Extrapolation on BCD for constrained Matrix/Tensor Factorization

- Andersen Ang (Postdoc in Dept. Combinatorics and Optimization, UWaterloo, Canada)

- Joint work with
  - Jérémy E. Cohen (CNRS, IRISA Rennes, France)
  - Nicolas Gillis, Le Thi Khanh Hien (UMONS, Belgium)
  - Valentin Leplat (UCLouvain, Belgium)

1. (On Complex NMF) **A.**, Leplat, Gillis, "Fast algorithm for complex-NMF with application to source separation", Submitted to conference EUSIPCO21, February 2021
2. (On NTF/NCPD) **A.**, Cohen, Gillis, Hien, "Accelerating Block Coordinate Descent for Nonnegative Tensor Factorization", Journal Numerical Linear Algebra with Applications, 2021
3. (On CPD) **A.**, Cohen, Hien and Gillis, "Extrapolated Alternating Algorithms for Approximate Canonical Polyadic Decomposition", Conference IEEE ICASSP 2020
4. (On NMF) **A.** and Gillis, "Accelerating Nonnegative Matrix Factorization Algorithms using Extrapolation", Journal Neural Computation 31 (2), 2019

One sentence summary / take-home

An heuristic extrapolated BCD algorithm with reduced restart cost, that works fast on X-factorization.

What problem to solve

- (Nonnegative) Canonical Polyadic Decomposition[1] (CPD) with known factorization rank

- Given $r \in \mathbb{N}_+$ and $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, find $\mathbf{A}_i \in \mathbb{R}^{I_i \times r}$ for $i = 1, 2 \ldots, N$ by solving

$$(\mathcal{P}) \ : \ \min_{\mathbf{A}_1, \ldots, \mathbf{A}_N} \left\| \mathcal{T} - \mathcal{I} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \times_3 \cdots \times_N \mathbf{A}_N \right\|_F^2$$

  subject to $\mathbf{A}_i \in \mathcal{C}_i$ (for example, nonnegativity).

- Statement: try to accelerate (exact and inexact) BCD algorithm for solving $(\mathcal{P})$.

[1] The acceleration also works for Tucker model and even in complex numbers.

What problem to solve – simplified case

- Given r and $\mathcal{T} \in \mathbb{R}^{I \times J \times K}$, find $\mathbf{U} \in \mathbb{R}^{I \times r}$, $\mathbf{V} \in \mathbb{R}^{J \times r}$, $\mathbf{W} \in \mathbb{R}^{K \times r}$ by

$$(\mathcal{P}) : \min_{\mathbf{U} \geq \mathbf{0}, \mathbf{V} \geq \mathbf{0}, \mathbf{W} \geq \mathbf{0}} f(\mathbf{U}, \mathbf{V}, \mathbf{W}) := \left\| \mathcal{T} - \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} \right\|_F^2$$

  subject to, says $\mathbf{U}, \mathbf{V}, \mathbf{W}$ nonnegative.
- The BCD that you know very well

Algorithm 1: A typical BCD to solve 3rd-order NCPD

Result: $\mathbf{U}, \mathbf{V}, \mathbf{W}$ that minimize f

Initialize $\mathbf{U}_0, \mathbf{V}_0, \mathbf{W}_0$;

while *not converge* **do**

  $\mathbf{U}_k = \underset{\mathbf{U} \geq \mathbf{0}}{\operatorname{argmin}} \ f(\mathbf{U}, \mathbf{V}_{k-1}, \mathbf{W}_{k-1})$;

  $\mathbf{V}_k = \underset{\mathbf{V} \geq \mathbf{0}}{\operatorname{argmin}} \ f(\mathbf{U}_k, \mathbf{V}, \mathbf{W}_{k-1})$;

  $\mathbf{W}_k = \underset{\mathbf{W} \geq \mathbf{0}}{\operatorname{argmin}} \ f(\mathbf{U}_k, \mathbf{V}_k, \mathbf{W})$;

end

- Can use what ever solver on each sub-problem:
  (acc-)gradient descent, active set, ADMM, (acc-)HALS, MU
- Note: exact BCD vs inexact BCD.

Algorithm 2: Heuristic Extrapolated BCD with Restarts (HER)

Result: $\mathbf{U}, \mathbf{V}, \mathbf{W}$ that minimize f

Initialize $\mathbf{U}_0, \mathbf{V}_0, \mathbf{W}_0$, $\hat{\mathbf{U}}_0, \hat{\mathbf{V}}_0, \hat{\mathbf{W}}_0$, $\beta_0 \in (0, 1]$, $\eta \geq \bar{\gamma} \geq \gamma \geq 1$;

while *not converge* **do**

$\mathbf{U}_k = \operatorname{argmin}_{\mathbf{U}} f(\mathbf{U}, \hat{\mathbf{V}}_{k-1}, \hat{\mathbf{W}}_{k-1})$;

$\hat{\mathbf{U}}_k = [\mathbf{U}_k + \beta_{k-1}(\mathbf{U}_k - \mathbf{U}_{k-1})]_+$;

