3. Computing OT for data sciences

- Reality check
- Regularizations
- Entropic regularization

What matters for practitioners?

i.i.d samples $\boldsymbol{x_1, \dots, x_n} \sim \boldsymbol{\mu}$, $\boldsymbol{y_1, \dots, y_m} \sim \boldsymbol{\nu}$,

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{n}} \stackrel{\text{def}}{=} \frac{1}{n}\sum_i \delta_{\boldsymbol{x_i}}, \hat{\boldsymbol{\nu}}_{\boldsymbol{m}} \stackrel{\text{def}}{=} \frac{1}{m}\sum_j \delta_{\boldsymbol{y_j}}$$

Computational properties

Compute/approximate $\boldsymbol{W_p}(\hat{\boldsymbol{\mu}}_{\boldsymbol{n}}, \hat{\boldsymbol{\nu}}_{\boldsymbol{m}})$?

Statistical properties

$$\mathbb{E}\left[|\boldsymbol{W_p}(\boldsymbol{\mu}, \boldsymbol{\nu}) - \boldsymbol{W_p}(\hat{\boldsymbol{\mu}}_{\boldsymbol{n}}, \hat{\boldsymbol{\nu}}_{\boldsymbol{m}})|\right] \leq f(n, m)?$$

What matters for practitioners?

i.i.d samples $\boldsymbol{x_1, \dots, x_n} \sim \boldsymbol{\mu}$, $\boldsymbol{y_1, \dots, y_m} \sim \boldsymbol{\nu}$,

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{n}} \stackrel{\text{def}}{=} \frac{1}{n} \sum_i \delta_{\boldsymbol{x_i}}, \hat{\boldsymbol{\nu}}_{\boldsymbol{m}} \stackrel{\text{def}}{=} \frac{1}{m} \sum_j \delta_{\boldsymbol{y_j}}$$

Computational properties

$$\boldsymbol{O((n+m)nm \log(n+m))}$$

Statistical properties

$$\mathbb{E}\left[|\boldsymbol{W_p}(\boldsymbol{\mu}, \boldsymbol{\nu}) - \boldsymbol{W_p}(\hat{\boldsymbol{\mu}}_{\boldsymbol{n}}, \hat{\boldsymbol{\nu}}_{\boldsymbol{m}})|\right] \leq f(n, m)?$$

Sample Complexity

$$\text{If } \Omega = \mathbb{R}^d, d > 3$$

$$\mathbb{E}\left[\left|\boldsymbol{W_p}(\boldsymbol{\mu}, \boldsymbol{\nu}) - \boldsymbol{W_p}(\hat{\boldsymbol{\mu}}_{\boldsymbol{n}}, \hat{\boldsymbol{\nu}}_{\boldsymbol{n}})\right|\right] = O(n^{-1/d})$$

- [**Dudley'69**][**Dereich+'11**][**Fournier+'13**] *& others..*
- [**Weed+'17**]: sharper results when measures' support has "low effective *d*" in metric spaces
- [**Weed+'19**] for smooth densities
- **Lower bounds:** optimal quantization error.

From theory to practice ?

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{n}} \stackrel{\text{def}}{=} \frac{1}{n}\sum_i \delta_{\boldsymbol{x_i}}, \hat{\boldsymbol{\nu}}_{\boldsymbol{m}} \stackrel{\text{def}}{=} \frac{1}{m}\sum_j \delta_{\boldsymbol{y_j}}$$

Computational properties

$$\boldsymbol{O((n+m)nm\log(n+m))}$$

Statistical properties

$$\mathbb{E}\left[|\boldsymbol{W_p}(\boldsymbol{\mu},\boldsymbol{\nu}) - \boldsymbol{W_p}(\hat{\boldsymbol{\mu}}_{\boldsymbol{n}},\hat{\boldsymbol{\nu}}_{\boldsymbol{n}})|\right] = O(n^{-1/d})$$

From theory to practice ?

$$\hat{\mu}_n \stackrel{\text{def}}{=} \frac{1}{n}\sum_i \delta_{x_i}, \hat{\nu}_m \stackrel{\text{def}}{=} \frac{1}{m}\sum_j \delta_{y_j}$$

Computational properties

$$O((n+m)nm\log(n+m))$$

Statistical properties

$$\mathbb{E}\left[|W_p(\mu,\nu) - W_p(\hat{\mu}_n,\hat{\nu}_n)|\right] = O(n^{-1/d})$$

If we use OT in data sciences we <u>must</u> regularize the problem to improve on both aspects!

Many ways to regularize (dual) OT

$$\sup_{\boldsymbol{\varphi}(x)+\boldsymbol{\psi}(y)\leq \boldsymbol{c}(x,y)} \int \boldsymbol{\varphi} d\boldsymbol{\mu} + \int \boldsymbol{\psi} d\boldsymbol{\nu}.$$

Many ways to regularize (dual) OT

$$\sup_{\boldsymbol{\varphi}(x)+\boldsymbol{\psi}(y)\leq \boldsymbol{c}(x,y)} \int \boldsymbol{\varphi} d\boldsymbol{\mu} + \int \boldsymbol{\psi} d\boldsymbol{\nu}.$$

- RKHS for potentials **[GCBP'16]**, dualize/smooth indicator constraint

Many ways to regularize (dual) OT

$$\sup_{\varphi(x)+\psi(y)\leq c(x,y)} \int \varphi d\mu + \int \psi d\nu.$$

- RKHS for potentials **[GCBP'16]**, dualize/smooth indicator constraint

$$W_1(\mu,\nu) = \sup_{\varphi \text{ 1-Lipschitz}} \int \varphi(d\mu - d\nu).$$

- Parameterize functions using ReLU Deep net with bounded weights **[ACB'17]** or use Wavelet decompositions **[Shirdhonkar+'08]** for low *d*.

Many ways to regularize (primal) OT

$$\inf_{\boldsymbol{P}\in\Pi(\boldsymbol{\mu},\boldsymbol{\nu})} \iint \boldsymbol{c}(x,y)\boldsymbol{P}(dx,dy).$$

Many ways to regularize (primal) OT

$$\inf_{\boldsymbol{P}\in\Pi(\boldsymbol{\mu},\boldsymbol{\nu})} \iint \boxed{\boldsymbol{c}(x,y)}\boldsymbol{P}(dx,dy).$$

- Change cost function: threshold metric [**Pele+'09**], use geodesic distance on graphs [**Beckman'52**] [**Lin+'07**], [**Solomon+'14**], simplifies the LP.

Many ways to regularize (primal) OT

$$\inf_{\boldsymbol{P} \in \Pi(\boldsymbol{\mu},\boldsymbol{\nu})} \iint \boldsymbol{c}(x,y)\boldsymbol{P}(dx,dy).$$

- Change cost function: threshold metric [**Pele+'09**], use geodesic distance on graphs [**Beckman'52**] [**Lin+'07**], [**Solomon+'14**], simplifies the LP.

- Change measures: Quantize first [**Canas+'12**]; use Gaussians [**Gelbrich'92**]; projections on random lines [**Rabin+'11**] & *k*-dimensional subspaces [**PC'19**][**Weed+'19**].

