band, wide-field, polarisation data

Measurement model

Extreme data size

- Measurement equation

$$y(\boldsymbol{u}) = \int D(\boldsymbol{l}, \boldsymbol{u}) x(\boldsymbol{l}) e^{-2i\pi \boldsymbol{u}\cdot\boldsymbol{l}} \mathrm{d}^2\boldsymbol{l}$$

- Discretised version of the ill-posed inverse problem

$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{n}$ with $\boldsymbol{\Phi} = \mathbf{GF}$

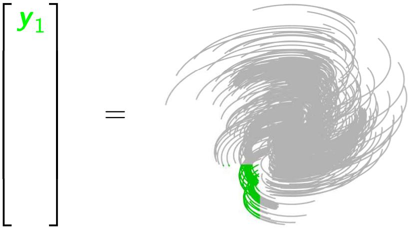
extremely large M and N

... even more for wide-band, wide-field, polarisation data

scalable algorithms, parallelisms and distributed processing

Data splitting

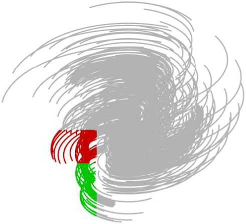
Problem formulation (1)



Data splitting

Problem formulation (1)

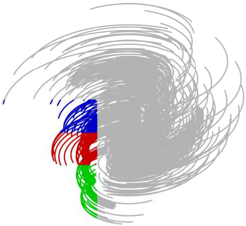
$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \\ \\ \\ \end{bmatrix} =$$



Data splitting

Problem formulation (1)

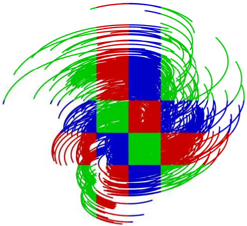
$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{y}_3 \\ \\ \\ \end{bmatrix} =$$



Data splitting

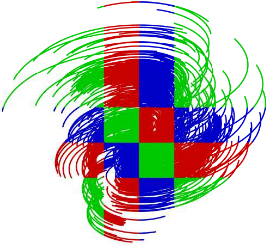
Problem formulation (1)

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \vdots \\ \mathbf{y}_d \end{bmatrix} =$$

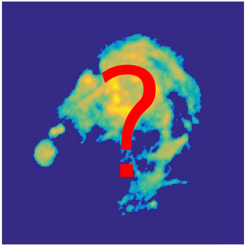


Data splitting

Problem formulation (1)



$$= \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \\ \mathbf{G}_3 \\ \vdots \\ \mathbf{G}_d \end{bmatrix} \mathbf{F}$$

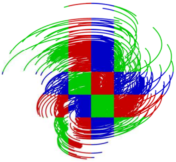


$$+ \begin{bmatrix} \boldsymbol{n}_1 \\ \boldsymbol{n}_2 \\ \boldsymbol{n}_3 \\ \vdots \\ \boldsymbol{n}_d \end{bmatrix}$$

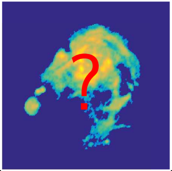
- Huge number of visibilities $\boldsymbol{y}$
  - Distribute and process the blocks independently in parallel

Data splitting

Problem formulation (1)



$$= \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \\ \mathbf{G}_3 \\ \vdots \\ \mathbf{G}_d \end{bmatrix} \mathbf{F} \left[\quad \right] + \begin{bmatrix} \boldsymbol{n}_1 \\ \boldsymbol{n}_2 \\ \boldsymbol{n}_3 \\ \vdots \\ \boldsymbol{n}_d \end{bmatrix}$$



Distributed processing nodes

CPU $\mathbf{G}_1$ — $\boldsymbol{y}_1 =$ 

CPU $\mathbf{G}_2$ — $\boldsymbol{y}_2 =$ 

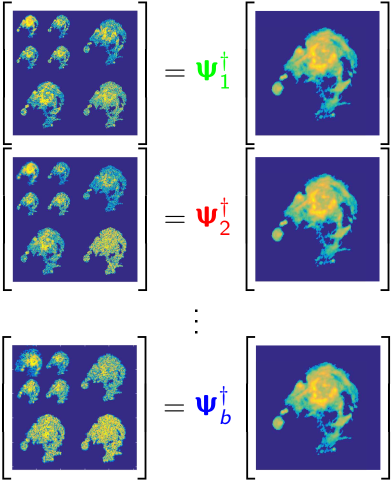
CPU $\mathbf{G}_3$ — $\boldsymbol{y}_3 =$ 

...

