Hierarchical subtensor partitioning for tensor compression

Virginie Ehrlacher Laura Grigori Damiano Lombardi*

ANR ADAPT

ERC EMC2

Paris 2019

- Motivation
  - Integration of high-dimensional PDEs
  - An example from Kinetic Theory: Vlasov-Poisson system
- Hierarchical subtensor partitioning
  - Partitioning: justifying heuristics
  - First step: a greedy error distribution method
  - Second step: partitioning tree optimisation
- Conclusions and perspectives

High-dimensional problems

- The *curse* of dimensionality
  - A generic classical discretisation is not affordable for high-dimensional problem!

$$\Omega \subseteq \mathbb{R}^d \qquad W^{k,p}(\Omega)$$

$$u \in \mathcal{B}^{k,p}(u_0, 1)$$

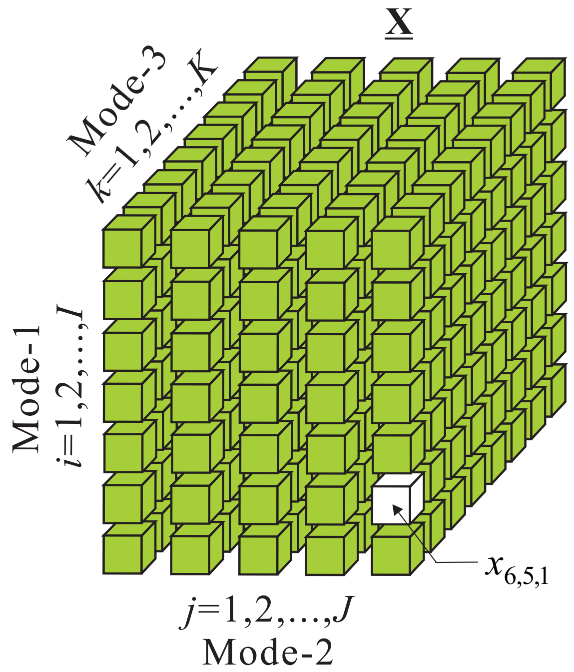
$$\boxed{\varepsilon \propto \mathcal{N}^{-\frac{k}{d}}}$$

- Does the problem nature let us escape the curse?
  - In general we **are not** interested in **generic** elements in the unit ball of a Sobolev space
  - It might happen that such a problem have a remarkable structure to be exploited

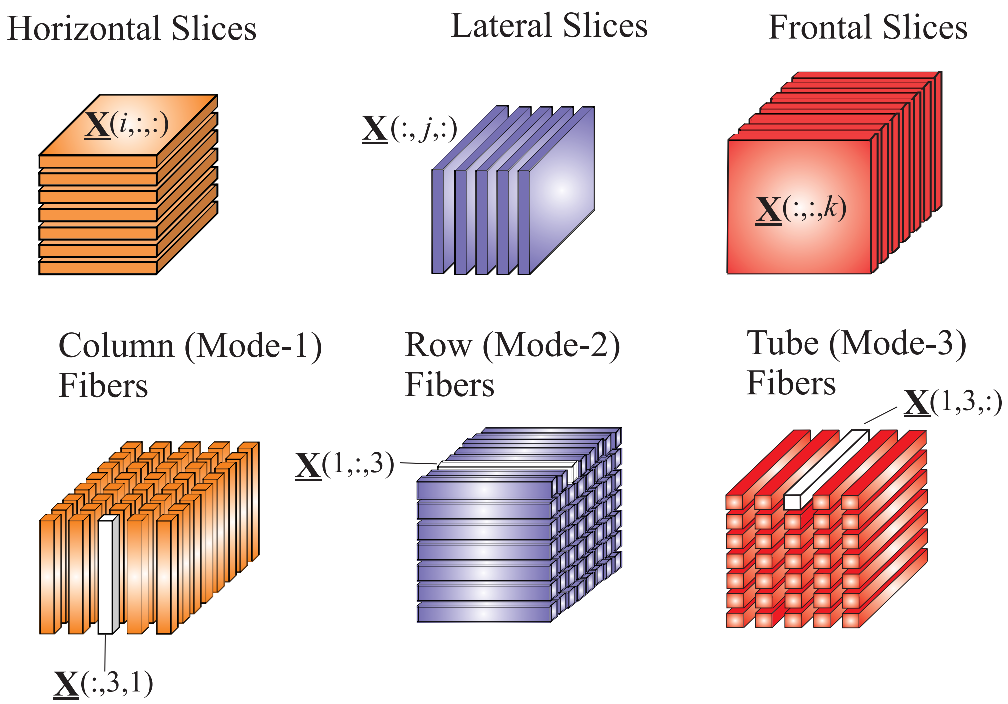
- Tensors.
  - *A priori* way to look for an approximation

Introduction to Tensors:

- Tensors:
  - A priori reduction: representation choice
  - Separation of variables

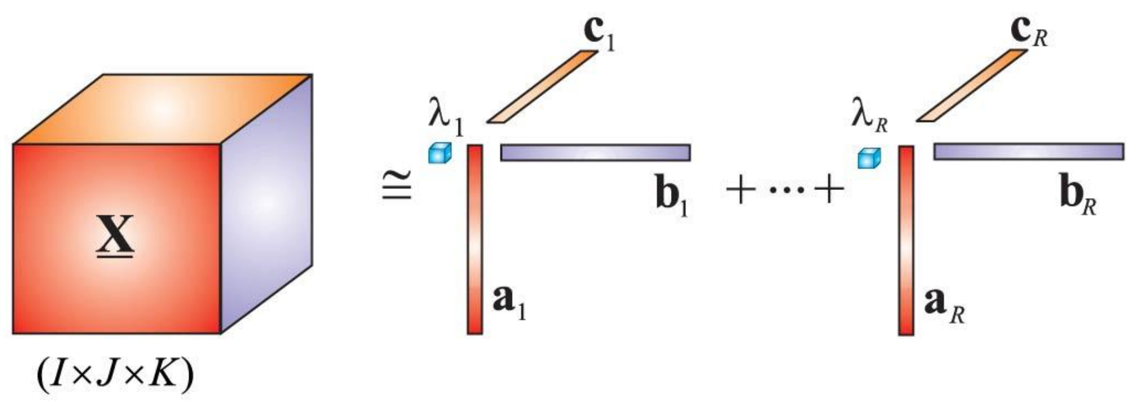


- Does there exist the best rank-k approximation?
  - For matrices (d=2): SVD provides it
  - For tensors d>2: problem is ill-posed (!)