Many ways to regularize (primal) OT

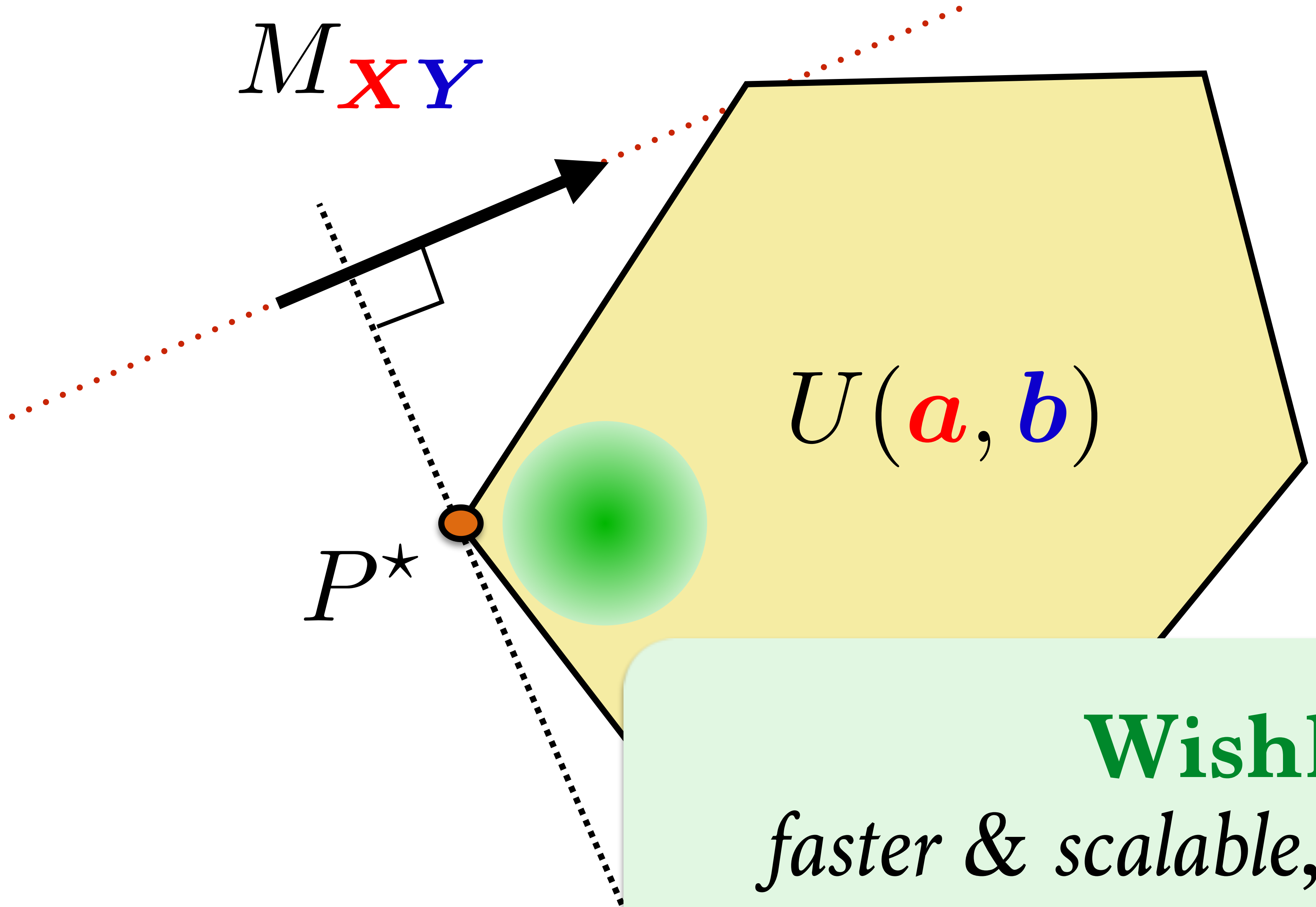
$$\inf_{\boldsymbol{P}\in\Pi(\boldsymbol{\mu},\boldsymbol{\nu})} \iint \boldsymbol{c}(x,y)\boldsymbol{P}(dx,dy).$$

- Change cost function: threshold metric [**Pele+'09**], use geodesic distance on graphs [**Beckman'52**] [**Lin+'07**], [**Solomon+'14**], simplifies the LP.

- Change measures: Quantize first [**Canas+'12**]; use Gaussians [**Gelbrich'92**]; projections on random lines [**Rabin+'11**] & k-dimensional subspaces [**PC'19**][**Weed+'19**].

- Add prior on coupling, e.g. (entropic) regularization [C'13][GP'16][GCBCP'19][JMPC'20]

Regularization on the Primal



Wishlist:
faster & *scalable*, more *stable*,
(automatically) differentiable

Entropic Regularization [Wilson'69]

Def. Regularized Wasserstein, $\gamma \geq 0$

$$W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \min_{\boldsymbol{P} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{P}, M_{\boldsymbol{X}\boldsymbol{Y}} \rangle - \gamma E(\boldsymbol{P})$$

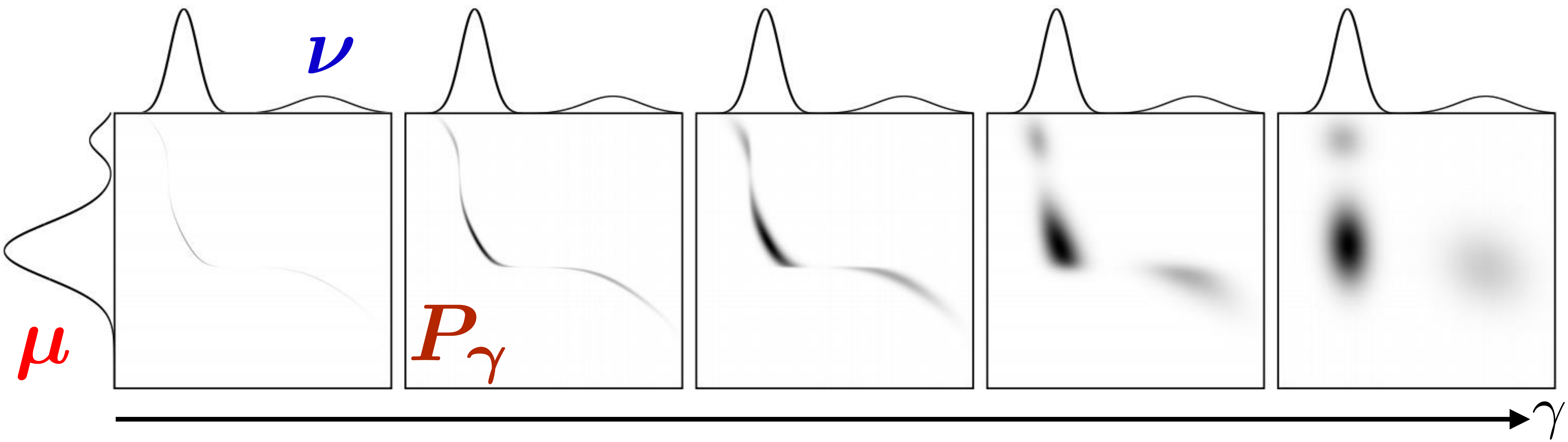
$$E(P) \stackrel{\text{def}}{=} - \sum_{i,j=1}^{nm} P_{ij}(\log P_{ij} - 1)$$

Note: Unique optimal solution because of strong concavity of entropy

Entropic Regularization [Wilson'69]

Def. Regularized Wasserstein, $\gamma \geq 0$

$$W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \min_{\boldsymbol{P} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{P}, M_{\mathbf{XY}} \rangle - \gamma E(\boldsymbol{P})$$