CPU $\mathbf{G}_d$ — $\boldsymbol{y}_d =$ 

Enforcing sparsity priors

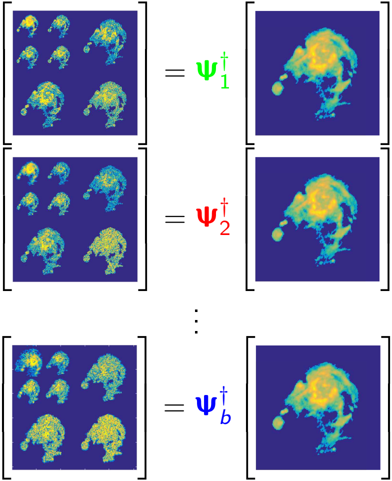
Problem formulation (2)



E.g. Average sparsity - a collection of wavelet bases to regularise the ill-posed problem, way beyond CLEAN.

Enforcing sparsity priors

Problem formulation (2)



Distributed imaging

Problem formulation (3)

- Split the large-scale inverse problem block wise

$$\boldsymbol{y}_j = \boldsymbol{\Phi}_j \boldsymbol{x} + \boldsymbol{n}_j \quad \text{with} \quad \boldsymbol{\Phi}_j = \mathbf{G}_j \mathbf{F}$$

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_d \end{bmatrix} \qquad \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_d \end{bmatrix}$$

- Regularisation of the ill-posed problem
  - Sparsity constraint for $\boldsymbol{x}$ in a collection of wavelet bases

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Psi}_1 & \dots & \boldsymbol{\Psi}_b \end{bmatrix}$$

Distributed imaging

Problem formulation (4)

- Convex optimisation task

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \gamma \sum_{i=1}^{b} l_i(\boldsymbol{\Psi}_i^\dagger \boldsymbol{x}) + \sum_{j=1}^{d} h_j(\boldsymbol{\Phi}_j \boldsymbol{x})$$

- Enforce positivity, sparsity and data fidelity

$$f(\boldsymbol{z}) = \iota_{\mathcal{C}}(\boldsymbol{z}), \mathcal{C} = \mathbb{R}_+^N$$

$$l_i(\boldsymbol{z}) = \|\boldsymbol{z}\|_1$$

$$h_j(\boldsymbol{z}) = \iota_{\mathcal{B}_j}(\boldsymbol{z}), \ \ \mathcal{B}_j = \{\boldsymbol{z} \in \mathbb{C}^{M_j} : \|\boldsymbol{z} - \boldsymbol{y}_j\|_2 \leq \epsilon_j\}$$

The primal dual approach

- Primal problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \gamma \sum_{i=1}^{b} l_i(\boldsymbol{\Psi}_i^\dagger \boldsymbol{x}) + \sum_{j=1}^{d} h_j(\boldsymbol{\Phi}_j \boldsymbol{x})$$

- Dual formulation of the original convex optimisation task

$$\min_{\substack{\boldsymbol{u}_i \\ \boldsymbol{v}_j}} f^*\left(-\sum_{i=1}^{b} \boldsymbol{\Psi}_i \boldsymbol{u}_i - \sum_{j=1}^{d} \boldsymbol{\Phi}_j^\dagger \boldsymbol{v}_j \right) + \frac{1}{\gamma} \sum_{i=1}^{b} l_i^*(\boldsymbol{u}_i) + \sum_{j=1}^{d} h_j^*(\boldsymbol{v}_j)$$

- Primal dual algorithm
  - Alternate solving the primal problem and the dual problem
  - Converges towards a Kuhn-Tucker point

Distributed imaging

Advantages of the primal dual approach

- Full splitting of the operators and functions (versus ADMM)
  - No inversion of the linear operators
  - The updates are performed on the dual variables in parallel
- Interlaced and parallel CLEAN-like iteration structure
  - Forward-backward iterations are applied in parallel for all dual variables in data, sparsity, and image space
- Randomised updates on the dual variables
  - Reduces computational and memory needs per iteration
  - Requires more iterations to converge