- Several tensor formats were proposed
  - Different ways to write the function approximation

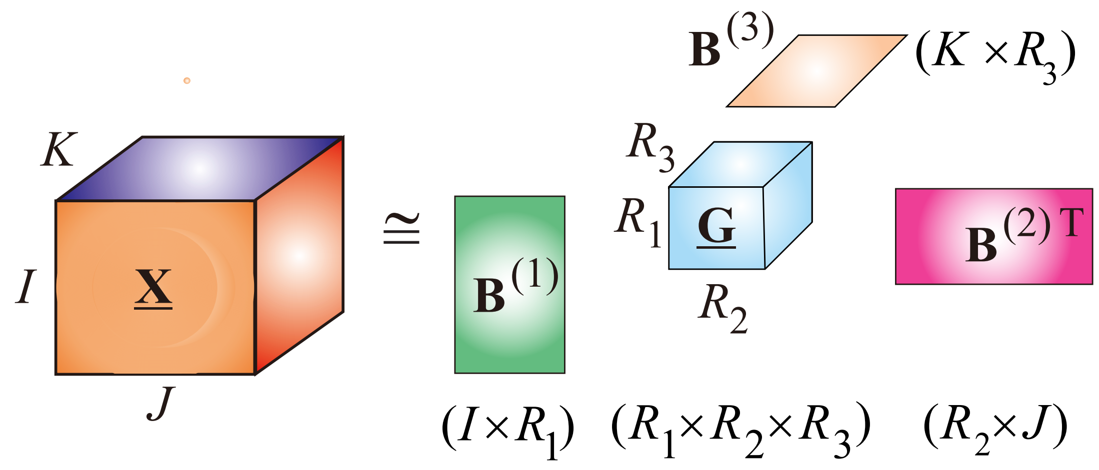
Canonical Polyadic format:



$$u(x_1, x_2, x_3) = \sum_{i=1}^{n} r_i^{(1)}(x_1) r_i^{(2)}(x_2) r_i^{(3)}(x_3)$$

- PGD: greedy algorithm, n is not necessarily fixed a priori
- Euler-Lagrange system solved by means of ALS
- Canonical rank: n
- Storage: $\mathcal{N} = n(I_1 + I_2 + I_3)$

Tucker format:



$$u(x_1, x_2, x_3) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} G_{ijk} r_i^{(1)}(x_1) r_j^{(2)}(x_2) r_k^{(3)}(x_3)$$

- HOSVD method
- Tucker rank: (n_1, n_2, n_3)
- Storage: $\mathcal{N} = n_1 n_2 n_3 + n_1 I_1 + n_2 I_2 + n_3 I_3$

Kinetic Theory

- Domain: $\Omega = \Omega_x \times \Omega_v \times \mathbb{R}^+$
- Unknown: $f(t,x,v) : \Omega \to \mathbb{R}^+$
- Vlasov-Poisson:

$$\partial_t f + v \cdot \nabla_x f + a \cdot \nabla_v f = 0$$

$$a(t,x) = -\nabla_x U(t,x)$$

$$\Delta U = 1 - \varrho(t,x) \qquad \varrho(t,x) = \int_{\Omega_v} f(t,x,v)\, dv$$

$$f(0,x,v) = f_0(x,v)$$

- Theorem:* If $f_0 \in L^1(\mathbb{R}^3 \times \mathbb{R}^3) \times L^\infty(\mathbb{R}^3 \times \mathbb{R}^3)$ satisfies the following condition:

$$\int_{\mathbb{R}^3 \times \mathbb{R}^3} |v|^m f_0(x,v)\, dx\, dv < +\infty \quad \text{for some } m > 3,$$

then, there exists a global strong non-negative solution f to the Vlasov-Poisson system so that

$$f \in \mathcal{C}(\mathbb{R}^+; L^1(\mathbb{R}^3 \times \mathbb{R}^3)) \cap L^\infty(\mathbb{R}^+ \times \mathbb{R}^3 \times \mathbb{R}^3).$$

* P.L.Lions, B.Perthame, *Propagation of moments and regularity for the 3-dimensional Vlasov-Poisson system.*, Inventiones mathema
(1991), Volume: 105, Issue: 2, page 415-430.

Notation:

- Tensor = separate discretisation for x and v

$$g(x,v) = \sum_{k=1}^{n} r_k(x) s_k(v)$$

n is the tensor rank

- For any functions $r : \Omega_x \to \mathbb{R}$ and $s : \Omega_v \to \mathbb{R}$, we use the notation

$$r \otimes s : \begin{cases} \Omega_x \times \Omega_v & \to & \mathbb{R} \\ (x,v) & \mapsto & r(x)s(v). \end{cases}$$

- For any linear operator A acting on real valued functions defined over Ω_x, and for any B acting on real valued functions defined over Ω_v

$$(A \otimes B)(r \otimes s) = (Ar) \otimes (Bs), \quad \text{for all } r : \Omega_x \to \mathbb{R}, \quad s : \Omega_v \to \mathbb{R}.$$

- The main idea:

$$f(t,x,v) \approx \sum_{k=1}^{n_t} r_k(x,t)\, s_k(v,t)$$

$n_t \propto t$, $r_k(x,t)\,s_k(v,t) \propto t$

- At each time step, the solution is decomposed into a sum of pure product tensor functions

$$\partial_t f + \{f,h\} = \partial_t f + \left(\sum_{i=1}^{d} \partial_{x_i} \otimes v_i + \sum_{i=1}^{d} a_i(x,t) \otimes \partial_{v_i} \right) f = 0.$$

tensorised operator:

$$\sum_{k=1}^{n_t} \left(\sum_{i=1}^{d} \partial_{x_i} r_k \otimes v_i s_k + \sum_{i=1}^{d} a_i(x,t) r_k \otimes \partial_{v_i} s_k \right)$$

- Discretisation in time (Verlet):
  - Elementary step structure: $(I + \Delta t P) f^{\frac{i+1}{3}} = (I + \Delta t Q) f^{\frac{i}{3}}$

- Approximation of each step (modified PGD):
  - The algorithm:

1. initial guess: $\delta f^{(0)} = 0$.
2. For $l \in \mathbb{N}^*$, compute $(r_l, s_l) \in H_x \times H_v$ such that:

$$(r_l, s_l) = \arg\min_{r,s} \|g - \Delta t P \delta f^{(n-1)} - I r \otimes s\|_{H_x \times H_v}$$

3. Set: $\delta f^{(l)} = \delta f^{(l-1)} + r_l \otimes s_l$

- Recompression (POD)

- **Proposition** (Ehrlacher, L. 2016):

  Let $\kappa := \max_{1\leq q\leq N} \|I_x^{-1}P_x^{(q)} \otimes I_v^{-1}P_v^{(q)}\|_{H_x\times H_v}$.
  Then, the algorithm converges provided that:

$$3N\Delta t\kappa < 1.$$

Computational Complexity:

- Modified PGD step: solved by Alternated Least Square

  at iteration $l = K$ of the fixed-point PGD: $\mathcal{O}((N_x + N_v)(n+K)(d+1))$

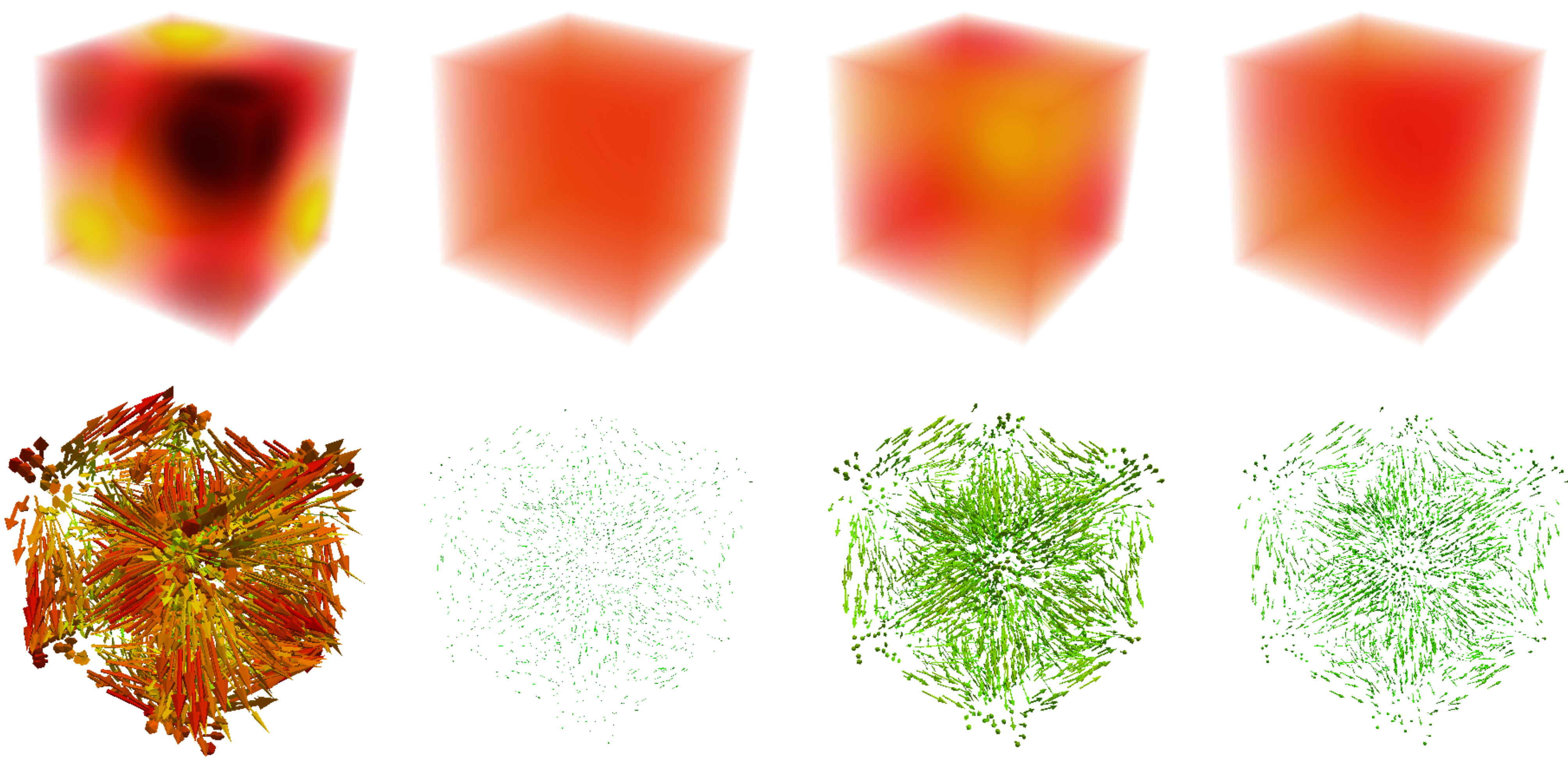
- POD step, solved by QR+SVD

  recompression step: $\mathcal{O}(n^3 + n^2(N_x + N_v))$

Kinetic Theory

Landau Damping 3D:

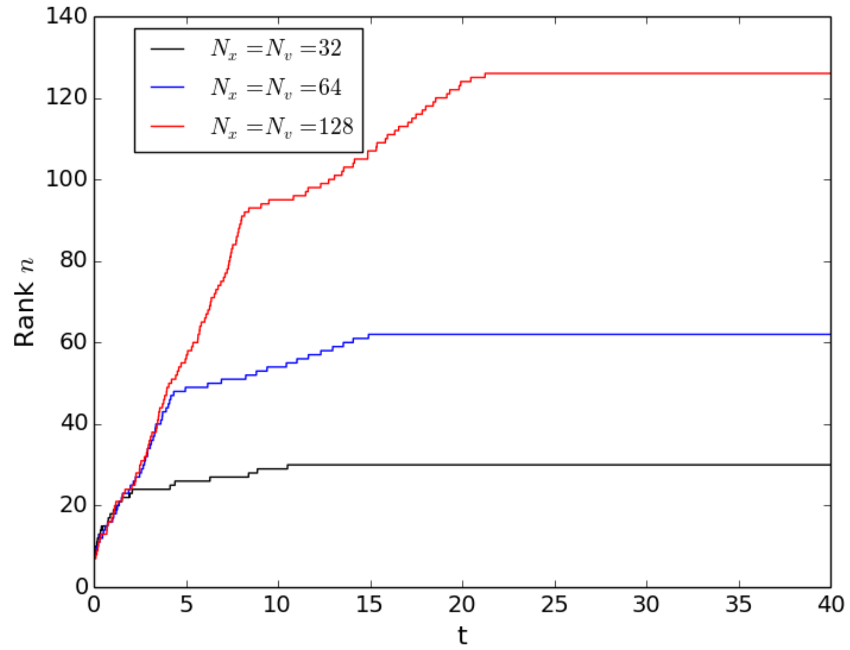
- $N_x = N_v = 64^3$ $\mathcal{N} \approx 6.9\ 10^{10}$ = 549 Gb per time step!
- $c = \mathcal{N} / (\max_t n_t (N_x + Nv)) \approx 1100$