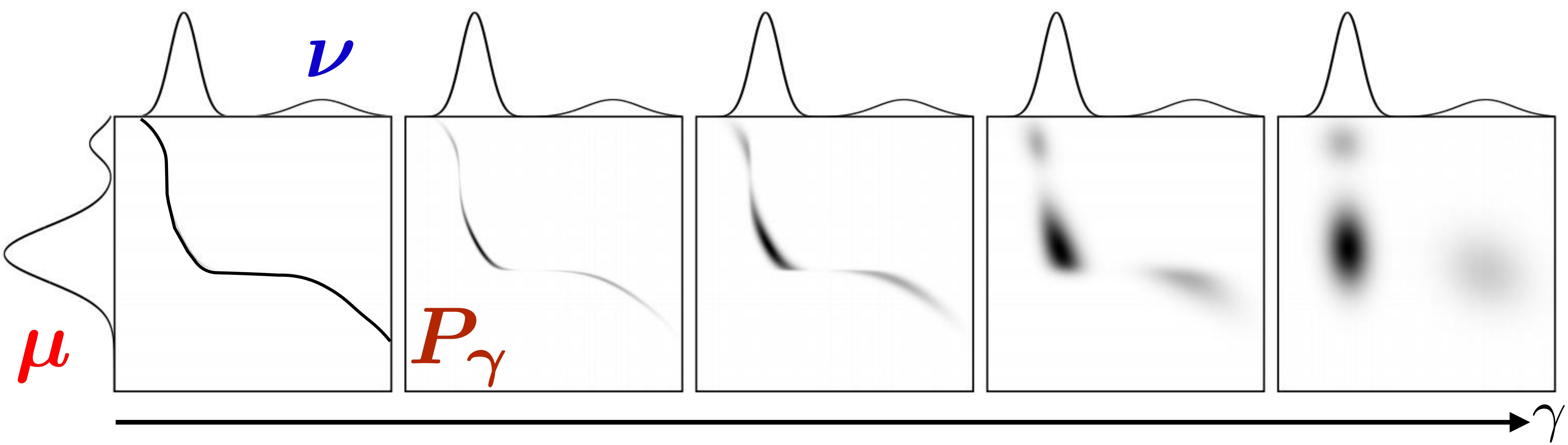


Note: Unique optimal solution because of strong concavity of entropy

Entropic Regularization [Wilson'69]

Def. Regularized Wasserstein, $\gamma \geq 0$

$$W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \min_{\boldsymbol{P} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle - \gamma E(\boldsymbol{P})$$

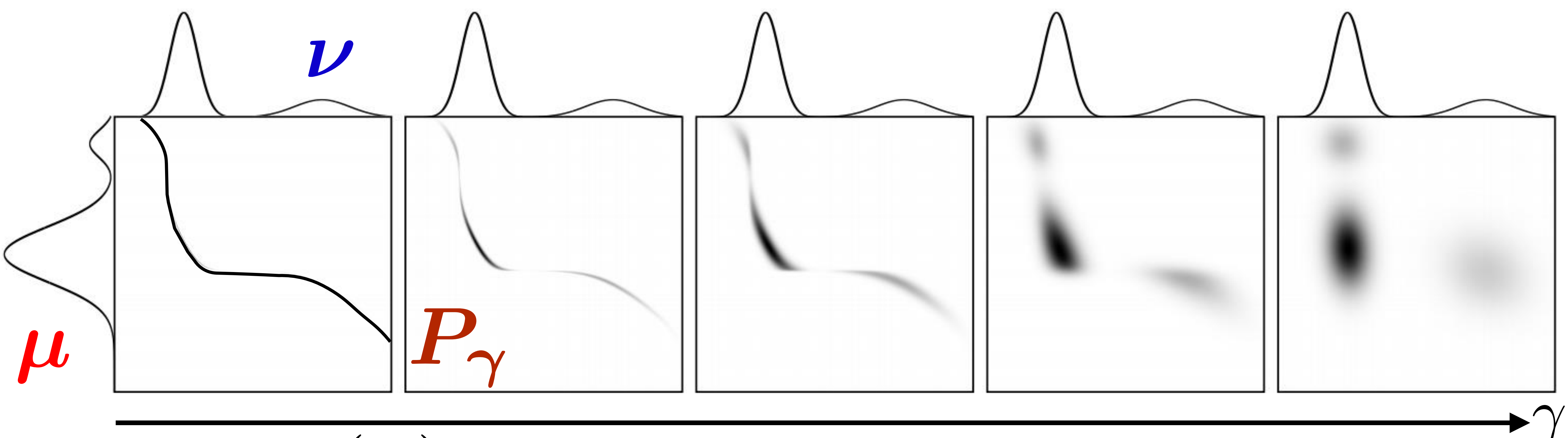


Note: Unique optimal solution because of strong concavity of entropy

Entropic Regularization [Wilson'69]

Def. Regularized Wasserstein, $\gamma \geq 0$

$$W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \min_{\boldsymbol{P} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{P}, M_{\mathbf{XY}} \rangle - \gamma E(\boldsymbol{P})$$



$\approx$ "$\boldsymbol{y} = \boldsymbol{T}(\boldsymbol{x})$"

Note: Unique optimal solution because of strong concavity of entropy

Fast & Scalable Algorithm

Prop. If $P_\gamma \stackrel{\text{def}}{=} \operatorname*{argmin}_{\boldsymbol{P} \in U(\boldsymbol{a},\boldsymbol{b})} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle - \gamma E(\boldsymbol{P})$

then $\exists! \boldsymbol{u} \in \mathbb{R}_+^n, \boldsymbol{v} \in \mathbb{R}_+^m$, such that

$$P_\gamma = \mathbf{diag}(\boldsymbol{u}) K \mathbf{diag}(\boldsymbol{v}), \quad K \stackrel{\text{def}}{=} e^{-M_{\boldsymbol{XY}}/\gamma}$$

Fast & Scalable Algorithm

Prop. If $P_\gamma \stackrel{\text{def}}{=} \underset{\boldsymbol{P}\in U(\boldsymbol{a},\boldsymbol{b})}{\text{argmin}} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle - \gamma E(\boldsymbol{P})$

then $\exists! \boldsymbol{u} \in \mathbb{R}^n_+, \boldsymbol{v} \in \mathbb{R}^m_+$, such that

$$P_\gamma = \textbf{diag}(\boldsymbol{u}) K \textbf{diag}(\boldsymbol{v}), \quad K \stackrel{\text{def}}{=} e^{-M_{\boldsymbol{XY}}/\gamma}$$

$$L(P,\alpha,\beta) = \boxed{\sum_{ij} P_{ij} M_{ij} + \gamma P_{ij}(\log P_{ij} - 1)} + \boxed{\alpha^T(P\mathbf{1} - \boldsymbol{a}) + \beta^T(P^T\mathbf{1} - \boldsymbol{b})}$$

$$\partial L/\partial P_{ij} = M_{ij} + \gamma \log P_{ij} + \alpha_i + \beta_j$$

$$(\boldsymbol{\partial L/\partial P_{ij}} = 0) \Rightarrow P_{ij} = \boldsymbol{e}^{\frac{\boldsymbol{\alpha_i}}{\gamma}} e^{-\frac{M_{ij}}{\gamma}} \boldsymbol{e}^{\frac{\boldsymbol{\beta_j}}{\gamma}} = \boldsymbol{u_i} K_{ij} \boldsymbol{v_j}$$

Fast & Scalable Algorithm

Prop. If $P_\gamma \stackrel{\text{def}}{=} \operatorname*{argmin}_{\boldsymbol{P}\in U(\boldsymbol{a},\boldsymbol{b})} \langle \boldsymbol{P}, M_{\boldsymbol{X}\boldsymbol{Y}} \rangle - \gamma E(\boldsymbol{P})$

then $\exists! \boldsymbol{u} \in \mathbb{R}_+^n, \boldsymbol{v} \in \mathbb{R}_+^m$, such that

$$P_\gamma = \mathbf{diag}(\boldsymbol{u}) K \mathbf{diag}(\boldsymbol{v}), \quad K \stackrel{\text{def}}{=} e^{-M_{\boldsymbol{X}\boldsymbol{Y}}/\gamma}$$

$$P_\gamma \in U(\boldsymbol{a}, \boldsymbol{b}) \Leftrightarrow \begin{cases} \mathbf{diag}(\boldsymbol{u}) K \mathbf{diag}(\boldsymbol{v}) \mathbf{1}_m & = \boldsymbol{a} \\ \mathbf{diag}(\boldsymbol{v}) K^T \mathbf{diag}(\boldsymbol{u}) \mathbf{1}_n & = \boldsymbol{b} \end{cases}$$

