From the modeling of direct problems in image processing to the resolution of inverse problems

Caroline Chaux
Joint work with M. Krémé, V. Emiya, B. Torrésani
Joint work with X. Vu, N. Thirion-Moreau, S. Maire

CNRS and Aix-Marseille Univ.

Contents

From modeling to resolution

$$\boxed{\begin{array}{c}\text{Object of interest}\\ y\end{array}} \longrightarrow \boxed{\begin{array}{c}\text{System}\\ \mathcal{D}_\alpha\end{array}} \longrightarrow \boxed{\begin{array}{c}\text{Observation}\\ z = \mathcal{D}_\alpha(y)\end{array}}$$

From modeling to resolution

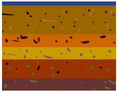
α - system parameters (e.g. noise)

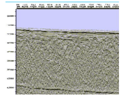
$$\boxed{\begin{array}{c}\text{Object of interest}\\ y\end{array}} \longrightarrow \boxed{\begin{array}{c}\text{System}\\ \mathcal{D}_\alpha\end{array}} \longrightarrow \boxed{\begin{array}{c}\text{Observation}\\ z = \mathcal{D}_\alpha(y)\end{array}}$$

From modeling to resolution

α - system parameters (e.g. noise)

Object of interest y → System $\mathcal{D}_\alpha$ → Observation $z = \mathcal{D}_\alpha(y)$

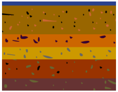


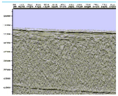


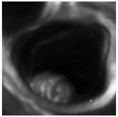
From modeling to resolution

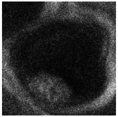
α - system parameters (e.g. noise)

Object of interest y	$\longrightarrow$	System $\mathcal{D}_\alpha$	$\longrightarrow$	Observation $z = \mathcal{D}_\alpha(y)$





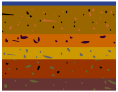


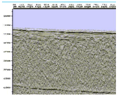
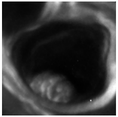


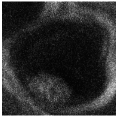
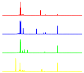
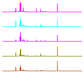
From modeling to resolution

α - system parameters (e.g. noise)

Object of interest y → System $\mathcal{D}_\alpha$ → Observation $z = \mathcal{D}_\alpha(y)$



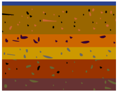



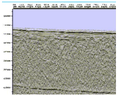
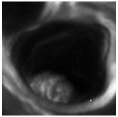





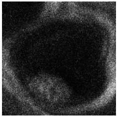
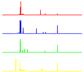
From modeling to resolution

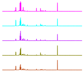
α - system parameters (e.g. noise)

Object of interest y	$\rightarrow$	System $\mathcal{D}_\alpha$	$\rightarrow$	Observation $z = \mathcal{D}_\alpha(y)$



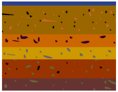


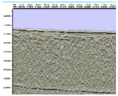
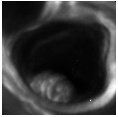
How? $\rightarrow$ Solving inverse problems

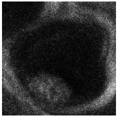
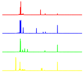
From modeling to resolution

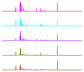
α - system parameters (e.g. noise)

Object of interest y → System $\mathcal{D}_\alpha$ → Observation $z = \mathcal{D}_\alpha(y)$





method parameters (e.g. regularization)

How? → Solving inverse problems

Inverse problem formulation

What?

Recovering the original (unknown data) from distorded observations.

How?

Formulating the inverse problem as a minimization problem

- Variational approach;
- Statistical approach (MAP).

And so

$$\underset{y}{\text{minimize}} \quad \underbrace{f_1(y)}_{\text{Fidelity}} \quad + \quad \underbrace{f_2(y)}_{\text{Regularization}}$$

Minimization problems

- Standard problem:

$$\underset{y\in\mathbb{R}^N}{\text{minimize}} \quad \underbrace{f_1(y)}_{\text{Fidelity}} \quad + \quad \underbrace{f_2(y)}_{\text{Regularization}} \quad .$$

- Taking into account several regularizations ($P-1$ terms):

$$\underset{y\in\mathbb{R}^N}{\text{minimize}} \; f_1(y) + \sum_{p=2}^{P} f_p(y).$$

- Introducing linear operators $(F_p)_{p\in\{1,\dots,P\}}$:

$$\underset{y\in\mathbb{R}^N}{\text{minimize}} \; \sum_{p=1}^{P} f_p(F_p y).$$

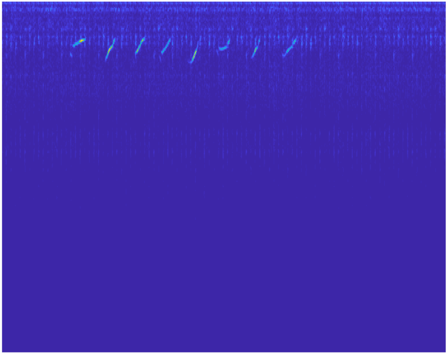
- For large size problem or for other reasons, can be interesting to work on data blocks $y^{(p)}$ of size L_p ($y = (y^{(p)})_{p=1}^{P}$)

$$\underset{y\in\mathbb{R}^N}{\text{minimize}} \; \sum_{p=1}^{P} f_p(y^{(p)}).$$

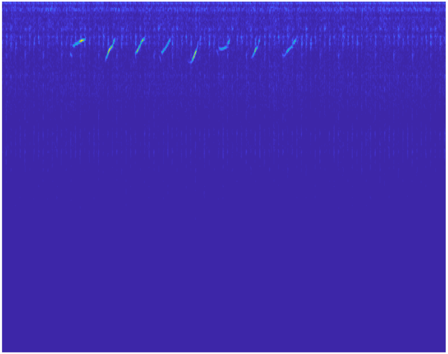
Some proximal approaches

- *Parallel ProXimal Algorithm + (PPXA+)* [Pesquet, Pustelnik, 2012]
- Generalized Forward-Backward [Raguet et al., 2012]
- M+SFBF [Briceño-Arias, Combettes, 2011]
- M+LFBF [Combettes, Pesquet, 2011]
- FB based algorithms [Chambolle, Pock, 2011],[Vũ,2013],[Condat,2013]
- Proximal Alternating Linearized Minimization (PALM) [Bolte et al., 2014]
- An accelerated projection gradient based algorithm [Zhang et al., 2016]
- Block-Coordinate Variable Metric Forward-Backward (BC-VMFB) algorithm [Chouzenoux et al., 2016]

Motivation



Motivation



Inpainting problem

$$\min_{\mathbf{Y}\in\mathbb{C}^{F\times T}} \frac{1}{2}\|\mathbf{M}\odot(\mathbf{X}-\mathbf{Y})\|_F^2+\lambda\|\mathbf{Y}\|_*, \text{ where } \lambda>0.$$

Initial point

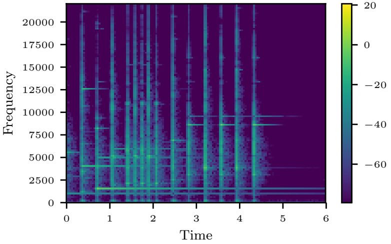


Figure: Spectrogram of the *Glockenspiel*, composed of about 50 spectral peaks distributed on 15 occurrences of 8 notes.