Distributed imaging

Primal dual algorithm

given $\boldsymbol{x}^{(0)}, \tilde{\boldsymbol{x}}^{(0)}, \boldsymbol{u}_i^{(0)}, \boldsymbol{v}_j^{(0)}, \tilde{\boldsymbol{u}}_i^{(0)}, \tilde{\boldsymbol{v}}_j^{(0)}, \gamma, \tau, \sigma_i$

repeat for $t = 1, \ldots$

generate sets $\mathcal{P} \subset \{1, \ldots, b\}$ **and** $\mathcal{D} \subset \{1, \ldots, d\}$

$$\boldsymbol{b}_j^{(t)} = \mathbf{M}_j \mathbf{F} \mathbf{Z} \tilde{\boldsymbol{x}}^{(t-1)}, \quad \forall j \in \mathcal{D}$$

run simultaneously

$\forall j \in \mathcal{D}$ **distribute** $\boldsymbol{b}_j^{(t)}$ **and do in parallel**

$$\boldsymbol{v}_j^{(t)} = \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{P}}_{\mathcal{B}_j}\right)\left(\boldsymbol{v}_j^{(t-1)} + \mathbf{G}_j \boldsymbol{b}_j^{(t)}\right)$$

$$\tilde{\boldsymbol{v}}_j^{(t)} = \mathbf{G}_j^* \boldsymbol{v}_i^{(t)}$$

"CLEAN $\tilde{\boldsymbol{v}}_j$"

end and gather $\tilde{\boldsymbol{v}}_j^{(t)}$

$\forall i \in \mathcal{P}$ **do in parallel**

$$\boldsymbol{u}_i^{(t)} = \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{S}}_{\frac{\gamma}{\sigma_i}}\right)\left(\boldsymbol{u}_i^{(t-1)} + \boldsymbol{\Psi}_i^* \tilde{\boldsymbol{x}}^{(t)}\right)$$

$$\tilde{\boldsymbol{u}}_i^{(t)} = \boldsymbol{\Psi}_i \boldsymbol{u}_i^{(t)}$$

"CLEAN $\tilde{\boldsymbol{u}}_i$"

end

end

$$\bar{\boldsymbol{x}}^{(t)} = \boldsymbol{\mathcal{P}}_{\mathcal{C}}\left(\boldsymbol{x}^{(t-1)} - \tau\left(\sum_{i=1}^{b} \sigma_i \tilde{\boldsymbol{u}}_i^{(t)} + \mathbf{Z}^* \mathbf{F}^\dagger \sum_{j=1}^{d} \varsigma_j \mathbf{M}_j^* \tilde{\boldsymbol{v}}_j^{(t)}\right)\right)$$

"CLEAN $\tilde{\boldsymbol{x}}$"

$$\tilde{\boldsymbol{x}}^{(t)} = 2\bar{\boldsymbol{x}}^{(t)} - \boldsymbol{x}^{(t-1)}$$