$\mathbf{V}_k = \operatorname{argmin}_{\mathbf{V}} f(\hat{\mathbf{U}}_k, \mathbf{V}, \hat{\mathbf{W}}_{k-1})$;

$\hat{\mathbf{V}}_k = [\mathbf{V}_k + \beta_{k-1}(\mathbf{V}_k - \mathbf{V}_{k-1})]_+$;

$\mathbf{W}_k = \operatorname{argmin}_{\mathbf{W}} f(\hat{\mathbf{U}}_k, \hat{\mathbf{V}}_k, \mathbf{W})$;

$\hat{\mathbf{W}}_k = [\mathbf{W}_k + \beta_{k-1}(\mathbf{W}_k - \mathbf{W}_{k-1})]_+$;

Compute $\hat{F}_k := f(\hat{\mathbf{U}}_k, \hat{\mathbf{V}}_k, \mathbf{W}_k)$;

if $\hat{F}_k > \hat{F}_{k-1}$ **then**

$\hat{\mathbf{U}}_k = \mathbf{U}_k, \hat{\mathbf{V}}_k = \mathbf{V}_k, \hat{\mathbf{W}}_k = \mathbf{W}_k$;

$\bar{\beta}_k = \beta_{k-1}, \beta_k = \beta_{k-1}/\eta$;

else

$\mathbf{U}_k = \hat{\mathbf{U}}_k, \mathbf{V}_k = \hat{\mathbf{V}}_k, \mathbf{W}_k = \hat{\mathbf{W}}_k$;

$\beta_k = \min\{\bar{\beta}_k, \beta_{k-1}\gamma\}$, $\bar{\beta}_k = \min\{1, \bar{\beta}_{k-1}\bar{\gamma}\}$;

end

end

It works well in NMF

$$\min_{\mathbf{W}\geq\mathbf{0},\mathbf{H}\geq\mathbf{0}} f(\mathbf{W},\mathbf{H}) := \left\|\mathbf{X}-\mathbf{WH}\right\|_F^2$$

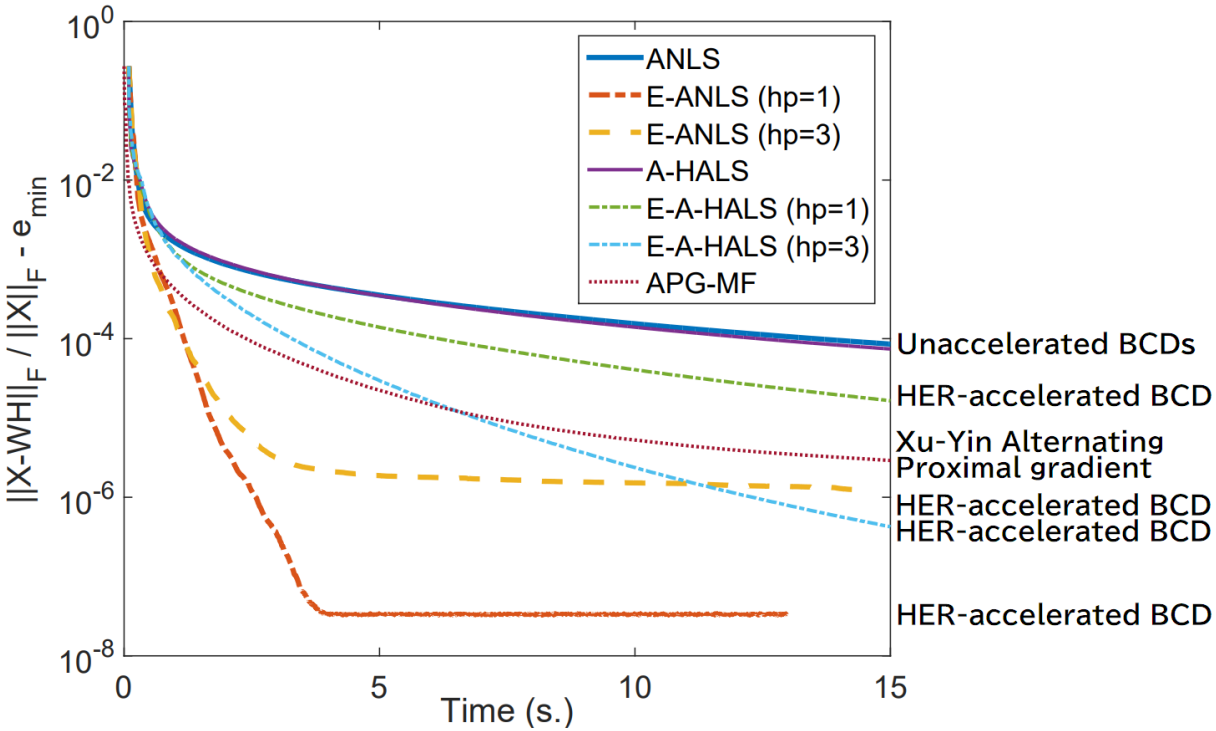


Figure: Average over 100 run of ANLS, A-HALS and their extrapolated variants applied on low-rank (left) synthetic data sets, the method of APG-MF[2] is used for comparisons.