Fast & Scalable Algorithm

Prop. If $P_\gamma \stackrel{\text{def}}{=} \underset{\boldsymbol{P}\in U(\boldsymbol{a},\boldsymbol{b})}{\operatorname{argmin}} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle - \gamma E(\boldsymbol{P})$

then $\exists! \boldsymbol{u} \in \mathbb{R}^n_+, \boldsymbol{v} \in \mathbb{R}^m_+$, such that

$$P_\gamma = \mathbf{diag}(\boldsymbol{u}) K \mathbf{diag}(\boldsymbol{v}), \quad K \stackrel{\text{def}}{=} e^{-M_{\boldsymbol{XY}}/\gamma}$$

$$P_\gamma \in U(\boldsymbol{a}, \boldsymbol{b}) \Leftrightarrow \begin{cases} \mathbf{diag}(\boldsymbol{u}) K \mathbf{diag}(\boldsymbol{v}) \mathbf{1}_m & = \boldsymbol{a} \\ \mathbf{diag}(\boldsymbol{v}) K^T \underbrace{\mathbf{diag}(\boldsymbol{u}) \mathbf{1}_n}_{\boldsymbol{u}} & = \boldsymbol{b} \end{cases}$$

Fast & Scalable Algorithm

Prop. If $P_\gamma \stackrel{\text{def}}{=} \underset{\boldsymbol{P}\in U(\boldsymbol{a},\boldsymbol{b})}{\operatorname{argmin}} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle - \gamma E(\boldsymbol{P})$

then $\exists! \boldsymbol{u} \in \mathbb{R}^n_+, \boldsymbol{v} \in \mathbb{R}^m_+$, such that

$$P_\gamma = \mathbf{diag}(\boldsymbol{u}) K \mathbf{diag}(\boldsymbol{v}), \quad K \stackrel{\text{def}}{=} e^{-M_{\boldsymbol{XY}}/\gamma}$$

$$P_\gamma \in U(\boldsymbol{a}, \boldsymbol{b}) \Leftrightarrow \begin{cases} \mathbf{diag}(\boldsymbol{u}) K \overbrace{\mathbf{diag}(\boldsymbol{v})\mathbf{1}_m}^{\boldsymbol{v}} & = \boldsymbol{a} \\ \mathbf{diag}(\boldsymbol{v}) K^T \underbrace{\mathbf{diag}(\boldsymbol{u})\mathbf{1}_n}_{\boldsymbol{u}} & = \boldsymbol{b} \end{cases}$$

Fast & Scalable Algorithm

Prop. If $P_\gamma \stackrel{\text{def}}{=} \underset{\boldsymbol{P}\in U(\boldsymbol{a},\boldsymbol{b})}{\operatorname{argmin}} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle - \gamma E(\boldsymbol{P})$

then $\exists! \boldsymbol{u} \in \mathbb{R}_+^n, \boldsymbol{v} \in \mathbb{R}_+^m$, such that

$$P_\gamma = \mathbf{diag}(\boldsymbol{u}) K \mathbf{diag}(\boldsymbol{v}), \quad K \stackrel{\text{def}}{=} e^{-M_{\boldsymbol{XY}}/\gamma}$$

$$P_\gamma \in U(\boldsymbol{a}, \boldsymbol{b}) \Leftrightarrow \begin{cases} \mathbf{diag}(\boldsymbol{u}) K \boldsymbol{v} & = \boldsymbol{a} \\ \mathbf{diag}(\boldsymbol{v}) K^T \boldsymbol{u} & = \boldsymbol{b} \end{cases}$$

Fast & Scalable Algorithm

Prop. If $P_\gamma \overset{\text{def}}{=} \underset{\boldsymbol{P} \in U(\boldsymbol{a},\boldsymbol{b})}{\operatorname{argmin}} \langle \boldsymbol{P}, M_{\boldsymbol{X}\boldsymbol{Y}} \rangle - \gamma E(\boldsymbol{P})$
then $\exists! \boldsymbol{u} \in \mathbb{R}_+^n, \boldsymbol{v} \in \mathbb{R}_+^m$, such that

$$P_\gamma = \mathbf{diag}(\boldsymbol{u}) K \mathbf{diag}(\boldsymbol{v}), \quad K \overset{\text{def}}{=} e^{-M_{\boldsymbol{X}\boldsymbol{Y}}/\gamma}$$

$$P_\gamma \in U(\boldsymbol{a}, \boldsymbol{b}) \Leftrightarrow \begin{cases} \boldsymbol{u} \odot K \boldsymbol{v} & = \boldsymbol{a} \\ \boldsymbol{v} \odot K^T \boldsymbol{u} & = \boldsymbol{b} \end{cases}$$

Fast & Scalable Algorithm

Prop. If $P_\gamma \stackrel{\text{def}}{=} \underset{\boldsymbol{P} \in U(\boldsymbol{a},\boldsymbol{b})}{\operatorname{argmin}} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle - \gamma E(\boldsymbol{P})$

then $\exists! \boldsymbol{u} \in \mathbb{R}_+^n, \boldsymbol{v} \in \mathbb{R}_+^m$, such that

$$P_\gamma = \mathbf{diag}(\boldsymbol{u}) K \mathbf{diag}(\boldsymbol{v}), \quad K \stackrel{\text{def}}{=} e^{-M_{\boldsymbol{XY}}/\gamma}$$

$$P_\gamma \in U(\boldsymbol{a}, \boldsymbol{b}) \Leftrightarrow \begin{cases} \boldsymbol{u} = \boldsymbol{a} / K\boldsymbol{v} \\ \boldsymbol{v} = \boldsymbol{b} / K^T \boldsymbol{u} \end{cases}$$

Fast & Scalable Algorithm

Sinkhorn's Algorithm : Repeat

1. $\boldsymbol{u} = \boldsymbol{a} / K\boldsymbol{v}$
2. $\boldsymbol{v} = \boldsymbol{b} / K^T\boldsymbol{u}$