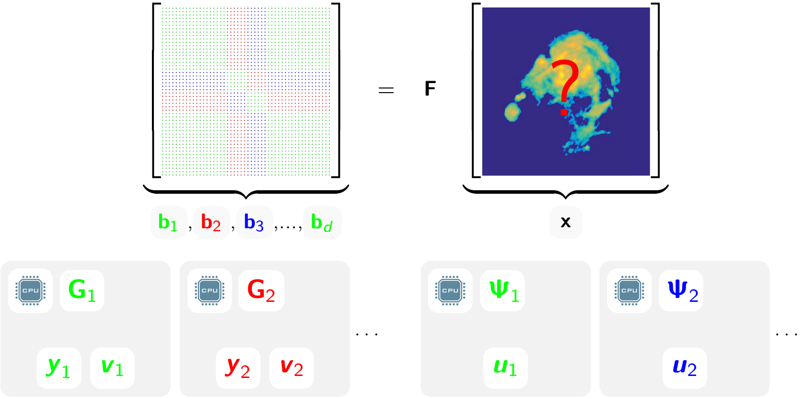
until convergence

BASP Biomedical and Astronomical Signal Processing

HERIOT WATT UNIVERSITY

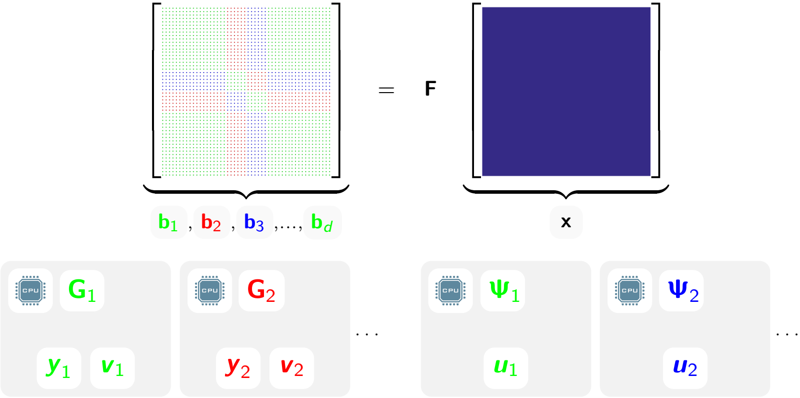
The primal dual algorithm

Data distribution and parallel processing



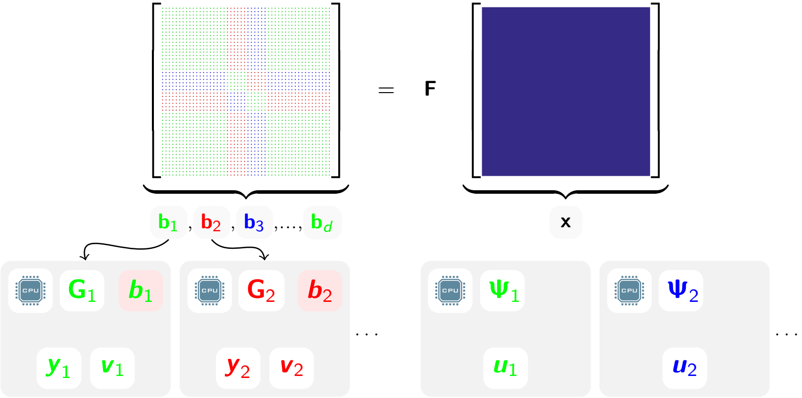
The primal dual algorithm

Data distribution and parallel processing



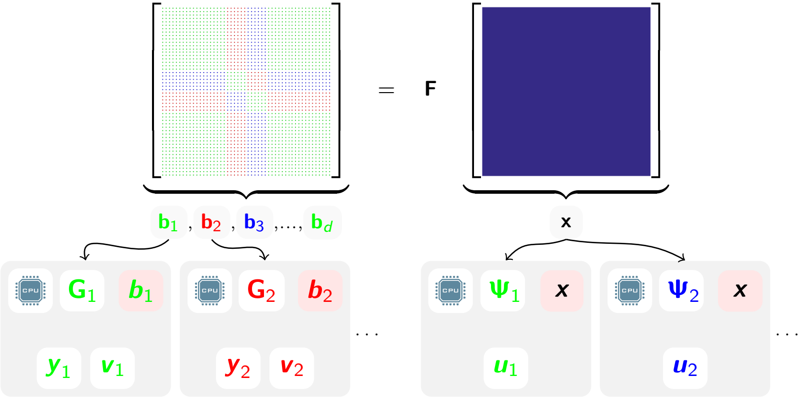
The primal dual algorithm

Data distribution and parallel processing