Similar figure for many other cases ...

[2]Xu, Y., Yin, W.: A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM J Imaging Sciences 6(3), 2013

It works well in NTF: CPD form

$$\min_{\mathbf{U}\geq\mathbf{0},\mathbf{V}\geq\mathbf{0},\mathbf{W}\geq\mathbf{0}} f(\mathbf{U},\mathbf{V},\mathbf{W}) := \left\|\mathcal{T} - \mathbf{U}\times_2\mathbf{V}\times_3\mathbf{W}\right\|_F^2$$

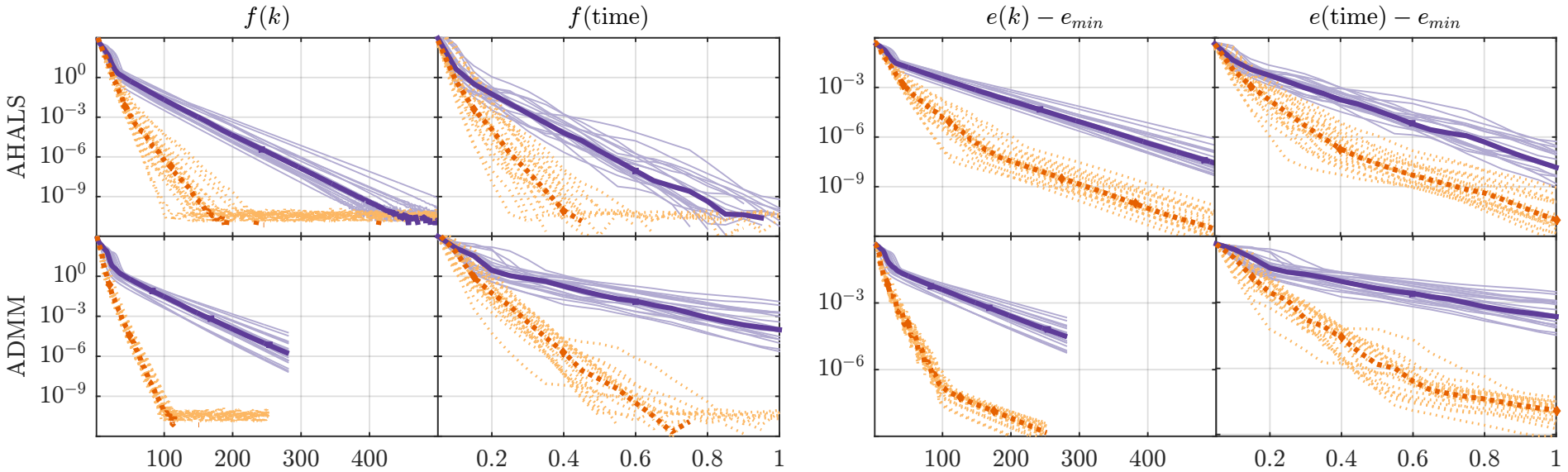


Figure: On 50^3 tensor, rank 10. Blue : ordinary BCD. Orange : HER-BCD.

Similar figure for AS, accPGD, GD, MU ...

It works well in NTF: other cases

Fig.	Test description	$[I_1, I_2, I_3, r, \sigma]$
	Synthetic data	
2	Cube size, low rank, noiseless	$[50, 50, 50, 10, 0]$
2	Unbalanced size, low rank, noiseless	$[150, 10^3, 50, 12, 0]$
3	Unbalanced size, larger rank, noiseless	$[150, 10^3, 50, 25, 0]$
4	Large cube size, low rank, noisy	$[500, 500, 500, 10, 0.01]$
5	Unbalanced size, low rank, noisy, ill-condition	$[150, 10^3, 50, 12, 0.001]$
6	HER-AO-gradients compared with APG and iBPG	$[150, 10^3, 50, 10, 0.01]$
7	Comparing {HER,Bro,GR,LS}-AHALS	$[50, 50, 50, 10, 0]$ $[150, 10^3, 50, 12, 0.01]$ $[150, 10^3, 50, 25, 0.01]$
	Real data	
8	Two HSI images : PaviaU and Indian Pine	$[610, 340, 103, 10]$ $[145, 145, 200, 15]$
9	Big data : black-and-white video sequence	$[153, 238, 1.4 \times 10^4, \{10, 20, 30\}]$

Table 4: List of experiments on NTF.

Same result for AS, accPGD, GD, MU ...

So many test cases because NO theory.