$1 - \varrho$

$-\nabla_x U$

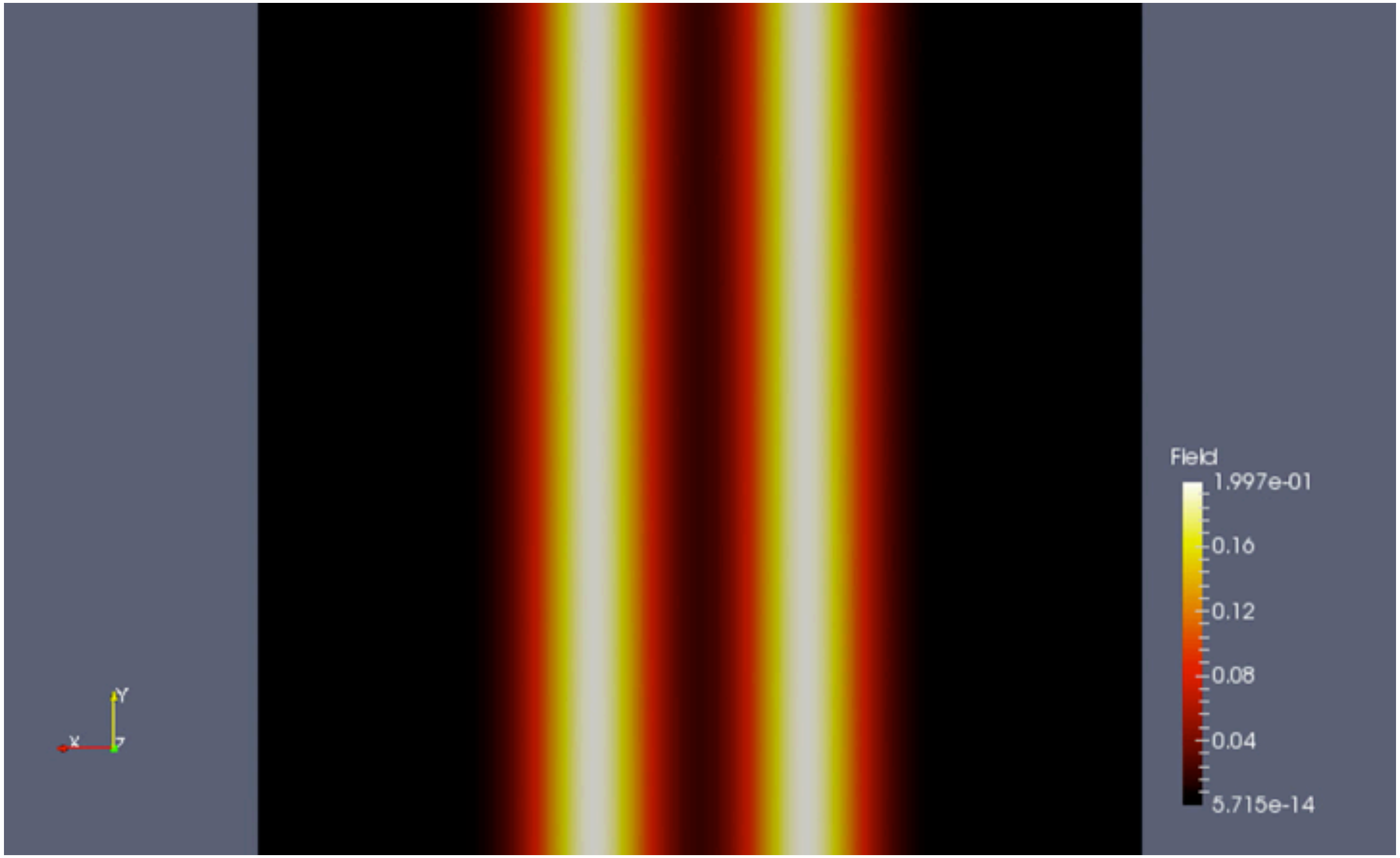
- The rank increase (too much?):
  - The solution cannot be well represented **globally** by tensor approximation



- Explanation:
  - The fact that the residual can be tensorised well does not mean that the solution is small rank.

Bachmayr, M., & Dahmen, W. (2015). Adaptive near-optimal rank tensor approximation for high-dimensional operator equations. *Foundations of Computational Mathematics*, *15*(4), 839-898.

- Two stream instability 1D:



Hierarchical sub-tensors

- Idea:
  - Maybe it is not true that the solution is low rank (can be globally represented with a tensor in an efficient way)
  - Is it true that there exists a congruent domain partitioning such that we can construct a low-rank tensor approximation in each subdomain and guarantee a certain error?

5. *Proposition.* *Let $\Omega_1 = \cdots = \Omega_d = (0,1)$ so that $\boldsymbol{\Omega} := (0,1)^d$. For all $M \in \mathbb{N}^*$, let us consider $P^M := \{(\frac{m-1}{M}, \frac{m}{M})\}_{1\leq m\leq M}$ be a collection of subsets of $(0,1)$ and let $\boldsymbol{P}^M$ be the tensorized domain partition of $\boldsymbol{\Omega}$ associated to the collection of domain partitions $(P_j)_{1\leq j\leq d}$ where $P_j = P^M$ for all $1 \leq j \leq d$. Let $k \in \mathbb{N}^*$, $1 \leq p \leq q \leq \infty$ and $\varepsilon > 0$. We denote $\lambda := \frac{k}{d} - \frac{1}{p} + \frac{1}{q} > 0$. Let $\mathcal{F} \in W^{k,p}(\boldsymbol{\Omega})$ such that $\|\mathcal{F}\|_{W^{k,p}(\boldsymbol{\Omega})} \leq 1$. Then, there exists a constant $C > 0$ which depends only on k, p, d, q such that for all $M \in \mathbb{N}^*$ such that $\ln M \geq -\frac{1}{d\lambda}\ln\left(\frac{\varepsilon}{C}\right)$, there exists a tensor $\mathcal{F}^{CPF}$ of Canonical Partitioning Format with domain partition $\boldsymbol{P}^M$ and rank $R \leq \frac{(k-1+d)!}{(k-1)!d!}$ such that*

$$\|\mathcal{F} - \mathcal{F}^{CPF}\|_{L^q(\boldsymbol{\Omega})} \leq \varepsilon.$$

- A sufficient condition: compact embedding

- **Proof:**
  - Performed in a constructive way.
  - Use of the results presented in:

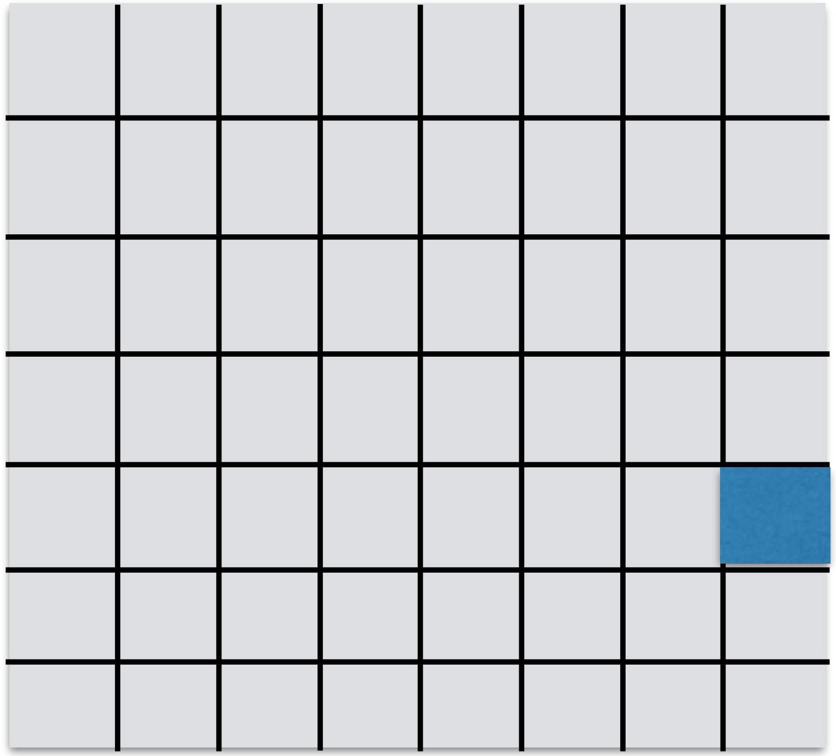
  Edmunds, D. E., & Sun, J. (1990). Approximation and entropy numbers of Sobolev embeddings over domains with finite measure. *The Quarterly Journal of Mathematics*, *41*(4), 385-394.

- **Warnings:**
  - Well adapted for **moderate** dimensional tensors.
  - It is a sufficient condition, the bound on the rank is pessimistic, but:

$$H^1(\Omega) \hookrightarrow L^2(\Omega) \qquad R = 1$$

The method:

- Two step strategy, bottom-up approach.
- Construct an approximation on the finest partition.
- Greedy algorithm to **distribute the error** of the approximation (HOSVD)



- For a generic subdomain i: ε_i

  $\sigma_k^{(i,j)}$ k-th singular value of the j-th unfolding of the i-th subtensor

- Add to the approximation the term that makes the global error decrease the most.

Hierarchical sub-tensors

Algorithm 3.2 PF-Greedy-HOSVD

Input:
$\mathcal{A} \in \mathbb{R}^{\boldsymbol{I}} \leftarrow$ a tensor of order d
$\mathcal{P} \leftarrow$ an admissible partition of $\boldsymbol{I}$
$\varepsilon > 0 \leftarrow$ error tolerance criterion

Output:
Set of Ranks $(\boldsymbol{R}^{\boldsymbol{J}})_{\boldsymbol{J} \in \mathcal{P}} \subset \mathbb{N}^d$
Set of local errors $(\varepsilon^{\boldsymbol{J}})_{\boldsymbol{J} \in \mathcal{P}}$ satisfying $\sum_{\boldsymbol{J} \in \mathcal{P}} |\varepsilon^{\boldsymbol{J}}|^2 < |\varepsilon|^2$.

Begin:
Set $\boldsymbol{R}^{\boldsymbol{J}} := (0, \cdots, 0)$ for all $\boldsymbol{J} \in \mathcal{P}$
while $\displaystyle\sum_{\boldsymbol{J} \in \mathcal{P}} \sum_{1 \le j \le d} \sum_{R_j^{\boldsymbol{J}}+1 \le q_j \le p_j(\mathcal{A}^{\boldsymbol{J}})} \left|\sigma_j^{q_j}(\mathcal{A}^{\boldsymbol{J}})\right|^2 \ge \varepsilon^2$ **do**

    Select $1 \le j_0 \le d$ and $\boldsymbol{J}_0 \in \mathcal{P}$ such that

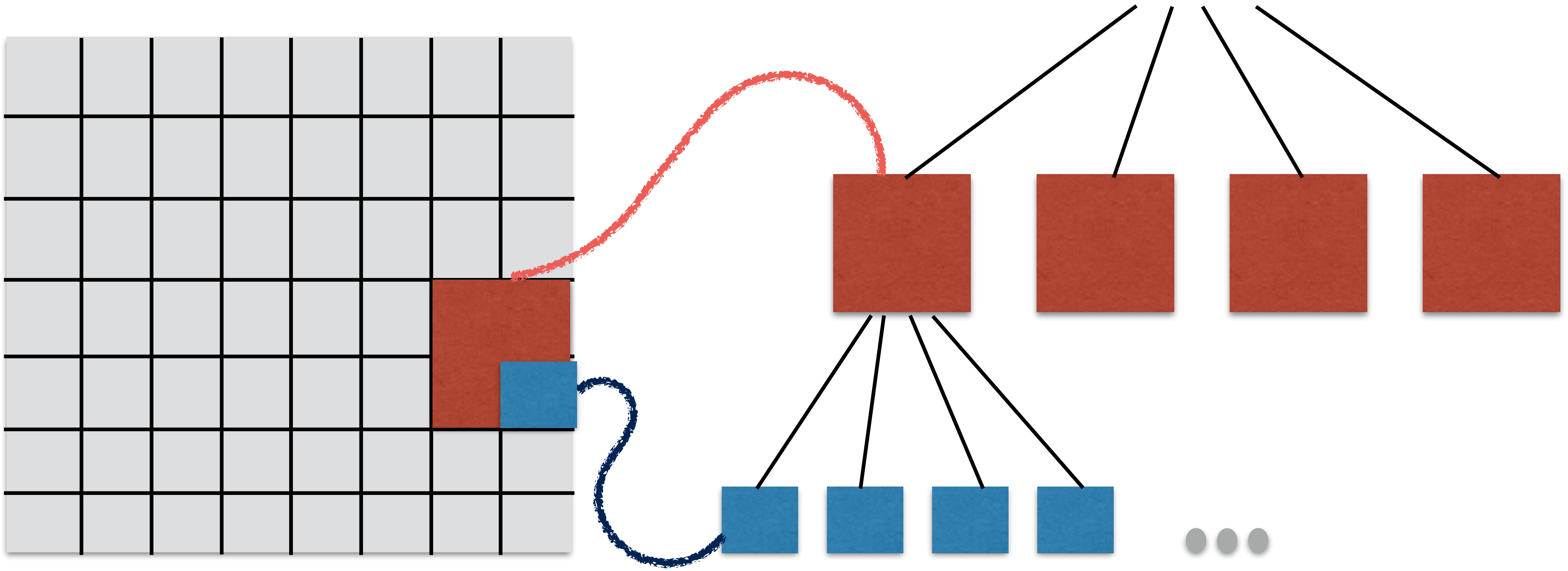
$$(j_0, \boldsymbol{J}_0) = \underset{1 \le j \le d,\ \boldsymbol{J} \in \mathcal{P}}{\operatorname{argmax}} \sigma_j^{R_j^{\boldsymbol{J}}+1}(\mathcal{A}^{\boldsymbol{J}}).$$

    if $\boldsymbol{R}^{\boldsymbol{J}_0} = (0, \cdots, 0)$ **then**
        Set $\boldsymbol{R}^{\boldsymbol{J}_0} := (1, \cdots, 1)$
    else
        Assume that $\boldsymbol{R}^{\boldsymbol{J}_0} = (R_1^{\boldsymbol{J}_0}, \cdots, R_d^{\boldsymbol{J}_0})$.
        $R_{j_0}^{\boldsymbol{J}_0} \leftarrow R_{j_0}^{\boldsymbol{J}_0} + 1$.