Fast & Scalable Algorithm

Sinkhorn's Algorithm : Repeat

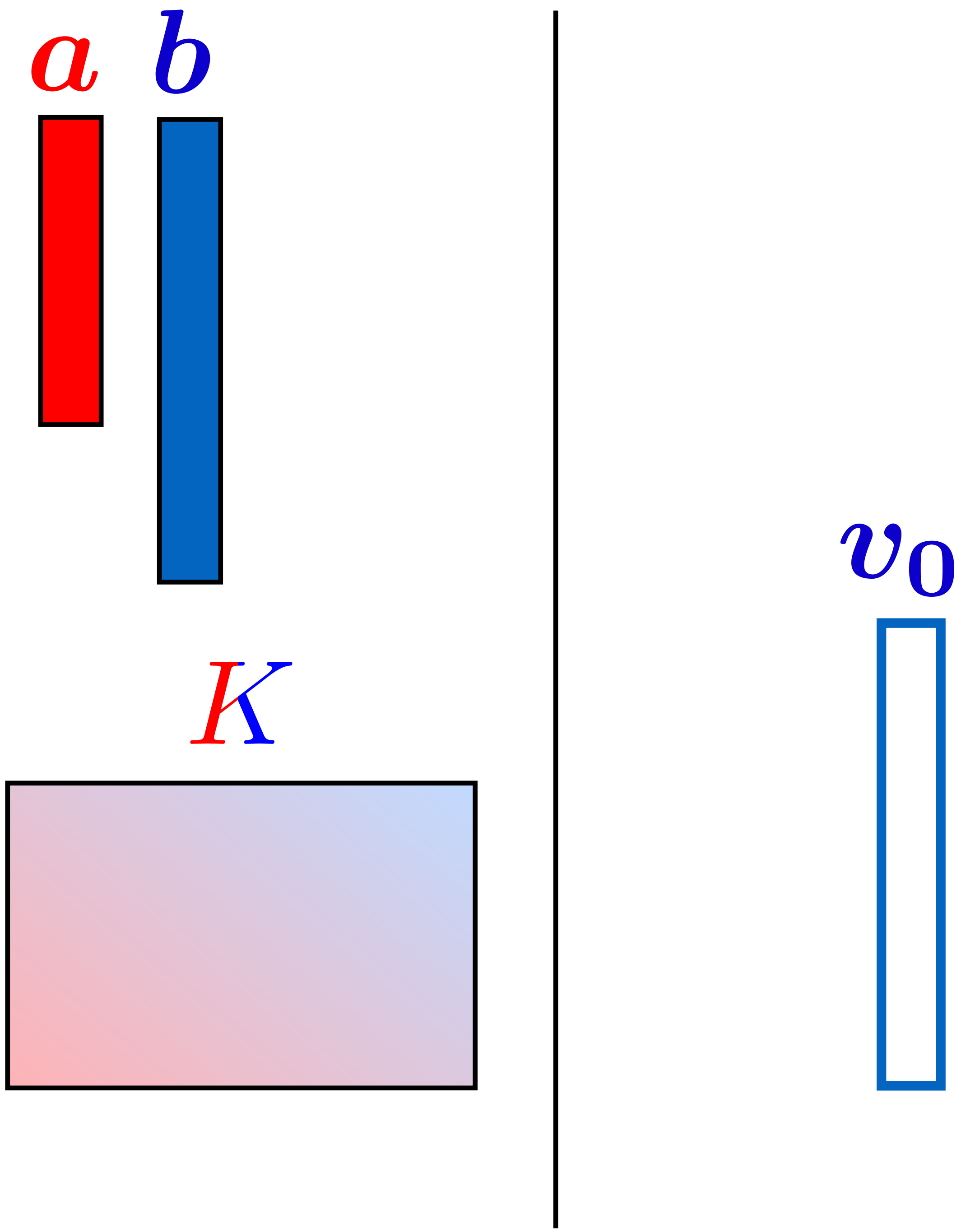
1. $\boldsymbol{u} = \boldsymbol{a} / K\boldsymbol{v}$
2. $\boldsymbol{v} = \boldsymbol{b} / K^T\boldsymbol{u}$

- [**Sinkhorn'64**] proved first convergence result [**Franklin+'89**] characterised linear convergence
- Recent wave of great results by [**Altschuler+'17**] [**Dvurechensky+18**][**Lin+19**]
- $O(nm)$ complexity, GPGPU parallel [**C'13**] .
- $O(n \log n)$ on gridded spaces using convolutions. [**Solomon'+15**]

Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

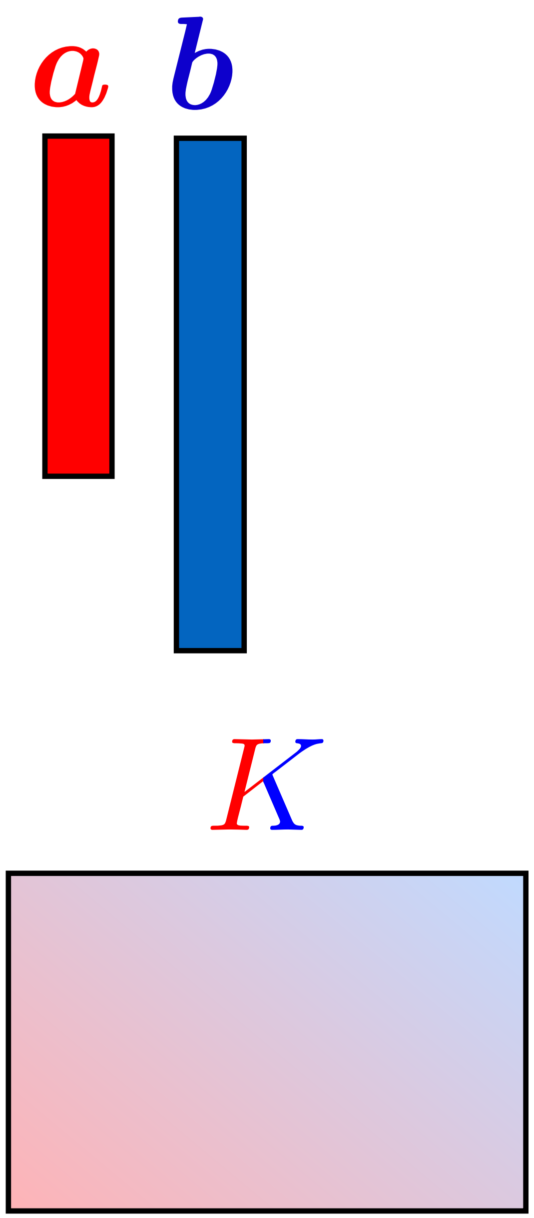
$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$