It works well in NTF: with Tucker form data compression

$$\min_{\mathbf{U}\geq\mathbf{0},\mathbf{V}\geq\mathbf{0},\mathbf{W}\geq\mathbf{0}} f(\mathbf{U},\mathbf{V},\mathbf{W}) := \Big\| \underbrace{\mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}}_{\mathcal{T} \text{ is a Tucker-3 tensor}} - \mathcal{I} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} \Big\|_F^2$$

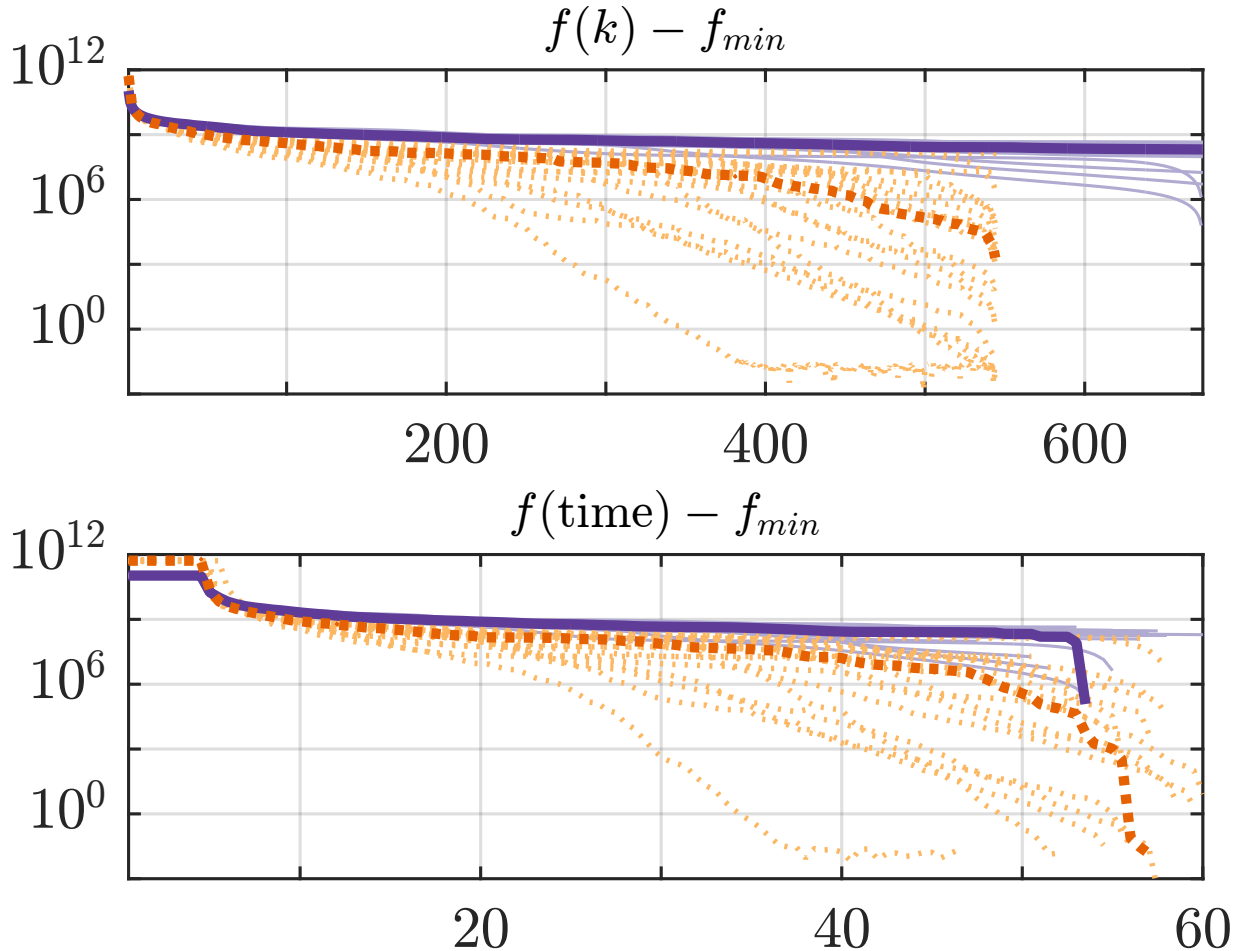


Figure: On a $153 \times 238 \times 14000$ video tensor that has very high mode-wise condition numbers, rank 30.

It works well in CPD

$$\min_{\mathbf{U},\mathbf{V},\mathbf{W}} f(\mathbf{U},\mathbf{V},\mathbf{W}) := \left\|\mathcal{T} - \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}\right\|_F^2$$