- How the intuitions of the low-rankness of the spectrograms can be extended to complex-valued time-frequency matrices ?
- What is a rank-one matrix, or more generally a rank-r matrix, in the time-frequency plane?
- Do time-frequency matrices of real-world sounds have good low-rank approximations?

STFT definitions

$(K \times N)$-STFT, band-pass convention

$$\mathbf{S}_{\mathrm{BP}}^{(K\times N)}\left[k,n\right] = \sum_{m} \mathbf{s}\left[t_n + m\right] \mathbf{w}\left[m\right] e^{-2i\pi\nu_k m}$$

$(K \times N)$-STFT, low-pass convention

$$\mathbf{S}_{\mathrm{LP}}^{(K\times N)}\left[k,n\right] = \sum_{m} \mathbf{s}\left[m\right] \mathbf{w}\left[m - t_n\right] e^{-2i\pi\nu_k m}.$$

where $\left(\mathbf{w}\left[m\right]\right)_{m\in[\![L]\!]} \in \mathbb{C}^L$ denotes the window, $\nu_k, k \in [\![K]\!]$ is a discrete frequency and $t_n, n \in [\![N]\!]$ a discrete time.

Relation between conventions

$\forall k \in [\![K]\!], n \in \mathbb{Z}, \mathbf{S}_{\mathrm{LP}}\left(k,n\right) = \mathbf{S}_{\mathrm{BP}}\left(k,n\right) \times e^{-2i\pi\nu_k t_n}$

Factorization of STFT matrices

$(L \times L)$-STFT (full redundancy $K = L = N$)

$$\forall k, n, \mathbf{S}_{\mathrm{BP}}[k, n] = \sum_m \mathbf{s}[n+m]\,\mathbf{w}[m]\,e^{-2i\pi\frac{km}{L}}$$

$$\forall k, n, \mathbf{S}_{\mathrm{LP}}[k, n] = \sum_m \mathbf{s}[m]\,\mathbf{w}[m-n]\,e^{-2i\pi\frac{km}{L}}.$$

For any signal $\mathbf{s} \in \mathbb{C}^L$ and window $\mathbf{w} \in \mathbb{C}^L$, we have

$$\mathbf{S}_{\mathrm{BP}} = \mathbf{E}\,\mathrm{diag}(\mathbf{w})\,\mathbf{E}^{-1}\,\mathrm{diag}(\widehat{\mathbf{s}})\,\mathbf{E}$$

and

$$\mathbf{S}_{\mathrm{LP}} = \mathbf{E}\,\mathrm{diag}(\mathbf{s})\,\mathbf{E}^{-1}\,\mathrm{diag}(\widehat{\mathbf{w}})\,\mathbf{E}$$

where

$$\boxed{\mathbf{E} = \left(e^{-2i\pi\frac{kt}{L}}\right)_{k\in[\![L]\!],t\in[\![L]\!]}}$$

Rank-r STFT matrices

Band-pass convention

If $\mathbf{w} \in \mathbb{C}^L$ is a window that does not vanish, i.e., $\forall k \in [\![L]\!], \mathbf{w}[k] \neq 0$,
then $\operatorname{rank}(\mathbf{S}_{\mathrm{BP}}) = \|\widehat{\mathbf{s}}\|_0$.

$\Rightarrow$ *The set of rank-r STFT matrices in the band-pass convention is composed of the signals that are a sum of r pure complex exponentials at Fourier frequencies.*

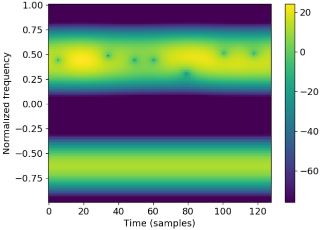
Low-pass convention

If $\mathbf{w} \in \mathbb{C}^L$ is a window such that $\widehat{\mathbf{w}}$ does not vanish, i.e., $\forall k \in [\![L]\!], \widehat{\mathbf{w}}[k] \neq 0$,
then $\operatorname{rank}(\mathbf{S}_{\mathrm{LP}}) = \|\mathbf{s}\|_0$.

$\Rightarrow$ *The set of rank-r STFT matrices in the low-pass convention is composed of the signals that are a sum r diracs at integer times.*

Analysis of low-rank STFT matrices

Context : Signal with length $L = 128$ composed of a sum of $N_c = 6$ complex sinusoids at exact Fourier frequencies (5 closed frequencies and 1 isolated frequency).



Results : $\text{rank}\,(\mathbf{S}_{\text{BP}}) = Nc$ while $\text{rank}\,(\mathbf{S}_{\text{LP}})$ is higher.

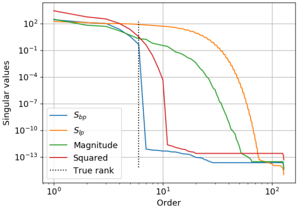


Figure: Analysis with a Gaussian window: singular values of STFT matrices, magnitude and energy spectrograms.

Analysis of low-rank STFT matrices

Context: rank vs. number of components N_c (frequencies drawn randomly at exact Fourier frequencies), signal length $L = 64$.

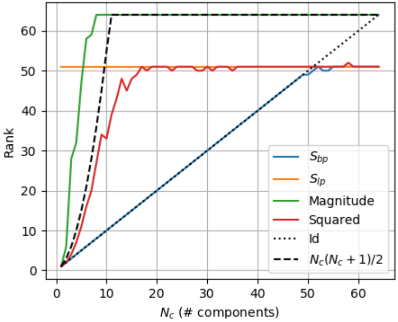


Figure: Rank of several types of time-frequency matrices vs. number of sinusoids in the signal.