Define $\varepsilon^{\boldsymbol{J}} := \sqrt{\sum_{1 \le j \le d} \sum_{R_j^{\boldsymbol{J}}+1 \le q_j \le p_j(\mathcal{A}^{\boldsymbol{J}})} \left|\sigma_j^{q_j}(\mathcal{A}^{\boldsymbol{J}})\right|^2}$ for all $\boldsymbol{J} \in \mathcal{P}$.
    return $\left(\boldsymbol{R}^{\boldsymbol{J}}\right)_{\boldsymbol{J} \in \mathcal{P}}$ and $\left(\varepsilon^{\boldsymbol{J}}\right)_{\boldsymbol{J} \in \mathcal{P}}$

Hierarchical sub-tensors

- Second step:
  - Optimise the partition to **minimise the storage**



Hierarchical sub-tensors

Algorithm 4.2 MERGE

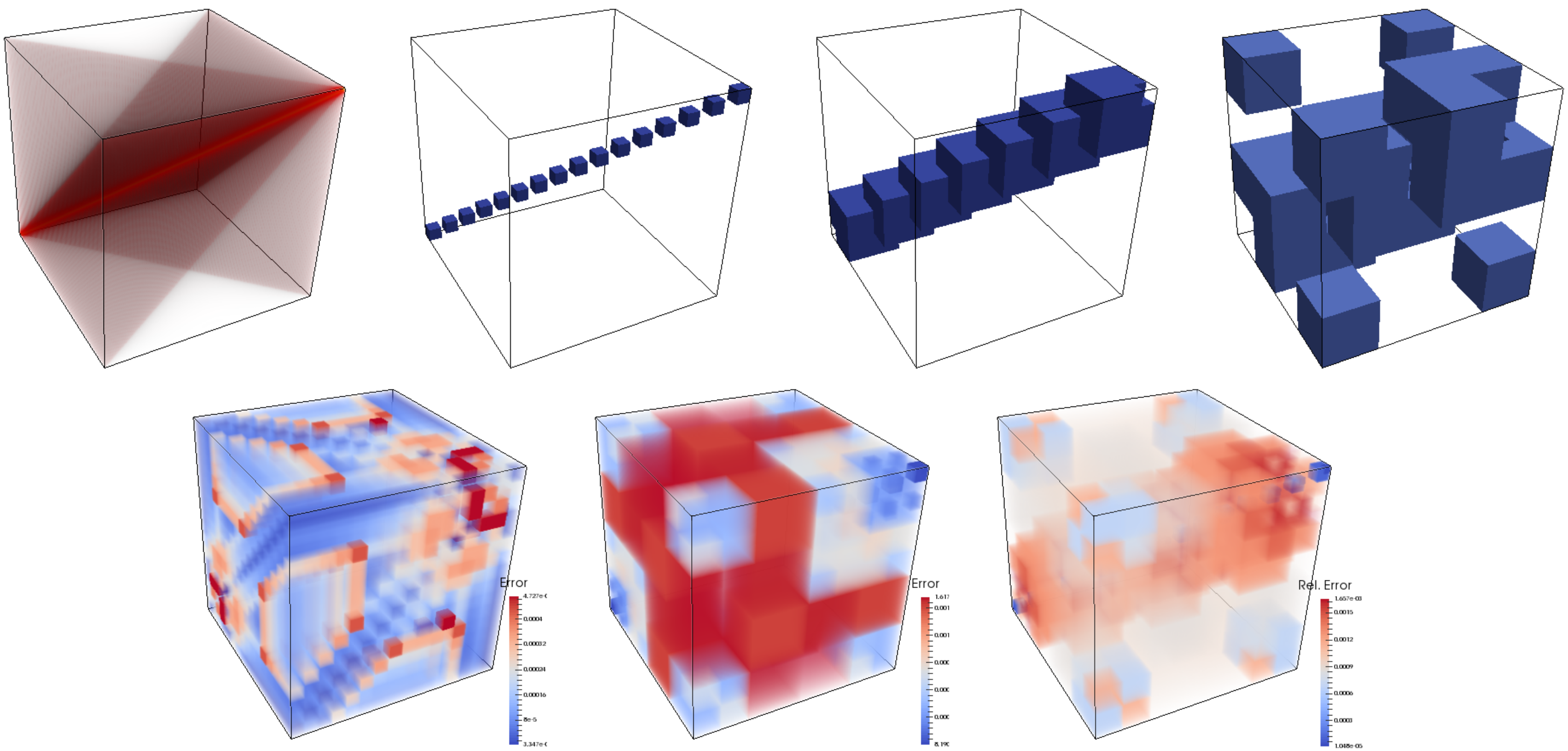
Input:
$\mathcal{A} \in \mathbb{R}^{\boldsymbol{I}} \leftarrow$ a tensor of order d
$T_{\boldsymbol{I}}$ an initial partition tree of $\boldsymbol{I}$
A set of leaf errors $(\varepsilon^{\boldsymbol{J}})_{\boldsymbol{J} \in \mathcal{L}(T_{\boldsymbol{I}})}$
A set of leaf ranks $(\boldsymbol{R}^{\boldsymbol{J}})_{\boldsymbol{J} \in \mathcal{L}(T_{\boldsymbol{I}})} \subset \mathbb{N}^d$
$\boldsymbol{J}_0 \in \mathcal{V}(T_{\boldsymbol{I}}) \setminus \mathcal{L}(T_{\boldsymbol{I}})$.
Output:
$T_{\boldsymbol{I}}^{\text{fin}}$ a final partition tree of $\boldsymbol{I}$
merge a boolean indicating if the tree has been merged or not
A set of leaf errors $(\varepsilon^{\text{fin},\boldsymbol{J}})_{\boldsymbol{J} \in \mathcal{L}(T_{\boldsymbol{I}}^{\text{fin}})}$
A set of leaf ranks $\left(\boldsymbol{R}^{\text{fin},\boldsymbol{J}}\right)_{\boldsymbol{J} \in \mathcal{L}(T_{\boldsymbol{I}}^{\text{fin}})} \subset \mathbb{N}^d$
Begin:
Set $\mathcal{P}_{\boldsymbol{J}_0} := \mathcal{D}_{\boldsymbol{J_0}}(T_{\boldsymbol{I}}) \cap \mathcal{L}(T_{\boldsymbol{I}})$. From Lemma 4.3 (iii), $\mathcal{P}_{\boldsymbol{J}_0}$ is an admissible partition of the set $\boldsymbol{J}_0$.
Set $M_{\text{nomerge}} := \sum_{\boldsymbol{J} \in \mathcal{P}_{\boldsymbol{J}_0}} M_{\text{TF}} \left(\boldsymbol{J}, \boldsymbol{R}^{\boldsymbol{J}}\right)$
Set $\eta := \sqrt{\sum_{\boldsymbol{J} \in \mathcal{P}_{\boldsymbol{J}_0}} |\varepsilon^{\boldsymbol{J}}|^2}$.
Compute $\boldsymbol{R} =$ Greedy-HOSVD$(\mathcal{A}^{\boldsymbol{J}_0}, \eta)$
Compute $M_{\text{merge}} := M_{\text{TF}}(\boldsymbol{J}_0, \boldsymbol{R})$.
if $M_{\text{merge}} < M_{\text{nomerge}}$ **then**
    Set *merge* = true, $T_{\boldsymbol{I}}^{\text{fin}} = T_{\boldsymbol{I}}^{m}(\boldsymbol{J}_0)$.
    for $\boldsymbol{J} \in \mathcal{L}\left(T_{\boldsymbol{I}}^{\text{fin}}\right)$ **do**
        if $\boldsymbol{J} = \boldsymbol{J}_0$ **then**
            Set $\boldsymbol{R}^{\text{fin},\boldsymbol{J}_0} := \boldsymbol{R}$ and $\varepsilon^{\text{fin},\boldsymbol{J}_0} := \eta$.
        else
            From Lemma 4.3 (v), necessarily $\boldsymbol{J} \in \mathcal{L}(T_{\boldsymbol{I}})$.
            Set $\boldsymbol{R}^{\text{fin},\boldsymbol{J}} := \boldsymbol{R}^{\boldsymbol{J}}$ and $\varepsilon^{\text{fin},\boldsymbol{J}} := \varepsilon^{\boldsymbol{J}}$.
else
    Set *merge* = false, $T_{\boldsymbol{I}}^{\text{fin}} = T_{\boldsymbol{I}}$.
    for $\boldsymbol{J} \in \mathcal{L}\left(T_{\boldsymbol{I}}^{\text{fin}}\right) = \mathcal{L}(T_{\boldsymbol{I}})$ **do**
        Set $\boldsymbol{R}^{\text{fin},\boldsymbol{J}} := \boldsymbol{R}^{\boldsymbol{J}}$ and $\varepsilon^{\text{fin},\boldsymbol{J}} := \varepsilon^{\boldsymbol{J}}$.
return $T_{\boldsymbol{I}}^{\text{fin}}$, *merge*, $(\varepsilon^{\text{fin},\boldsymbol{J}})_{\boldsymbol{J} \in \mathcal{L}(T_{\boldsymbol{I}}^{\text{fin}})}$, $\left(\boldsymbol{R}^{\text{fin},\boldsymbol{J}}\right)_{\boldsymbol{J} \in \mathcal{L}(T_{\boldsymbol{I}}^{\text{fin}})}$