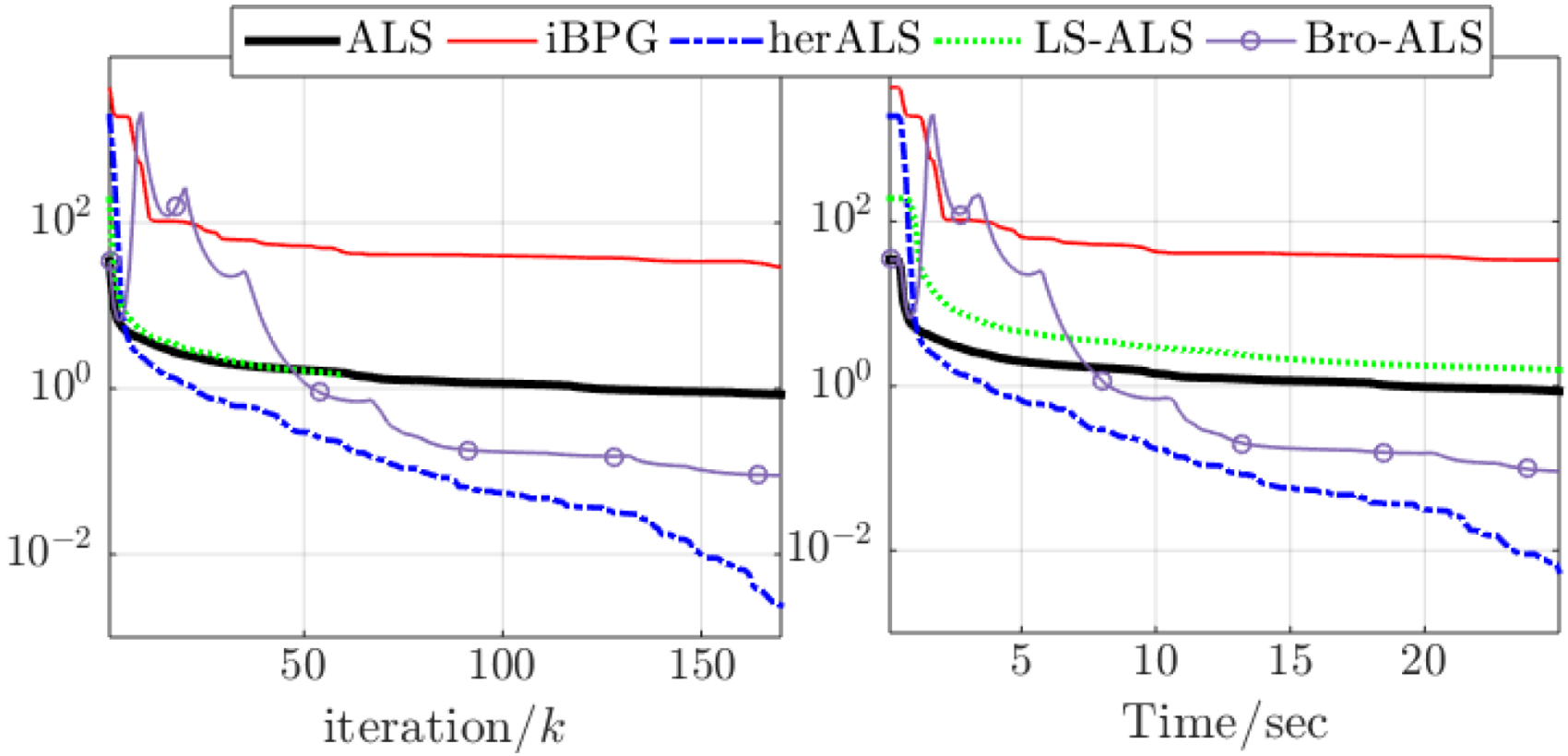


Figure: On Wine data $44 \times 2700 \times 200$, rank 15.

Note: here the problem has no constraint, block-wise subproblem has closed-form solution (ALS, which is also Newton's iteration), so this shows that HER also works with second-order method.

It works well in complex number

$$\min_{\mathbf{W}\geq\mathbf{0},\mathbf{H}\geq\mathbf{0},\mathbf{\Theta}\in[-\pi,\pi]^{m\times n}} f(\mathbf{W},\mathbf{H},\mathbf{\Theta}) := \left\|\mathbf{X}-\mathbf{W}\mathbf{H}\odot\mathbf{e}^{i\mathbf{\Theta}}\right\|_F^2$$

or

$$\min_{\mathbf{W}\geq\mathbf{0},\mathbf{H}\geq\mathbf{0},\mathbf{D}\in\mathbb{C}^{m\times n},|\mathbf{D}_{ij}|=1} f(\mathbf{W},\mathbf{H},\mathbf{D}) := \left\|\mathbf{X}-\mathbf{W}\mathbf{H}\odot\mathbf{D}\right\|_F^2$$