Results: $\mathrm{rank}\left(\mathbf{S}_{\mathrm{BP}}\right) = N_c$ while $\mathrm{rank}\left(\mathbf{S}_{\mathrm{LP}}\right)$ is higher.

rg STFT matrix $<$ related spectrograms.

Formulation of phase inpainting problem

Gabor atoms (STFT): $\mathbf{a}_{t,\nu} = \mathbf{w}[n - th]e^{2\imath\pi\frac{\nu}{F}n}$ for $\begin{cases} t \in \{0, \dots, T-1\} \\ \nu \in \{0, \dots, F-1\} \\ \mathbf{w} : \text{window} \\ h : \text{hop size} \end{cases}$

Known binary mask: $\boldsymbol{m} \in \{0, 1\}^{F \times T}$

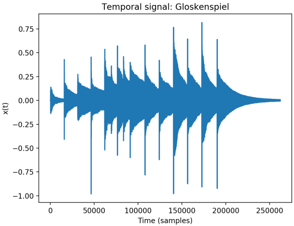
Observations: $\mathbf{b} \in \mathbb{C}^{F \times T}$ $\begin{cases} \mathbf{b}(\boldsymbol{m}) \text{ fully known ceofficients} \\ \mathbf{b}(\neg\boldsymbol{m}) \text{ known magnitudes} \end{cases}$

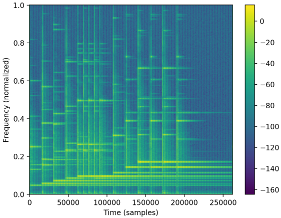
Proposition

$$\textit{Find } \mathbf{x} \in \mathbb{C}^N \textit{ s.t. } \begin{cases} \langle \mathbf{x}, \mathbf{a}_{t,\nu} \rangle & = \mathbf{b}[t, \nu], \forall t, \nu \in \operatorname{supp}(\boldsymbol{m}) \ (\textit{our contribution}) \\ |\langle \mathbf{x}, \mathbf{a}_{t,\nu} \rangle| & = \mathbf{b}[t, \nu], \forall t, \nu \in \operatorname{supp}(\neg\boldsymbol{m}) \end{cases}$$

STFT with some missing data

- Missing data in TF plane = Missing phases ie magnitudes assumed to be known.
- What about the quality of the reconstructed signal with 30% of missing phases in its spectrogram ?





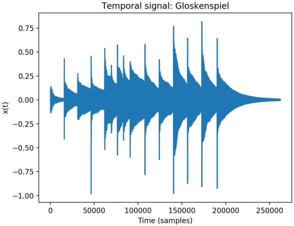
Original glockenspiel Reconstructed signal

- What about putting random phases ?

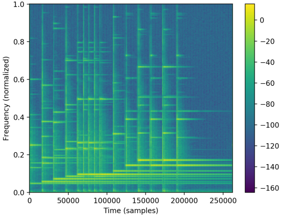
RPI reconstruction:

STFT with some missing data

- Missing data in TF plane = Missing phases ie magnitudes assumed to be known.
- What about the quality of the reconstructed signal with 30% of missing phases in its spectrogram ?



Original glockenspiel



Reconstructed signal

- What about putting random phases ?

RPI reconstruction:

- $\boxed{\text{Phases are very important}}$

Algorithm 1 Griffin and Lim for phase inpainting (GLI)

Require: $\begin{cases} \mathbf{b} : \text{observations} , \boldsymbol{m} : \text{binary mask} \\ n_{\text{iter}} : \text{number of iterations} \\ \text{STFT}_{\mathbf{x}} \text{ and } \text{STFT}^{-1} : \text{operators related to } \{a_{t,\nu}\} \end{cases}$

Random initialization φ_0 of missing phases:

$$\varphi \leftarrow \boldsymbol{m} \circ \angle \mathbf{b} + (1 - \boldsymbol{m}) \circ \varphi_0 \quad \text{and} \quad \mathbf{y}^{(0)} \leftarrow \mathbf{b} \circ \exp(\imath \varphi)$$

for $i \in \{1, 2, \ldots, n_{\text{iter}}\}$ **do**

$\mathbf{z}^{(i)} \leftarrow \text{STFT}_{\mathbf{x}} \left(\text{STFT}^{-1} \left(\mathbf{y}^{(i-1)} \right) \right)$

$\varphi^{(i)} \leftarrow \boldsymbol{m} \circ \angle \mathbf{b} + (1 - \boldsymbol{m}) \circ \angle \mathbf{z}^{(i)}$

$\mathbf{y}^{(i)} \leftarrow \mathbf{b} \circ \exp(\imath \varphi^{(i)})$

end for

return $\text{STFT}^{-1}(\mathbf{y}^{(n_{\text{iter}})})$

Original signal: RPI reconstruction: GLI reconstruction:

PhaseLift for phase inpainting (PLI)

Phase inpainting problem (non-convex)

$$\text{Find } \mathbf{x} \in \mathbb{C}^N \text{ s.t. } \begin{cases} \langle \mathbf{x}, \mathbf{a}_{t,\nu} \rangle & = \mathbf{b}[t,\nu], \forall t,\nu \in \text{supp}\,(\boldsymbol{m}) \\ |\langle \mathbf{x}, \mathbf{a}_{t,\nu} \rangle| & = \mathbf{b}[t,\nu], \forall t,\nu \in \text{supp}\,(\neg \boldsymbol{m}) \end{cases}$$

$$\Big\downarrow \ \textbf{lifting} \quad \mathbf{X} = \mathbf{x}\mathbf{x}^H$$

$$\min_{\mathbf{X} \in \mathbb{C}^{N \times N}} \text{Rank}(\mathbf{X}) \text{ s.t. } \begin{cases} \text{Trace}(\mathbf{A}_{(t,\nu),(t',\nu')}\mathbf{X}) = \mathbf{b}[t',\nu']\bar{\mathbf{b}}[t,\nu], & \forall\ (t,\nu) \in \text{supp}(\boldsymbol{m}) \\ \text{Trace}(\mathbf{A}_{(t',\nu')(t',\nu')}\mathbf{X}) = \mathbf{b}^2[t',\nu'], & \forall\ (t',\nu') \in \text{supp}\,(\neg \boldsymbol{m}) \\ \mathbf{X} \succeq 0 \quad \text{(positive semidefinite matrix (PSD))} \end{cases}$$

$$\Big\downarrow \ \textbf{relaxation}$$

$$\min_{\mathbf{X} \in \mathbb{C}^{N \times N}} \text{Trace}(\mathbf{X}) \text{ s.t. } \begin{cases} \text{Trace}(\mathbf{A}_{(t,\nu),(t',\nu')}\mathbf{X}) = \mathbf{b}[t',\nu']\bar{\mathbf{b}}[t,\nu], & \forall\ (t,\nu) \in \text{supp}(\boldsymbol{m}) \\ \text{Trace}(\mathbf{A}_{(t',\nu')(t',\nu')}\mathbf{X}) = \mathbf{b}^2[t',\nu'], & \forall\ (t',\nu') \in \text{supp}(\neg \boldsymbol{m}) \\ \mathbf{X} \succeq 0 \end{cases}$$

1

[1] TFOCS: Templates for convex cone problems with applications to sparse signal recovery, S. Becker , E.J. Candès and M. Grant, 2010.