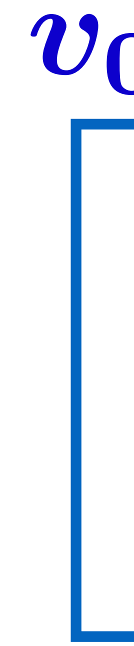


Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$

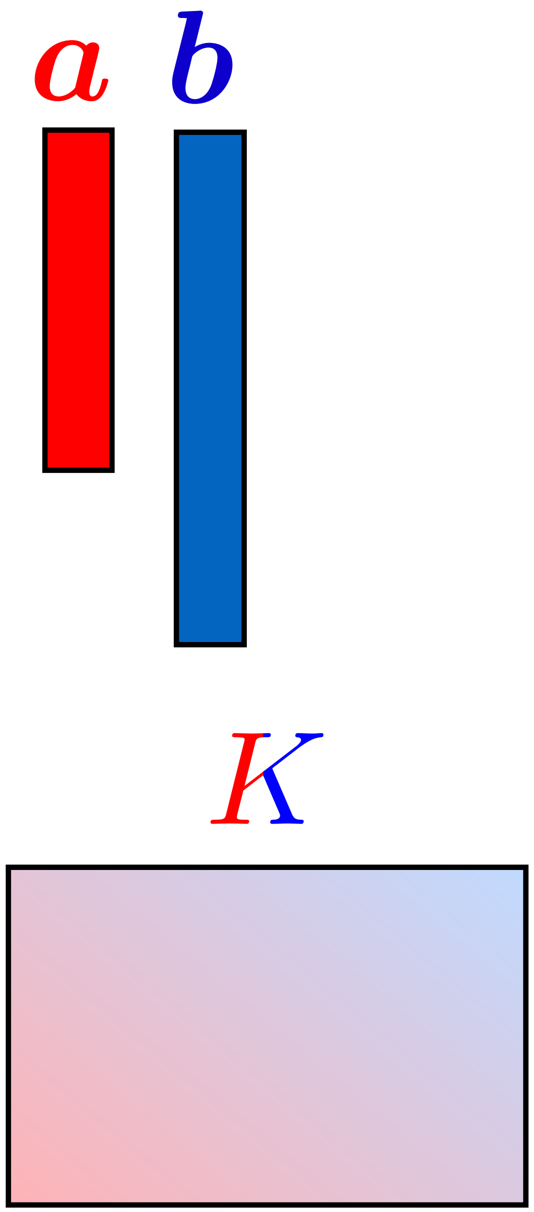


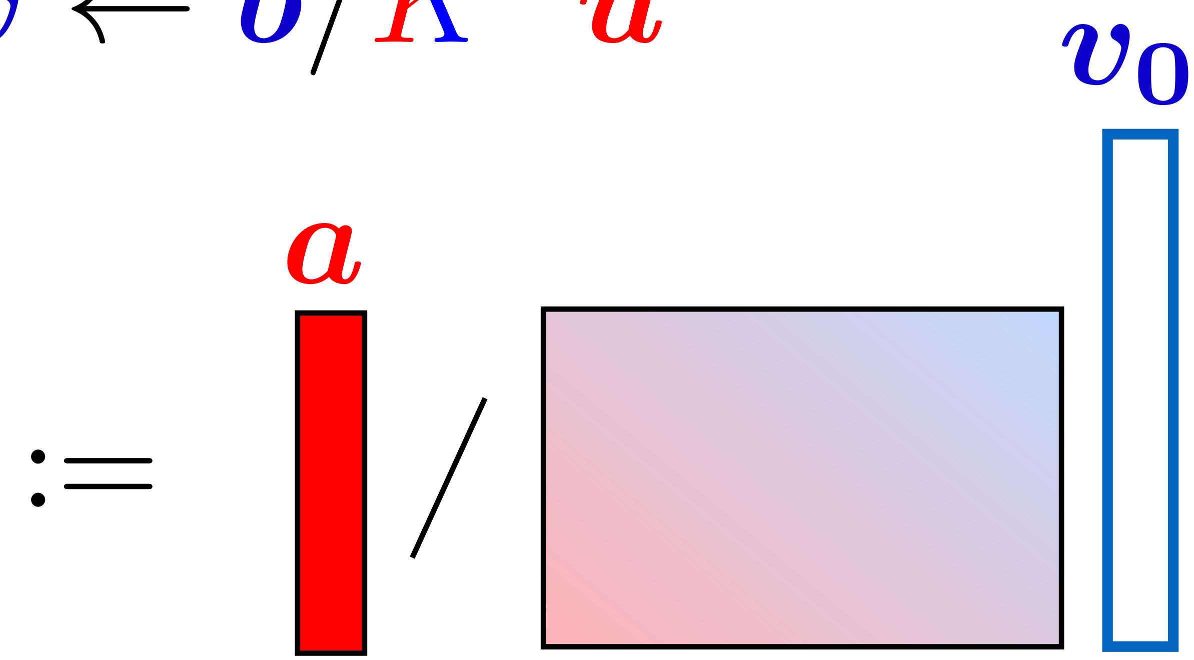


Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$

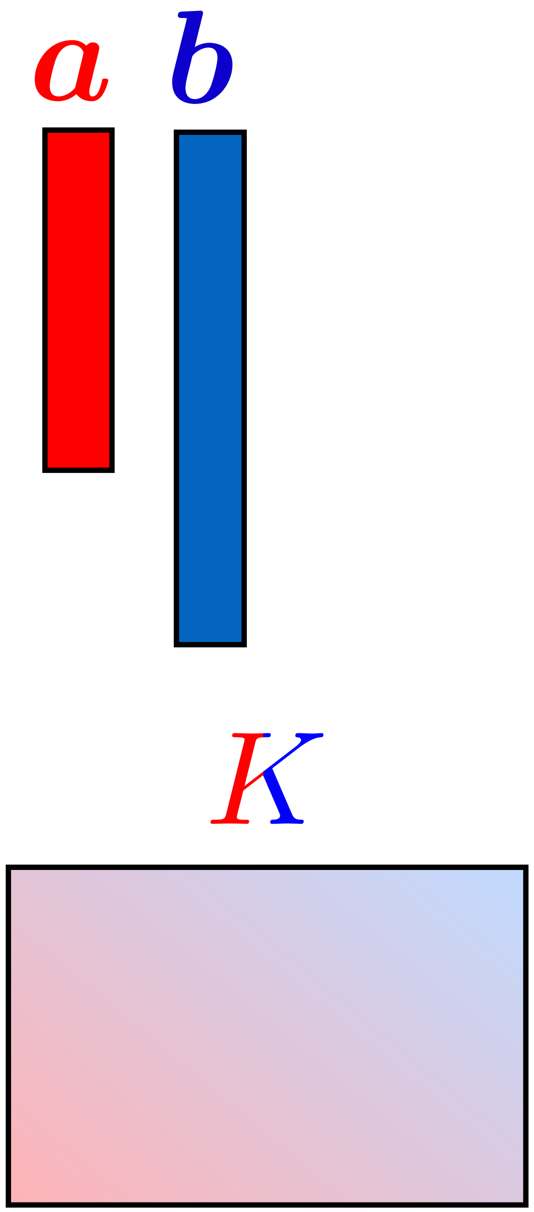