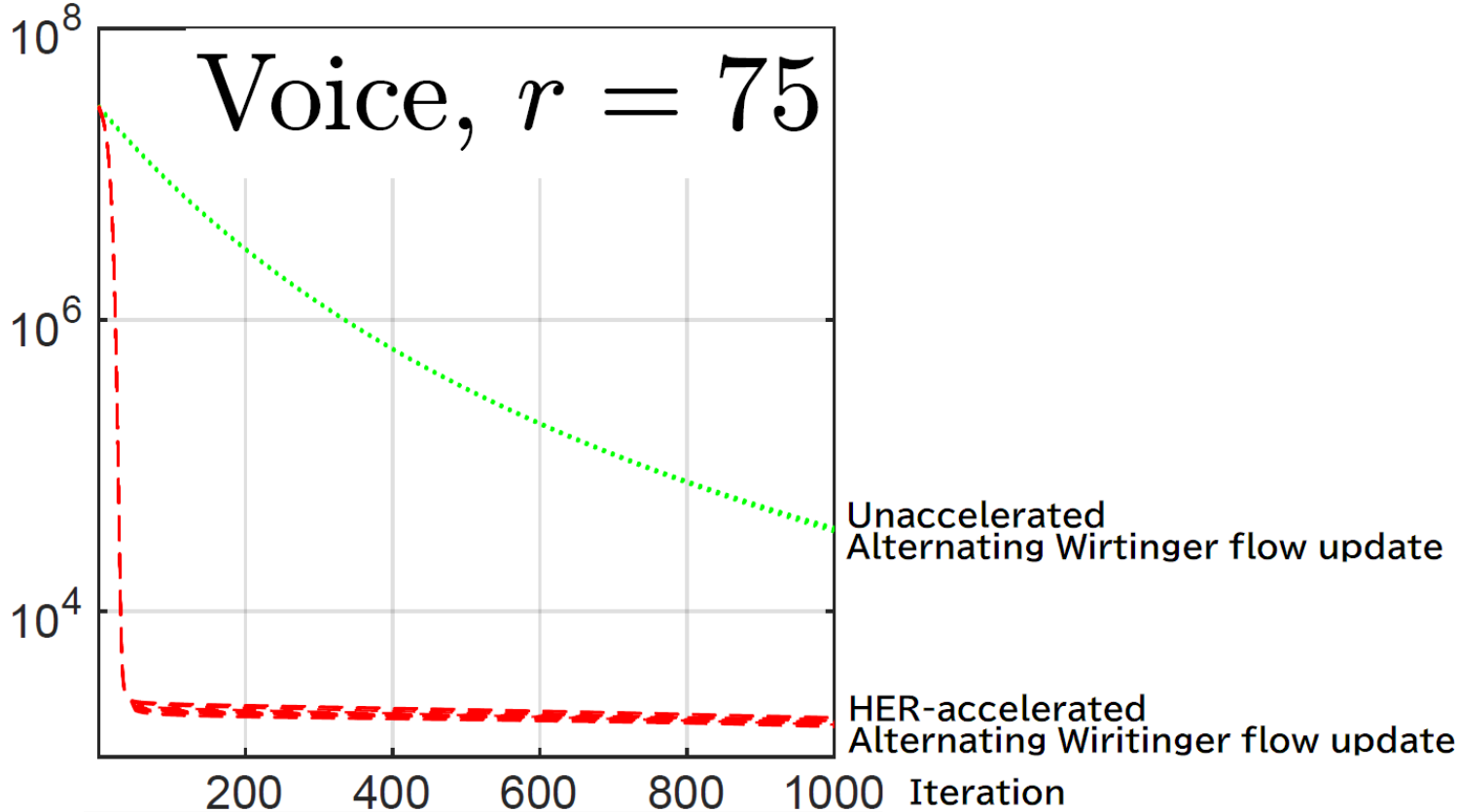


Figure: On fitting a complex matrix (spectrogram of speech data), rank 75.

Algorithm 2: HER

Input: a nonnegative N-way tensor

Output: nonnegative factors $A^{(1)}, A^{(2)}, \ldots, A^{(N)}$.

Initialization: Choose $\beta_0 \in (0,1)$, $\eta \geq \bar{\gamma} \geq \gamma \geq 1$ and 2 sets of initial factor matrices $(A_0^{(1)}, \ldots, A_0^{(N)})$ and $(\hat{A}_0^{(1)}, \ldots, \hat{A}_0^{(N)})$. Set $\bar{\beta}_0 = 1$ and $k = 1$.

repeat

  for $i = 1, \ldots, N$ **do**

    Update step Let $A_k^{(i)}$ be an exact/inexact solution of

$$\min_{A^{(i)} \geq 0} F\left(\hat{A}_k^{(1)}, \ldots, \hat{A}_k^{(i-1)}, A^{(i)}, \hat{A}_{k-1}^{(i+1)}, \ldots, \hat{A}_{k-1}^{(N)}\right). \tag{15}$$

    Extrapolation step

$$\hat{A}_k^{(i)} = \max\left(0, A_k^{(i)} + \beta_{k-1}(A_k^{(i)} - A_{k-1}^{(i)})\right). \tag{16}$$

  end for

  Compute $\hat{F}_k := F\left(\hat{A}_k^{(1)}, \hat{A}_k^{(2)}, \ldots, \hat{A}_k^{(N-1)}, A_k^{(N)}\right)$.

  if $\hat{F}_k > \hat{F}_{k-1}$ **then**

    Set $\hat{A}_k^{(i)} = A_k^{(i)}$, $i = 1, ..., N$ % *abandon the sequence* $\hat{A}_k^{(i)}$