PhaseCut for phase inpainting (PCI)

$$\text{Find } \mathbf{x} \in \mathbb{C}^N \text{ s.t. } \begin{cases} \langle \mathbf{x}, \mathbf{a}_{t,\nu} \rangle & = \mathbf{b}[t,\nu], \forall t,\nu \in \operatorname{supp}(\boldsymbol{m}) \\ |\langle \mathbf{x}, \mathbf{a}_{t,\nu} \rangle| & = \mathbf{b}[t,\nu], \forall t,\nu \in \operatorname{supp}(\neg \boldsymbol{m}) \end{cases}$$

$$\Big\downarrow \textbf{splitting}$$

$$\text{Find } \mathbf{x} \in \mathbb{C}^N, \mathbf{u} \in \mathbb{C}^{F\times T} \text{ s.t. } \begin{cases} \mathbf{A}\mathbf{x} & = \operatorname{Diag}(\mathbf{c})\mathbf{u} \\ \mathbf{u}[t,\nu] & = e^{\imath \angle \mathbf{b}[t,\nu]} \ \forall t,\nu \in \operatorname{supp}(\boldsymbol{m}) \\ |\mathbf{u}[t,\nu]| & = 1 \ \forall t,\nu \end{cases}$$

$$\Big\downarrow \textbf{lifting and relaxation}$$

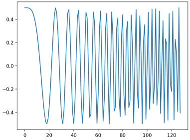
$$\min_{\mathbf{U} \in \mathbb{C}^{(F\times T)^2}} \operatorname{Trace}(\mathbf{U}\boldsymbol{\Gamma}) \text{ s.t. } \begin{cases} \operatorname{Diag}(\mathbf{U}) = \mathbb{1} \\ \mathbf{U}[(t,\nu),(t',\nu')] = \frac{\mathbf{b}[t,\nu]}{|\mathbf{b}[t,\nu]|} \frac{\bar{\mathbf{b}}[t',\nu']}{|\mathbf{b}[t',\nu']|}, \forall (t,\nu),(t',\nu') \in \operatorname{supp}(\boldsymbol{m}) \\ \mathbf{U} \succeq 0 \end{cases}$$

2

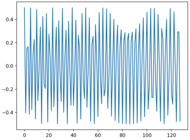
[2] Block coordinate descent methods for semidefinite programming. Wen, Z. and Goldfarb, D. and Scheinberg, K. 2012.

Data

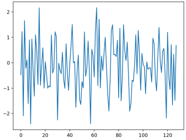
- A signal composed of a mixture of 3 signals: s = (a) +(b) + (c)



(a) chirp $\nearrow$

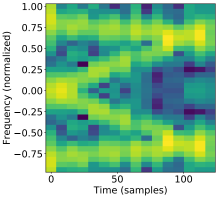


(b) Chirp $\searrow$



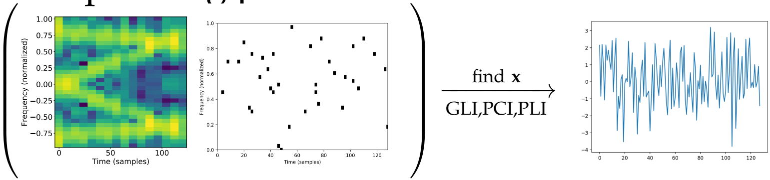
(c)

$\|$

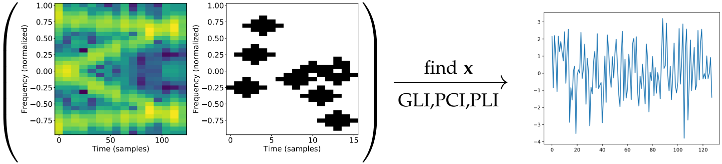


- Signal spectrogram ($T = 16$ and $F = 32$)

Phase inpainting problem



Reconstruction from a random mask with holes of **width one**.

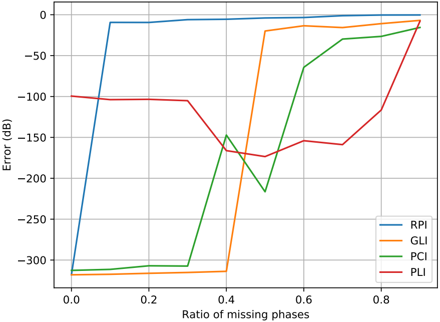


Reconstruction from a random mask with **larger holes**.

Reconstruction error up to a global phase:

$$E_{dB}(\mathbf{x}, \tilde{\mathbf{x}}) = 20 \log_{10} \min_{\theta} \frac{\|\mathbf{x} - e^{\imath\theta}\tilde{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2}$$

Reconstruction error: random holes of size one

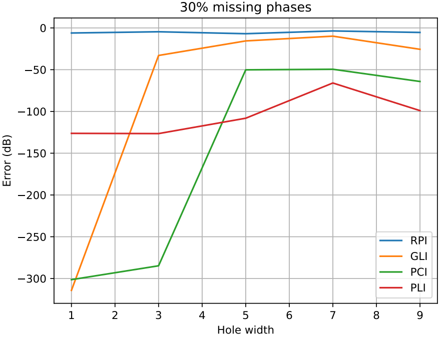


- Perfect reconstruction below 40% by GLI and PCI

- PLI works very well but not perfect

- >= 40% PLI and PCI perform better than GLI, but the performance is even better for PLI

Reconstruction error: large randomly distributed holes

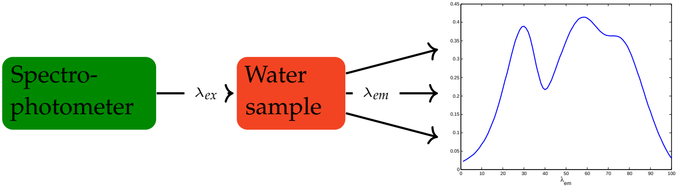


- Bad reconstruction of GLI

- Good performance for SDP methods: PLI and PCI

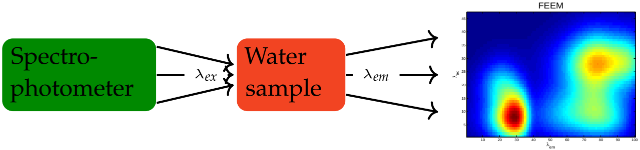
3D fluorescence spectroscopy

- **Problem**: identifying dissolved fluorescent substances in water solutions
- **Method**: fluorescence spectroscopy technique
- **Data acquisition:**