- Methods characteristics:
  - Well adapted to **moderate dimension** tensors (partition is subjected to *curse of dimensionality*)
  - It is easily **parallelisable** (!), contrary to classical HOSVD, better suited for large number of degrees of freedom
  - The error is **distributed automatically** and guaranteed throughout the whole approximation
  - Memory gain is significant with respect to classical HOSVD format for a wide class of functions of interest

Hierarchical sub-tensors

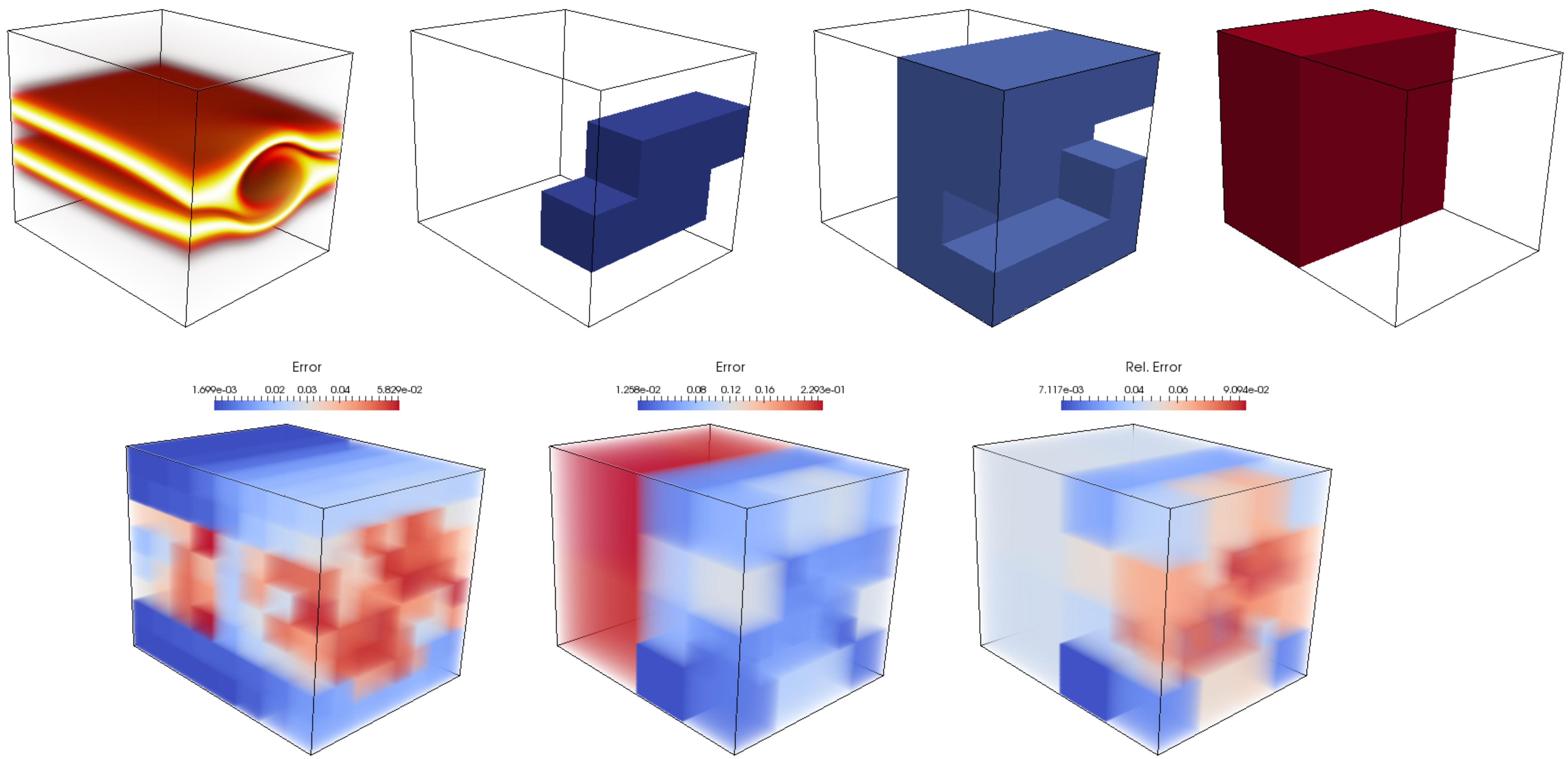
- Coulomb potential:

$$V(x_1,\ldots,x_d) = \sum_{1\le i<j\le d} \frac{1}{|x_i - x_j|}.$$



Hierarchical sub-tensors

- Vlasov Poisson: double stream instability



- Compression factor in phase-space time: 250

Conclusions:

- First steps towards adaptive dynamical discretisations
- Encouraging results on Vlasov-Poisson system
- Moderate order tensors: piece-wise tensor approximation

Perspectives

- Apply piece-wise tensor approximation to Kinetic Theory
- Generalisation to high-order tensors

CEMRACS 2021

Data Assimilation and Model Reduction in high-dimensional problems

CIRM, Luminy, Marseille.
July 17 — August 27, 2021

<u>Organising Committee</u>:

Virginie Ehrlacher
Damiano Lombardi
Olga Mula
Fabio Nobile
Tommaso Taddei

<u>Scientific Committee</u>:

Albert Cohen
Yvon Maday
Gianluigi Rozza
Karen Veroy

Thank you

Second order in time symplectic integrator:

- Let $\Delta t > 0$, $m \in \mathbb{N}^*$, $t_m := m\Delta t$, we denote:

$$f^{(m)}(x,v) \approx f(t_m,x,v)$$

- Three step scheme:

$$\begin{cases} \left(I + \frac{\Delta t}{2} a^{(m)}(x)\cdot\nabla_v\right) f^{(m+1/3)} & = \left(I - \frac{\Delta t}{2} v\cdot\nabla_x\right) f^{(m)}, \\ \left(I + \frac{\Delta t}{2} v\cdot\nabla_x\right) f^{(m+2/3)} & = f^{(m+1/3)}, \\ f^{(m+1)} & = \left(I - \frac{\Delta t}{2} a^{(m+2/3)}(x)\cdot\nabla_v\right) f^{(m+2/3)}, \end{cases}$$

- Elementary step structure:

$$(I + \Delta t P) f^{\frac{i+1}{3}} = (I + \Delta t Q) f^{\frac{i}{3}}$$

- The algorithm:

1. initial guess: $\delta f^{(0)} = 0$.
2. For $l \in \mathbb{N}^*$, compute $(r_l, s_l) \in H_x \times H_v$ such that:

$$(r_l, s_l) = \arg\min_{r,s} \|g - \Delta t P \delta f^{(n-1)} - I r \otimes s\|_{H_x \times H_v}$$

3. Set: $\delta f^{(l)} = \delta f^{(l-1)} + r_l \otimes s_l$

- **Proposition** (Ehrlacher, L. 2016):

Let $\kappa := \max_{1 \leq q \leq N} \|I_x^{-1} P_x^{(q)} \otimes I_v^{-1} P_v^{(q)}\|_{H_x \times H_v}$.
Then, the algorithm converges provided that:

$$3N\Delta t \kappa < 1.$$

- Recompression step:

$$f^{\frac{i}{3}} \approx \sum_{k=1}^{n} \tilde{r}_k \otimes \tilde{s_k}$$

$$\tilde{f}^{\frac{i+1}{3}} \approx \sum_{k=1}^{n} \tilde{r}_k \otimes \tilde{s_k} + \sum_{k=1}^{n_1} r_k \otimes s_k$$

$$f^{\frac{i+1}{3}} = POD(\tilde{f}^{\frac{i+1}{3}}, \eta)$$

Computational Complexity:

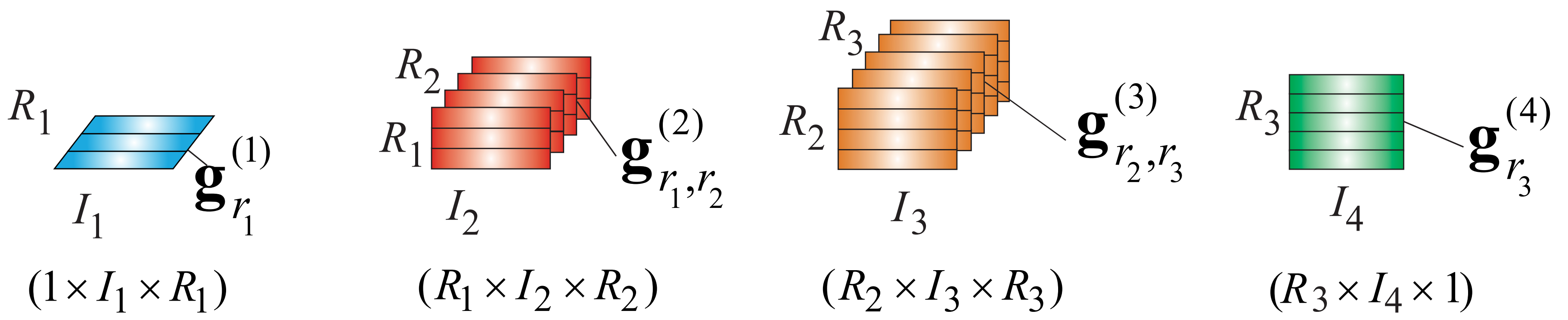
- Modified PGD step: solved by Alternated Least Square

  at iteration $l = K$ of the fixed-point PGD: $\mathcal{O}((N_x + N_v)(n + K)(d + 1))$

- POD step, solved by QR+SVD

  recompression step: $\mathcal{O}(n^3 + n^2(N_x + N_v))$

Tensor Train:



$$u(x_1, x_2, x_3, x_4) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} r_i^{(1)}(x_1) G_{ij}^{(2)}(x_2) G_{jk}^{(3)}(x_3) r_k^{(1)}(x_4)$$

- Successive, decreasing size SVD and reshaping
- Stable, used in Physics, Machine Learning
- Storage: $\mathcal{N} = n_1 I_1 + n_2 n_3 I_2 + n_2 n_3 I_3 + n_4 I_4$