    Set $\bar{\beta}_k = \beta_{k-1}$, $\beta_k = \beta_{k-1}/\eta$. % *Update* $\bar{\beta}$, *decrease* β

  else

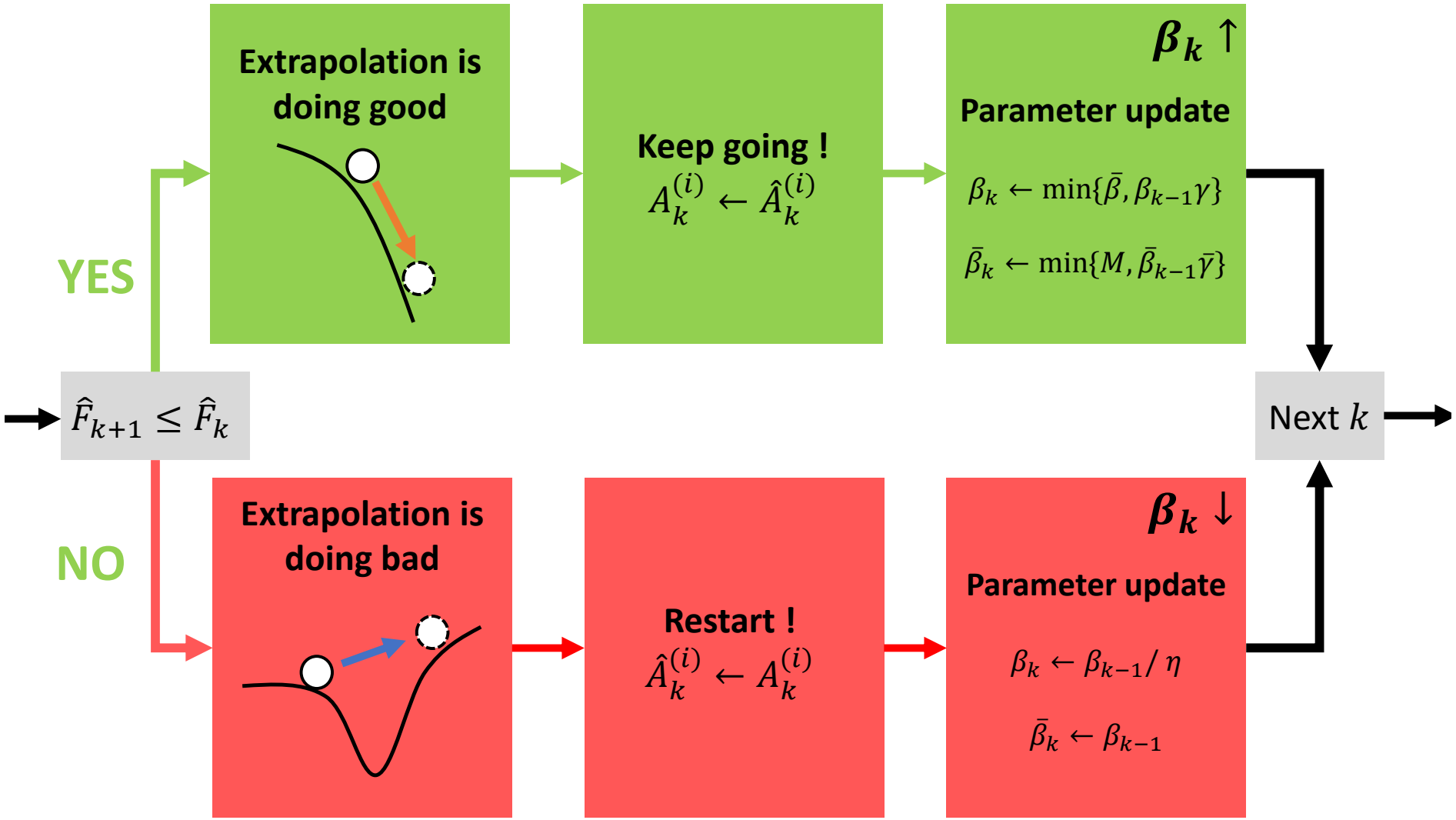
    Set $A_k^{(i)} = \hat{A}_k^{(i)}$, $i = 1, ..., N$. % *keep the sequence* $\hat{A}_k^{(i)}$

    Set $\bar{\beta}_k = \min\{1, \bar{\beta}_{k-1}\bar{\gamma}\}$, $\beta_k = \min\{\bar{\beta}_{k-1}, \beta_{k-1}\gamma\}$. % *Increase* $\bar{\beta}$ *and* β

  end if

  Set $k = k + 1$.

until some criteria is satisfied


Extrapolation is doing good
Keep going !
$A_k^{(i)} \leftarrow \hat{A}_k^{(i)}$
$\boldsymbol{\beta_k} \uparrow$
Parameter update
$\beta_k \leftarrow \min\{\bar{\beta}, \beta_{k-1}\gamma\}$
$\bar{\beta}_k \leftarrow \min\{M, \bar{\beta}_{k-1}\bar{\gamma}\}$
YES
$\hat{F}_{k+1} \leq \hat{F}_k$
Next k
NO
Extrapolation is doing bad
Restart !
$\hat{A}_k^{(i)} \leftarrow A_k^{(i)}$
$\boldsymbol{\beta_k} \downarrow$
Parameter update
$\beta_k \leftarrow \beta_{k-1} / \eta$
$\bar{\beta}_k \leftarrow \beta_{k-1}$