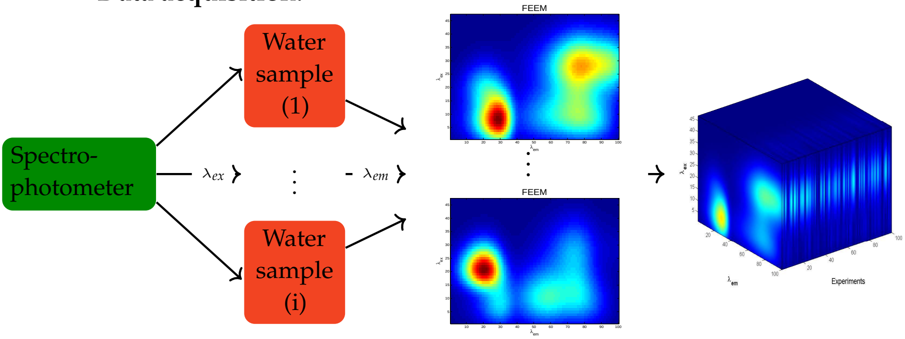
3D fluorescence spectroscopy

- **Problem**: identifying dissolved fluorescent substances in water solutions
- **Method**: fluorescence spectroscopy technique
- **Data acquisition:**



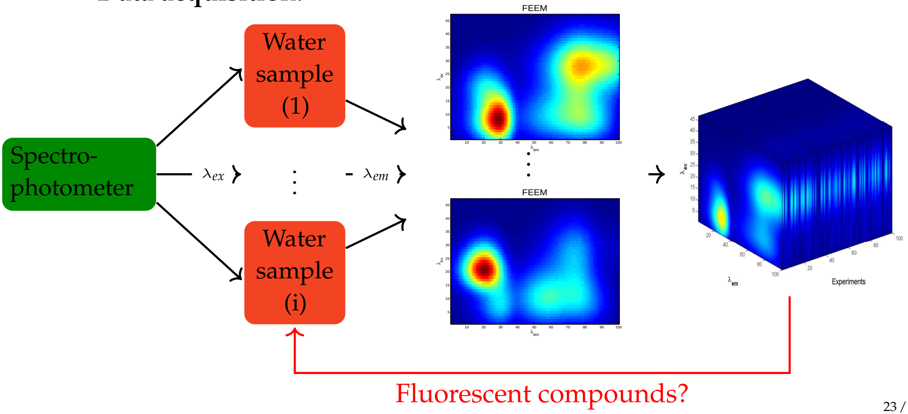
3D fluorescence spectroscopy

- **Problem**: identifying dissolved fluorescent substances in water solutions
- **Method**: fluorescence spectroscopy technique
- **Data acquisition:**

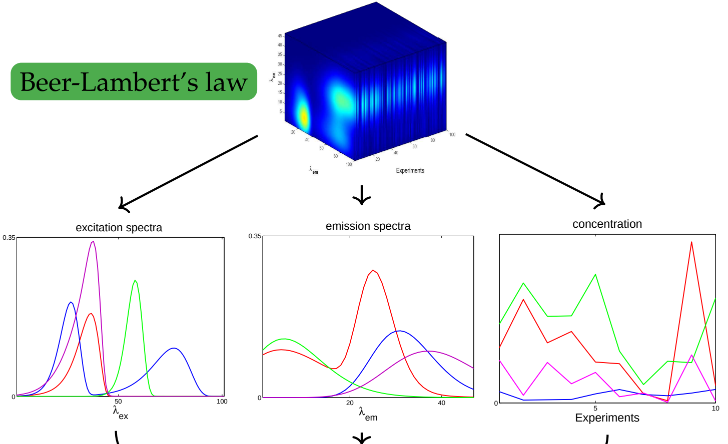


3D fluorescence spectroscopy

- **Problem**: identifying dissolved fluorescent substances in water solutions
- **Method**: fluorescence spectroscopy technique
- **Data acquisition:**



3D fluorescence spectroscopy and tensors



$$\overline{\mathcal{T}} = \sum_{r=1}^{\overline{R}} \bar{\mathbf{a}}_r^{(1)} \circ \bar{\mathbf{a}}_r^{(2)} \circ \bar{\mathbf{a}}_r^{(3)}$$

$\circ$: outer product

(Canonical) Polyadic Decomposition (CPD)

Tensor form: [Harshman1927]

$$\overline{\mathcal{T}} = \sum_{r=1}^{\overbrace{\overline{R}}^{\text{Tensor rank}}} \underbrace{\bar{\mathbf{a}}_r^{(1)} \circ \bar{\mathbf{a}}_r^{(2)} \circ \ldots \circ \bar{\mathbf{a}}_r^{(N)}}_{\text{Rank-1 tensor}} = [\![\bar{\mathbf{A}}^{(1)}, \bar{\mathbf{A}}^{(2)}, \ldots, \bar{\mathbf{A}}^{(N)}]\!]$$

Loading vectors: $\bar{\mathbf{a}}_r^{(n)}$

Loading matrices: $\bar{\mathbf{A}}^{(n)}$

$\forall n \in \{1, 2, \ldots, N\}$, $\bar{\mathbf{a}}_r^{(n)} \in \mathbb{R}^{I_n}$ and $\bar{\mathbf{A}}^{(n)} \in \mathbb{R}^{I_n \times \overline{R}}$

Canonical Polyadic Decomposition (CPD) (2)

Scalar form:

$$\bar{t}_{i_1 \dots i_N} = \sum_{r=1}^{\overline{R}} \bar{a}^{(1)}_{i_1 r} \bar{a}^{(2)}_{i_2 r} \dots \bar{a}^{(N)}_{i_N r}$$

Matrix form: [Cichocki2009]

$$\overline{\mathbf{T}}^{(n)}_{I_n, I_{-n}} = \bar{\mathbf{A}}^{(n)} (\overline{\mathbf{Z}}^{(-n)})^\top, \quad n \in \{1, \dots, N\},$$

$\overline{\mathbf{T}}^{(n)}_{I_n, I_{-n}} \in \mathbb{R}_+^{I_n \times I_{-n}}$: the matrix obtained by unfolding $\overline{\mathcal{T}}$ in the n-th mode, $I_{-n} = I_1 \dots I_N / I_n$; for all $n \in \{1, \dots, N\}$,

$$\boxed{\overline{\mathbf{Z}}^{(-n)} = \bar{\mathbf{A}}^{(N)} \odot \dots \bar{\mathbf{A}}^{(n+1)} \odot \bar{\mathbf{A}}^{(n-1)} \odot \bar{\mathbf{A}}^{(1)} \in \mathbb{R}_+^{I_{-n} \times \overline{R}}}$$

$\odot$: Khatri-Rao product.