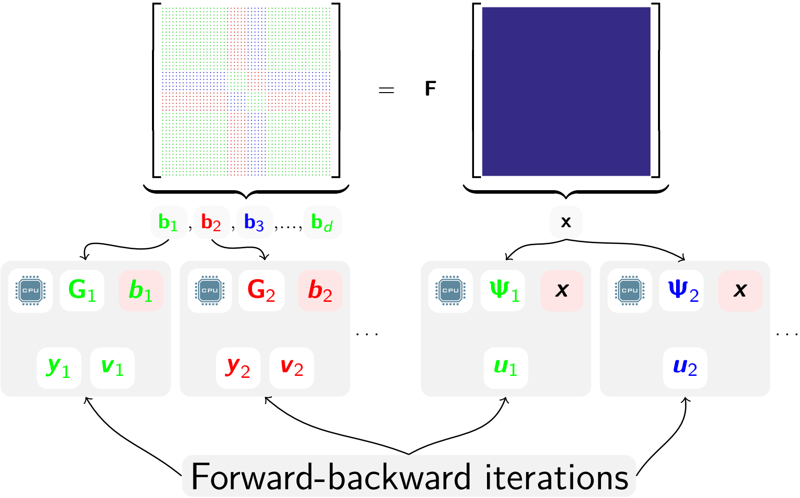
The primal dual algorithm

Data distribution and parallel processing



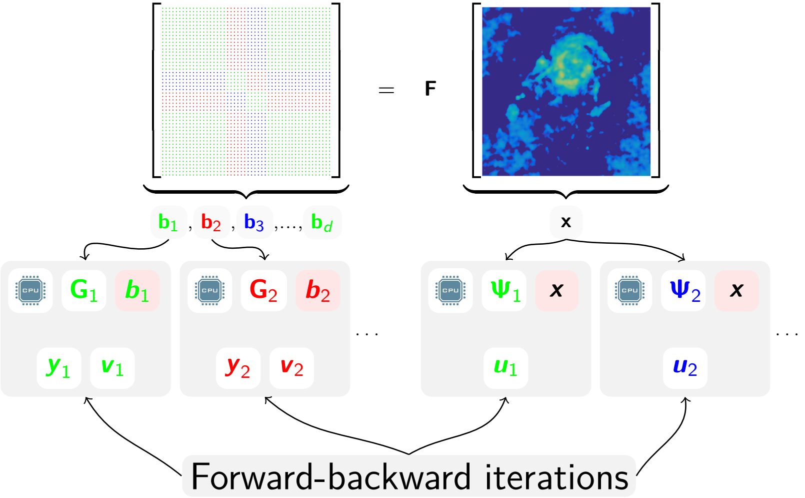
The primal dual algorithm

Data distribution and parallel processing



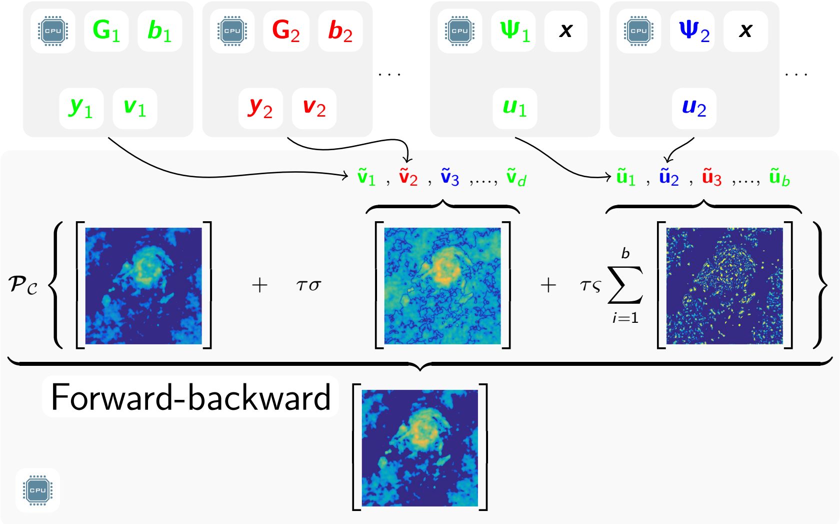
The primal dual algorithm

Data distribution and parallel processing



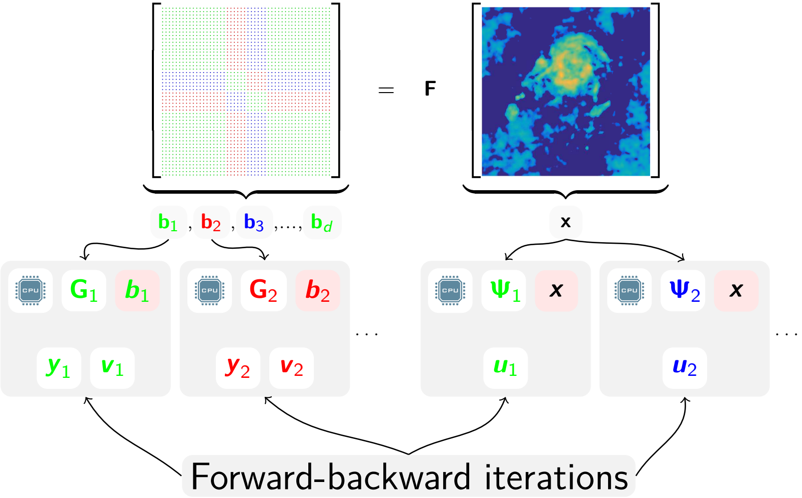
The primal dual algorithm

Image update



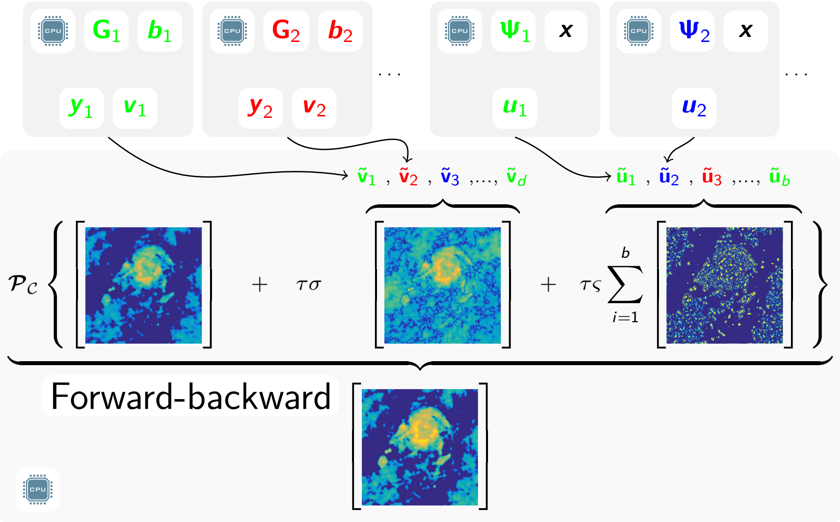