What are the important points

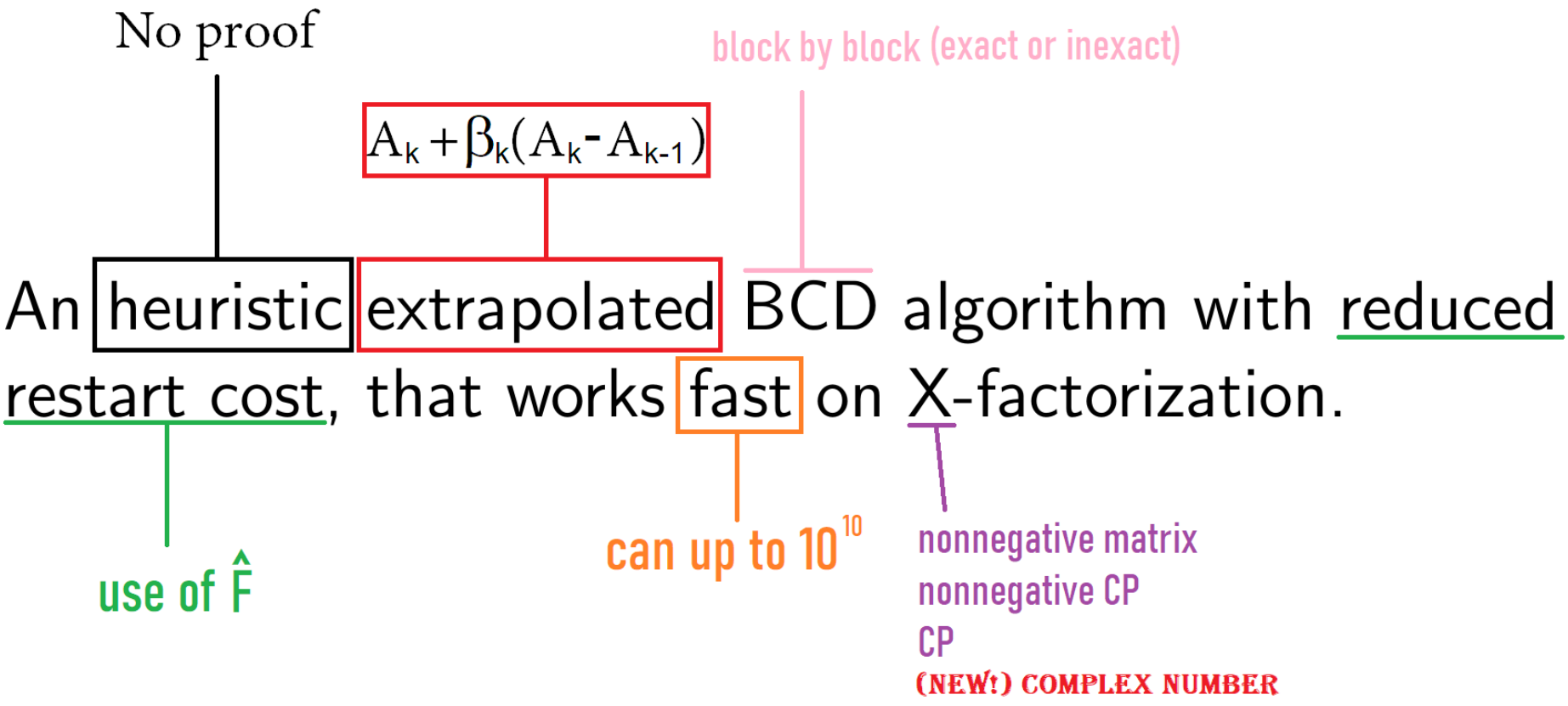
- Why no theory: it is accelerating nonconvex BCD, not Gradient Descent, not many useful tools to use, so cannot prove convergence

- Why faster: acceleration due to extrapolation with **restarts** (safe-guard mechanism) that is **cheap** to compute.

- Why restart cost is low: the use of $\hat{F}$ — reuse already computed component when updating the blocks
  The reduction: from $\mathcal{O}\big(\prod_{i=1}^{N} I_i\big)$ to $\mathcal{O}\big(r^{N-1} I_N\big)$

- Use of $\hat{F}$ as a surrogate of F makes sense: $\lim_{k\to\infty} |\hat{F}_k - F_k| = 0.$

- What about other findings
  - BCD can be replaced by inexact BCD, even 1 step gradient descent
  - Constrained problem can be replaced by unconstrained problem
  - Tensor can be replaced by matrix
  - Even works for complex number



The End.

Papers, code, slide available at angms.science