Objective: tensor decomposition

- **Input**: Observed tensor $\mathcal{T}$
- **Output**: Estimated loading factors $\widehat{\mathbf{a}}_r^{(n)}$ for all $n \in \{1, \ldots, N\}$

Constraint:

- Loading factors $\bar{\mathbf{a}}_r^{(n)}$ entrywise nonnegative

Difficulties:

- Large dimension tensors
- Rank $\overline{R}$ unknown $\rightarrow$ needs to be estimated (overestimation problems)

Proximal algorithm for CP decomposition

$$\overline{\mathcal{T}} = \sum_{r=1}^{\overline{R}} \bar{\mathbf{a}}_r^{(1)} \circ \ldots \circ \bar{\mathbf{a}}_r^{(N)} = [\![\bar{\mathbf{A}}^{(1)}, \ldots, \bar{\mathbf{A}}^{(N)}]\!].$$

Tensor structure: naturally leads to consider N blocks corresponding to the loading matrices $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}$

Proposed optimization problem

$$\underset{\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R},\, n \in \{1,\ldots,N\}}{\text{minimize}} \quad \mathcal{F}(\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}) + \mathcal{R}_1(\mathbf{A}^{(1)}) + \ldots + \mathcal{R}_N(\mathbf{A}^{(N)})$$

Proximal algorithm for tensor decomposition

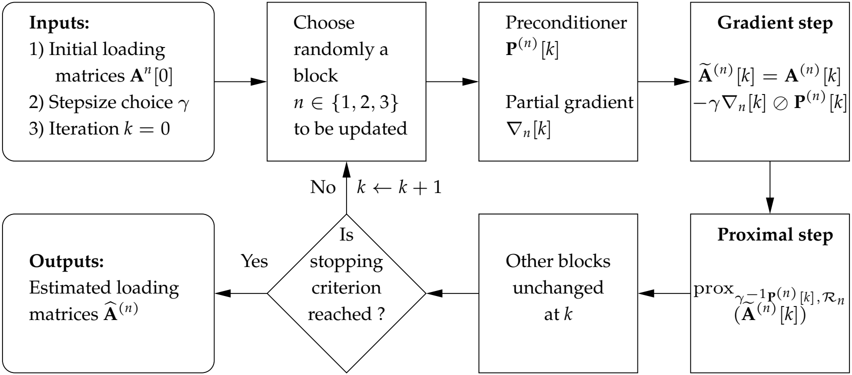


Figure: BC-VMFB algorithm for CPD.

Fidelity term

- $\mathcal{F}(\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)})$: quadratic data fidelity term

$$\begin{aligned}\mathcal{F}(\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)}) &= \frac{1}{2}\|\mathcal{T} - [\![\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)}]\!]\|_F^2 \\ &= \frac{1}{2}\|\mathbf{T}^{(n)}_{I_n, I_{-n}} - \mathbf{A}^{(n)}\mathbf{Z}^{(-n)^\top}\|_F^2\end{aligned}$$

- Gradient matrices of $\mathcal{F}$ with respect to $\mathbf{A}^{(n)}, \forall n = 1, \dots, N$

$$\boxed{\nabla_n \mathcal{F}(\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)}) = -(\mathbf{T}^{(n)}_{I_n, I_{-n}} - \mathbf{A}^{(n)}\mathbf{Z}^{(-n)^\top})\mathbf{Z}^{(-n)}}$$

Regularization terms

- $\mathcal{R}_n(\mathbf{A}^{(n)})$: block dependent penalty terms enforcing sparsity and nonnegativity

$$\mathcal{R}_n(\mathbf{A}^{(n)}) = \sum_{i_n=1}^{I_n} \sum_{r=1}^{R} \rho_n(a_{i_n r}^{(n)}) \qquad \forall n \in \{1, \dots, N\}$$

where loading matrices $\mathbf{A}^{(n)} = (a_{i_n r}^{(n)})_{(i_n, r) \in \{1,\dots,I_n\} \times \{1,\dots,R\}}$

$$\boxed{\rho_n(\omega) = \begin{cases} \alpha^{(n)} |\omega|^{\pi^{(n)}} & \text{if } \eta_{\min}^{(n)} \leq \omega \leq \eta_{\max}^{(n)} \\ +\infty & \text{otherwise} \end{cases}}$$

$\alpha^{(n)} \in]0, +\infty[$, $\pi^{(n)} \in \mathbb{N}^*$, $\eta_{\min}^{(n)} \in [-\infty, +\infty[$ and $\eta_{\max}^{(n)} \in [\eta_{\min}^{(n)}, +\infty]$
$\Rightarrow$ block dependent but constant within a block regularization parameters

Preconditioning

- Preconditioner matrix $\mathbf{P}$ for the n-th block, $\forall n \in \{1, \ldots, N\}$

$$\boxed{\mathbf{P}^{(n)}(\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}) = \mathbf{A}^{(n)}(\mathbf{Z}^{(-n)^{\top}}\mathbf{Z}^{(-n)}) \oslash \mathbf{A}^{(n)}}$$

$\forall n \in \{1, \ldots, N\}$, $\mathbf{A}^{(n)}$ must be non zero
$\oslash$: Hadamard entry-wise division
(Preconditioning: extension of the one used in NMF [Lee and Seung, 2001])

Proximity operator

Proximity operator of $\mathcal{R}_n$ associated with $\mathbf{P}^{(n)}$

$$\operatorname{prox}_{\gamma[k]^{-1}\mathbf{P}^{(n)}[k],\mathcal{R}_n}(y) = \left(\operatorname{prox}_{\gamma[k]^{-1}p_i^{(n)}[k],\rho_n}(y^{(i)})\right)_{i\in\{1,\ldots,RI_n\}}$$