The primal dual algorithm

Data distribution and parallel processing



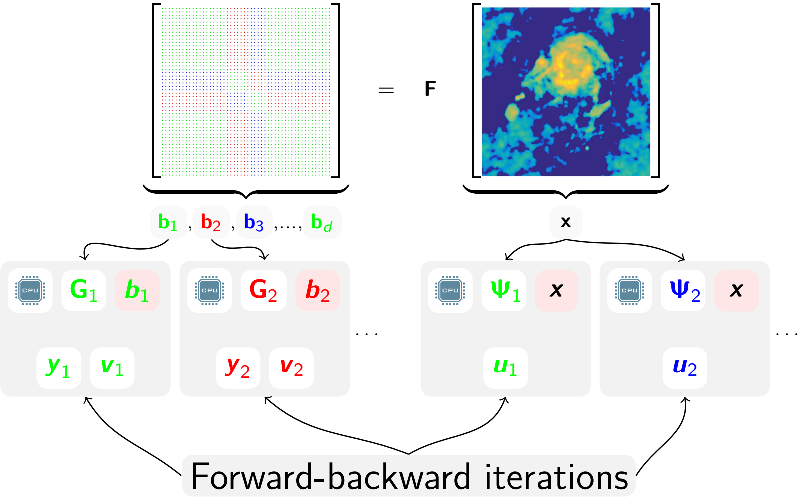
The primal dual algorithm

Image update



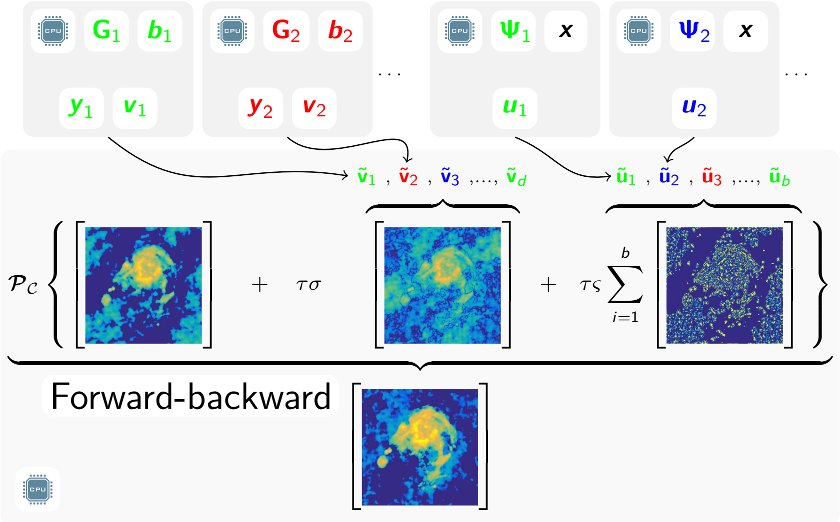
The primal dual algorithm

Data distribution and parallel processing



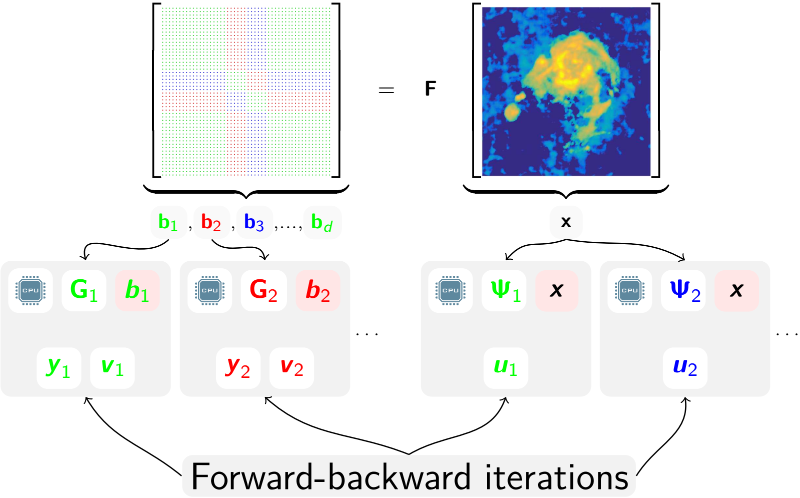
The primal dual algorithm

Image update



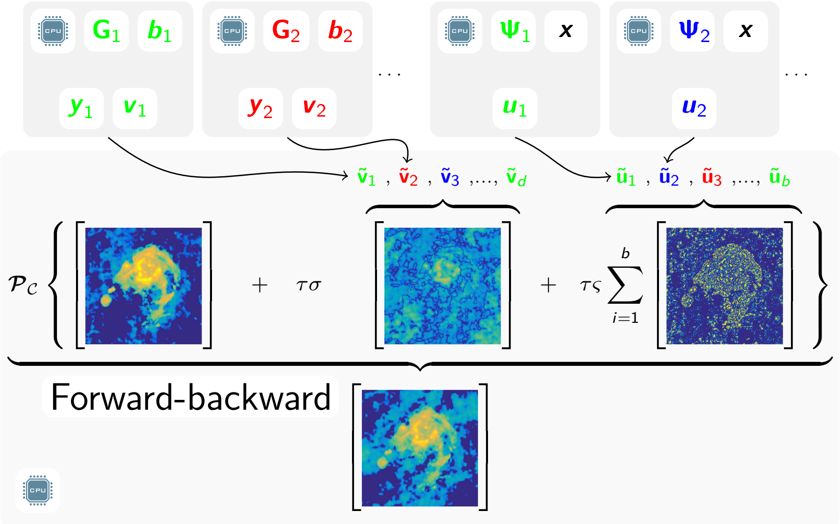
The primal dual algorithm

Data distribution and parallel processing



The primal dual algorithm

Image update



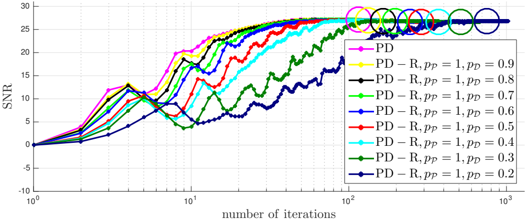
Performance illustration

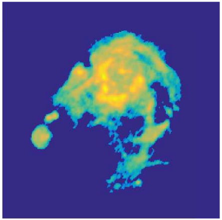
Recovery example using SKA coverage

- SKA coverage
  - Input SNR = 40dB; Reconstruction SNR = 26dB; DR ≈ 900

Performance illustration

Randomisation





- Incomplete gaussian coverage; input SNR = 20dB
- Lower complexity per iteration but require more iterations
- Overall the computational load does not increase significantly