$(\forall y = (y^{(i)})_{i\in\{1,\ldots,RI_n\}} \in \mathbb{R}^{RI_n})$, where $(\forall i \in \{1, \ldots, RI_n\})$, $(\forall v \in \mathbb{R})$

$$\boxed{\operatorname{prox}_{\gamma[k]^{-1}p_i^{(n)},\rho_n}(v) = \min\left\{\eta_{\max}^{(n)}, \max\left\{\eta_{\min}^{(n)}, \operatorname{prox}_{\gamma[k]\alpha^{(n)}(p_i^{(n)}[k])^{-1}|\,.\,|^{\pi^{(n)}}}(v)\right\}\right\}}$$

(separable structure, diagonal preconditioning matrices, componentwise calculation)

Proximity operator

Proximity operator of $\mathcal{R}_n$ associated with $\mathbf{P}^{(n)}$

$$\mathrm{prox}_{\gamma[k]^{-1}\mathbf{P}^{(n)}[k],\mathcal{R}_n}(y) = \left(\mathrm{prox}_{\gamma[k]^{-1}p_i^{(n)}[k],\rho_n}(y^{(i)})\right)_{i\in\{1,\dots,RI_n\}}$$

$(\forall y = (y^{(i)})_{i\in\{1,\dots,RI_n\}} \in \mathbb{R}^{RI_n})$, where $(\forall i \in \{1,\dots,RI_n\})$, $(\forall v \in \mathbb{R})$

$$\boxed{\mathrm{prox}_{\gamma[k]^{-1}p_i^{(n)},\rho_n}(v) = \min\left\{\eta^{(n)}_{\max}, \max\left\{\eta^{(n)}_{\min}, \mathrm{prox}_{\gamma[k]\alpha^{(n)}(p_i^{(n)}[k])^{-1}|\,.\,|^{\pi^{(n)}}}(v)\right\}\right\}}$$

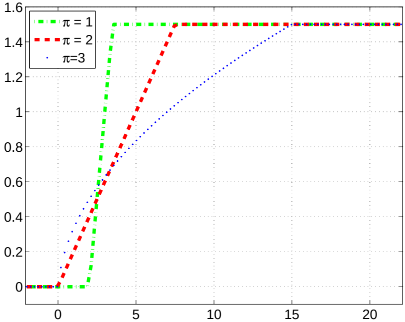
Example:

$\mathrm{prox}_{\rho_n}(v)$ where $v \in [-2, 22]$,
$[\eta^{(n)}_{\min}, \eta^{(n)}_{\max}] = [0, 1.5]$, $\alpha^{(n)} = 2$ and

1) $\pi^{(n)} = 1$

2) $\pi^{(n)} = 2$

3) $\pi^{(n)} = 3$



Experiments on simulated data

- Simulated tensor $\overline{\mathcal{T}}$: (uni or bimodal type) emission and excitation spectra, $\overline{R} = 5$
- Simulated observed tensor: $\mathcal{T} = \overline{\mathcal{T}} + \mathcal{B}$, $\mathcal{B}$: white Gaussian noise
- 2 considered cases:
  1. **3D tensor**: $\overline{\mathcal{T}} \in \mathbb{R}_+^{100\times100\times100}$
     + Noiseless case: no noise added, $\widehat{R} = 6$ (overestimation)
  2. **4D tensor**: $\overline{\mathcal{T}} \in \mathbb{R}_+^{100\times100\times100\times100}$
     + Noisy case: SNR $= 18.46$ dB, $\widehat{R} = 7$ (overestimation)
- Error measure:
  1. Signal to Noise Ratio defined as $\mathsf{SNR} = 20\log_{10}\frac{\|\overline{\mathcal{T}}\|_F}{\|\widehat{\mathcal{T}}-\overline{\mathcal{T}}\|_F}$
  2. Estimation error: $E_1 = 10\log_{10}\left(\frac{\sum_{n=1}^{N}\|\widehat{\mathbf{A}}^{(n)}(1{:}\overline{R}) - \bar{\mathbf{A}}^{(n)}\|_1}{\sum_{n=1}^{N}\|\bar{\mathbf{A}}^{(n)}\|_1}\right)$
  3. Over-factoring error $E_2 = 10\log_{10}\left(\|\sum_{r=\overline{R}+1}^{\widehat{R}}\widehat{\mathbf{a}}_r^{(1)}\circ\ldots\circ\widehat{\mathbf{a}}_r^{(N)}\|_1\right)$

Numerical results - 3D tensor

Noisy case

Elapsed time (s)	BC-VMFB	N-way	fast HALS
For 50 iterations	0.2	11	0.5
To reach stopping conditions ($\mathbf{E}_1$, $\mathbf{E}_2$) dB	75 (-11.2, -409)	8 (-12.5, 30.6)	8 (-12.5, 30.6)

Noiseless case

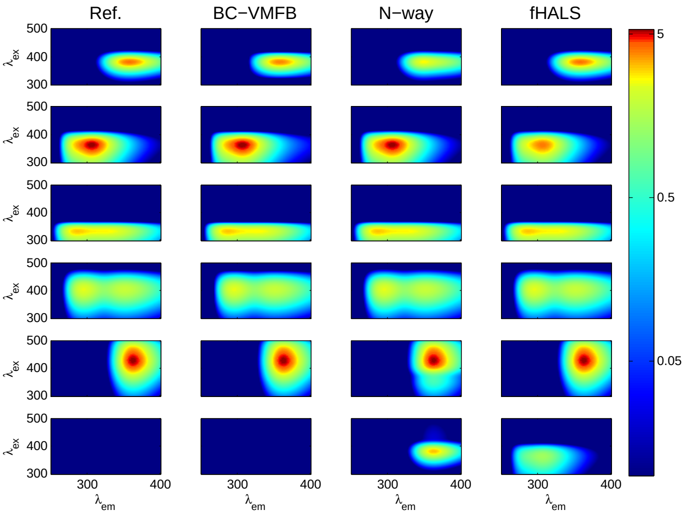
Elapsed time (s)	BC-VMFB	N-way	fast HALS
To reach stopping conditions ($\mathbf{E}_1$, $\mathbf{E}_2$) dB	74 (-15, -409)	80 (-8.7, 31.7)	3.7 (-6.1, 31.7)

Computation time comparison: BC-VMFB (with penalty), N-way [Bro, 1997], fast HALS [Phan et al., 2013] using the same initial value

BC-VMFB:

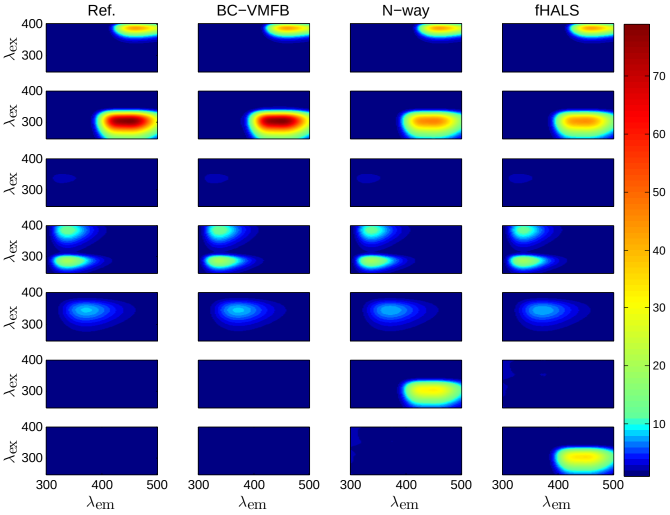
+ Fastest computation time / iteration
+ Smallest estimation error E_1 (noisy case), overestimation error E_2 (both cases)

Visual results: 3D tensor, noiseless case



Penalized BC-VMFB $\alpha = 0.05$

Visual results: 4D tensor, noisy case



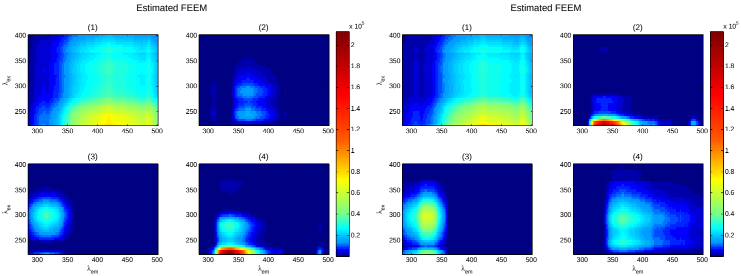
Computer simulation: real experimental data - water monitoring to detect pollutants

- Data were acquired automatically every 3 minutes, during a 10 days monitoring campaign performed on water extracted from an urban river $\Rightarrow$ tensor of size $36 \times 111 \times 2594$.
- The excitation wavelengths range from 225nm to 400nm with a 5nm bandwidth, whereas the emission wavelengths range from 280nm to 500nm with a 2nm bandwidth.
- The FEEM have been pre-processed using the Zepp's method (negative values were set to 0).

Contamination

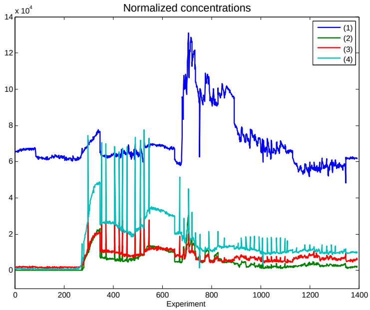
During this experiment, a contamination with diesel oil appeared 7 days after the beginning of the monitoring.

Results: assuming that $\widehat{R} = 4$

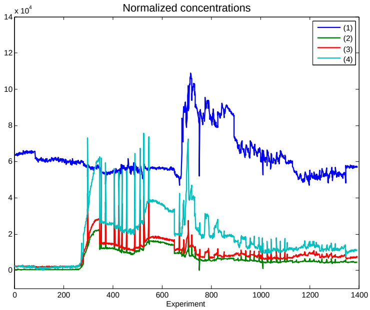


penalized BC-VMFB algorithm | Bro's *N*-way algorithm

Results: concentrations



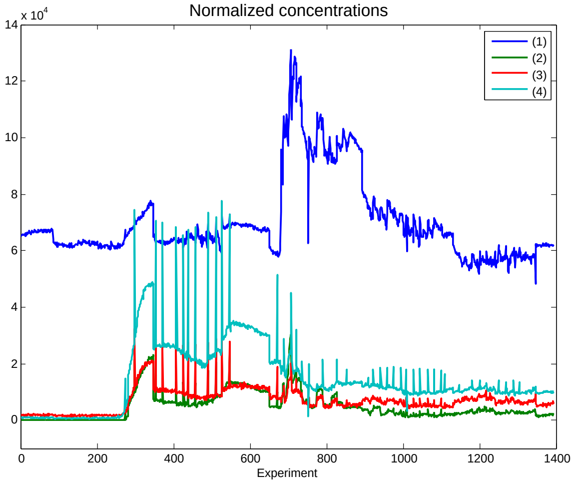
penalized BC-VMFB algorithm



Bro's N-way algorithm

Case $\widehat{R} = 4$

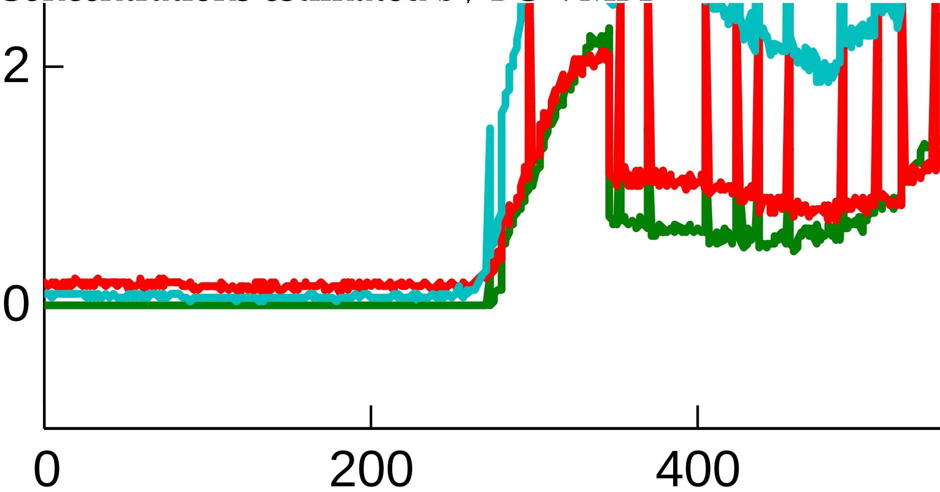
Concentrations estimated by BC-VMFB



↑ Day 7

Case $\widehat{R} = 4$

Concentrations estimated by BC-VMFB



Conclusions and perspectives

- Inverse problems study from model to resolution through parameterization.
- Performance study on simulated data but also on real data.
- Elaboration of efficient methods based on wavelets, optimization, proximal algorithms.

- More efficient methods should be developed for TF inpainting.
- Real data preprocessing should be directly incorporated in the optimization problems.

Conclusions and perspectives

- Inverse problems study from model to resolution through parameterization.
- Performance study on simulated data but also on real data.
- Elaboration of efficient methods based on wavelets, optimization, proximal algorithms.

- More efficient methods should be developed for TF inpainting.
- Real data preprocessing should be directly incorporated in the optimization problems.

Thank you !