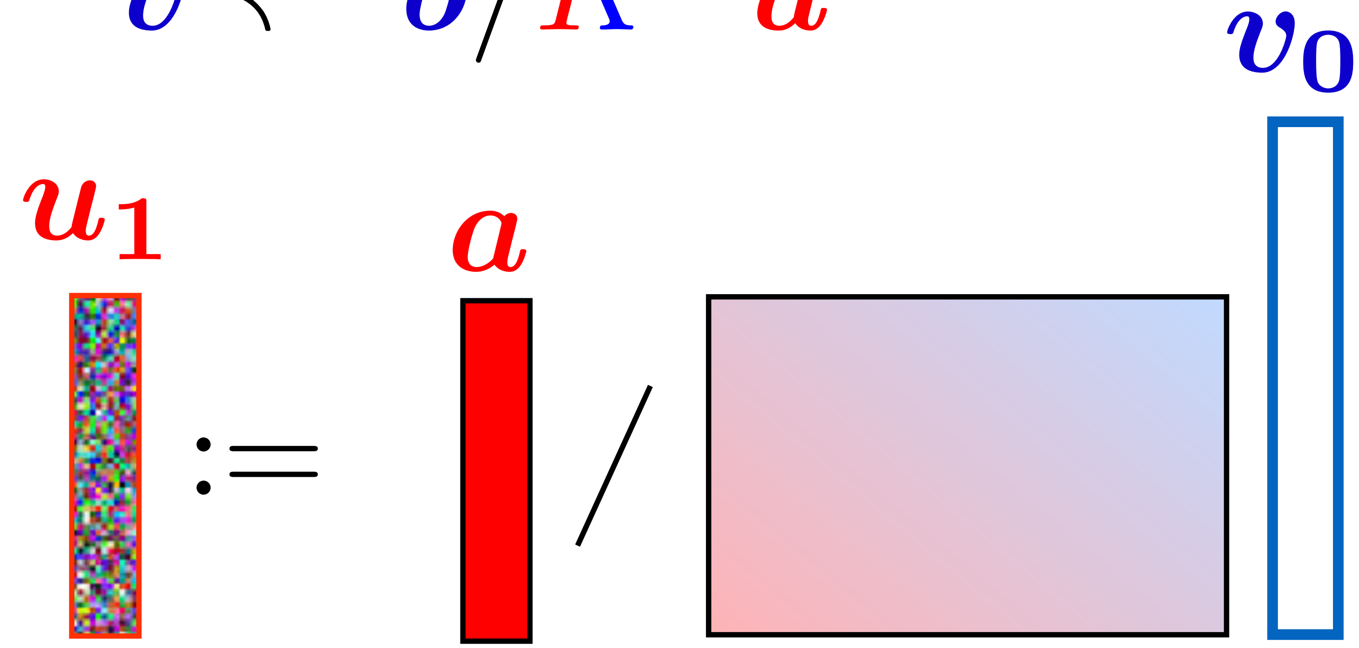


Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$

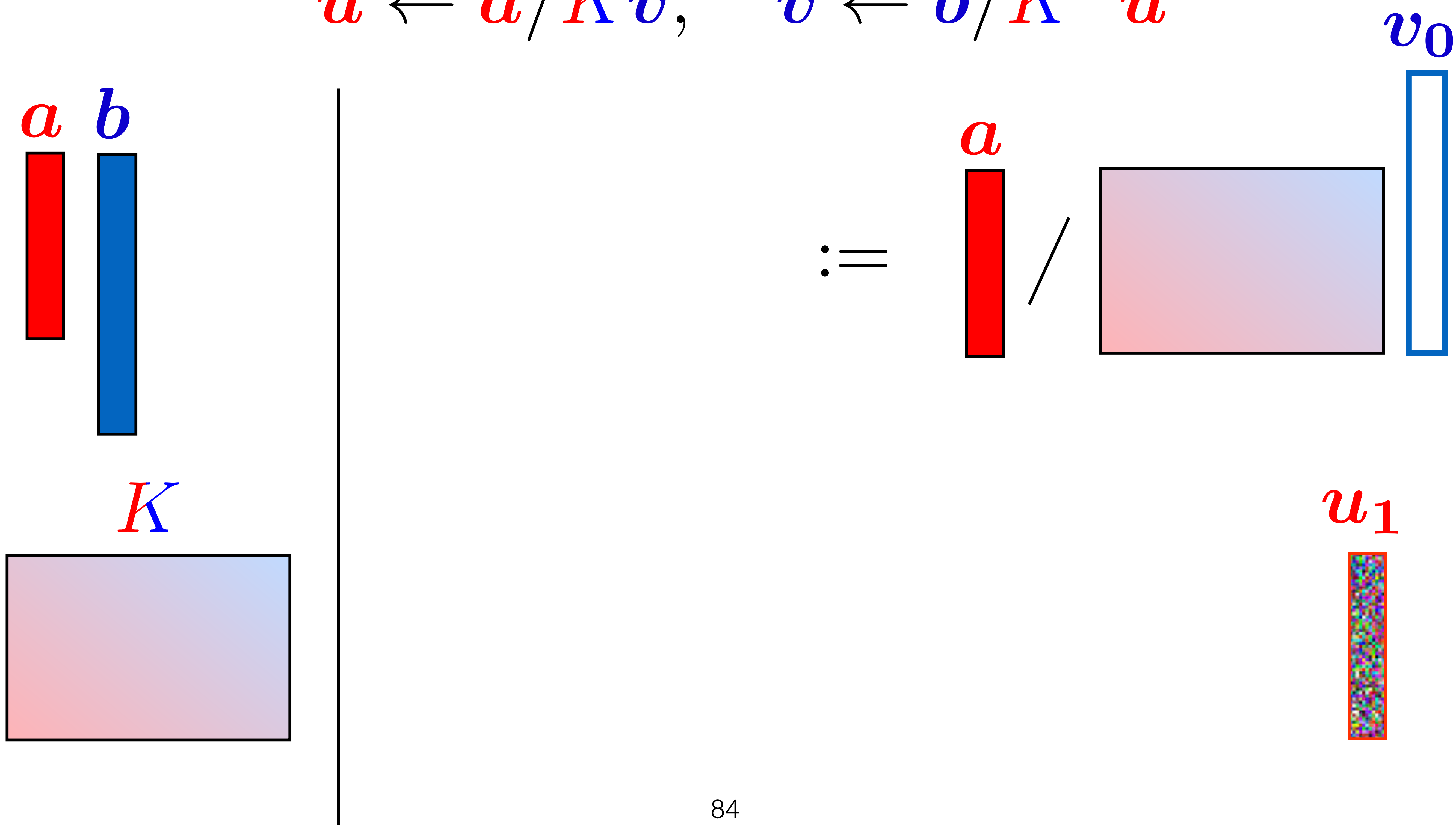




Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

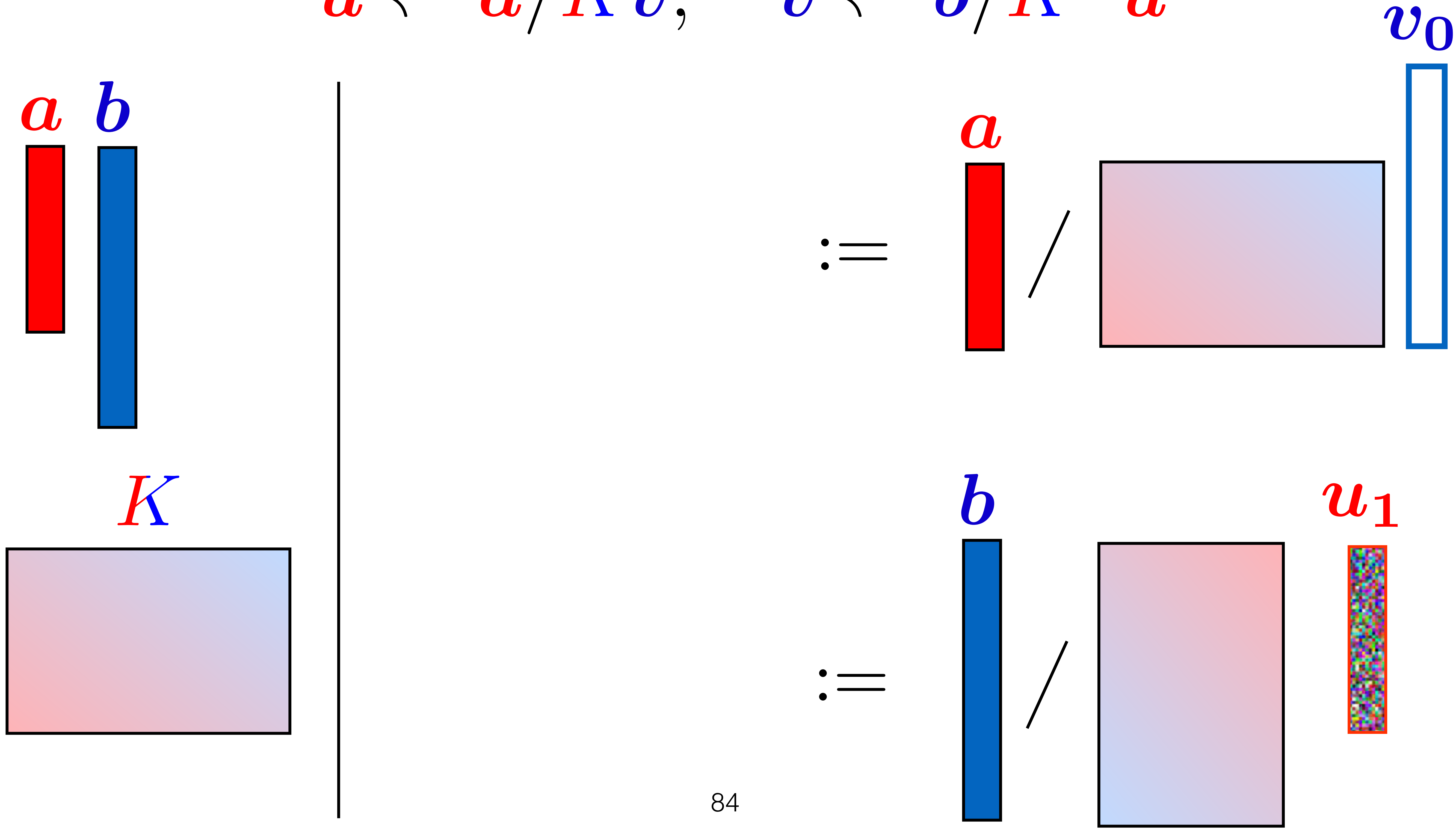
$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$



Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

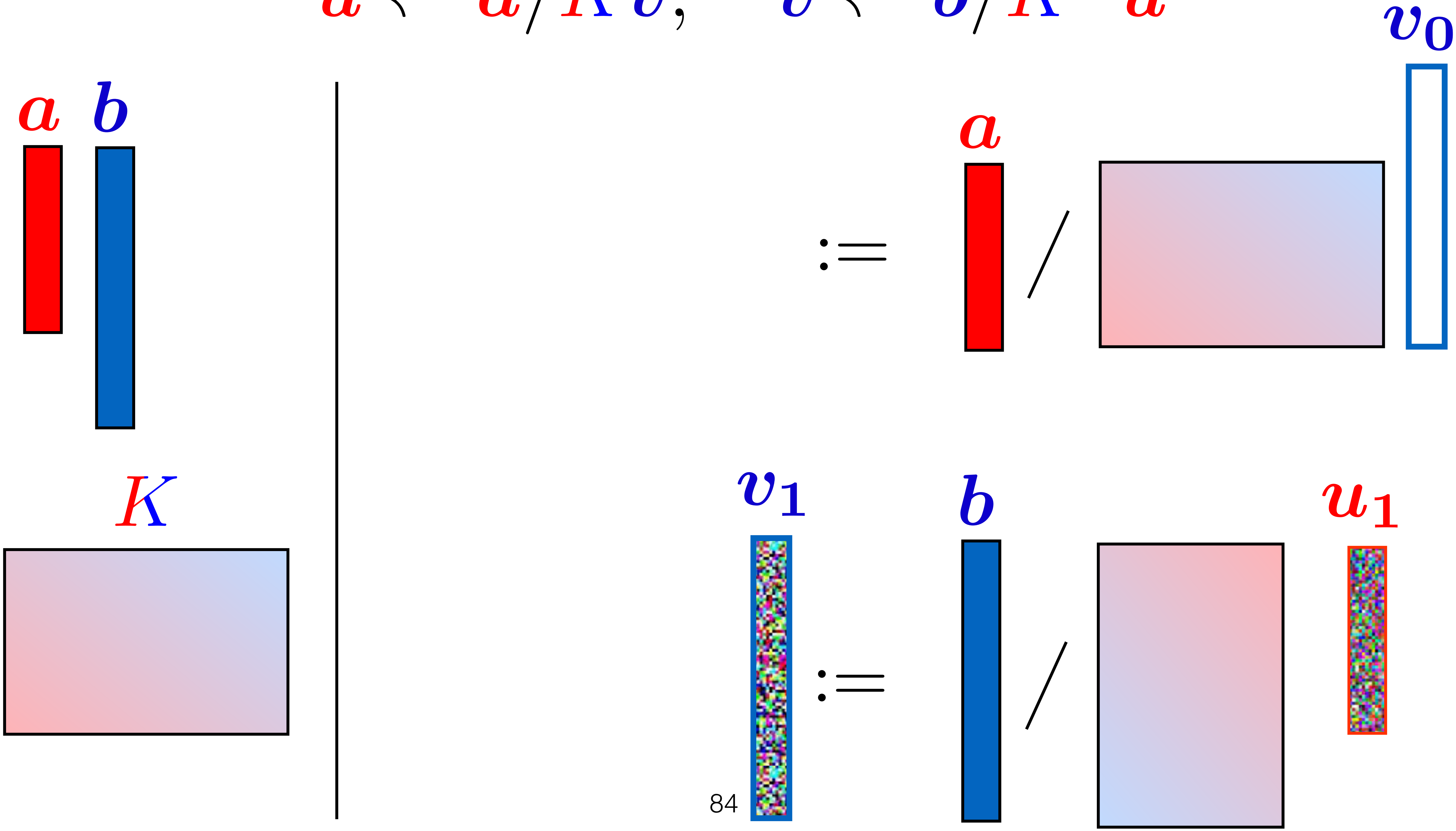
$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$



Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

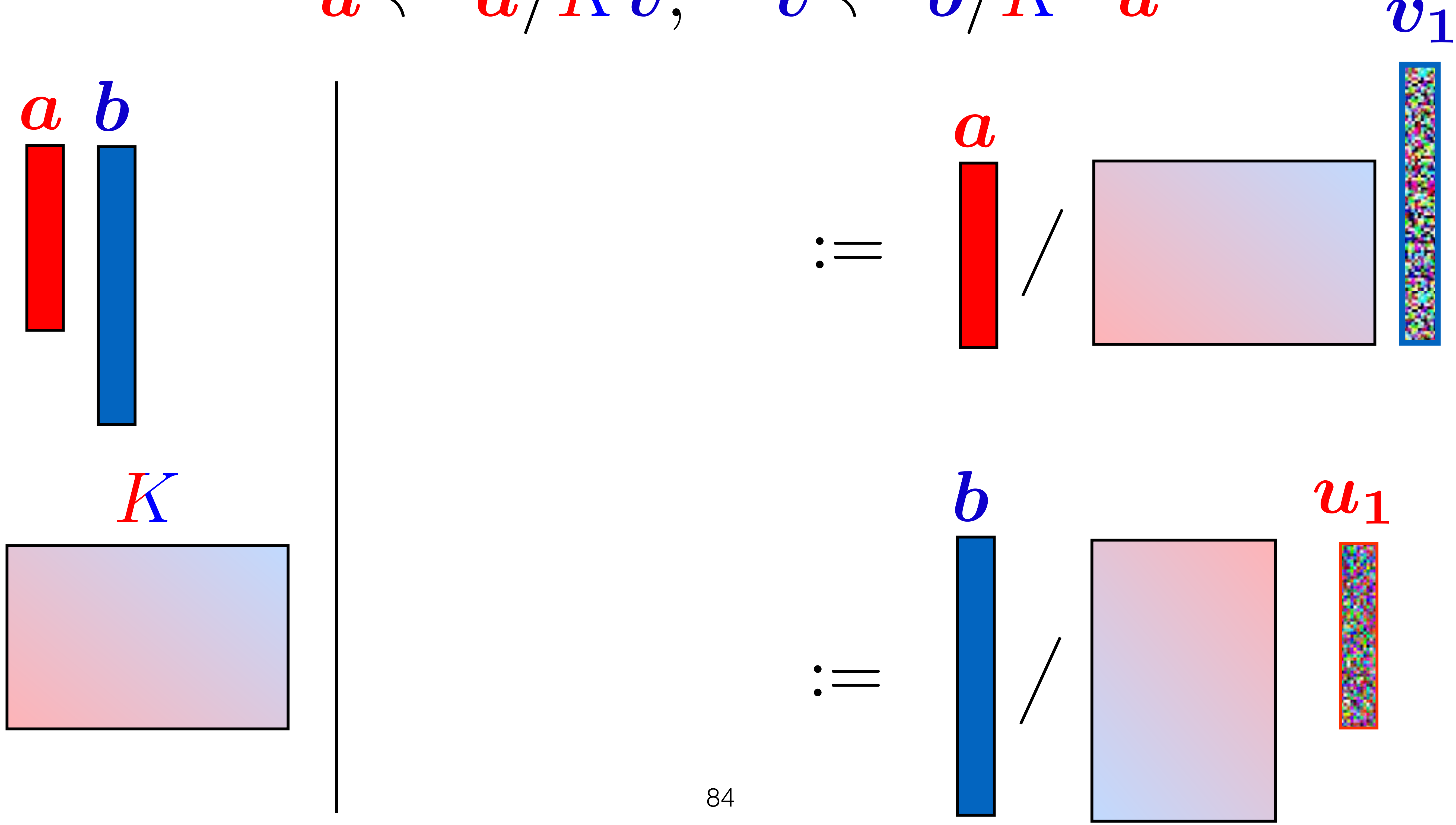
$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$



Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

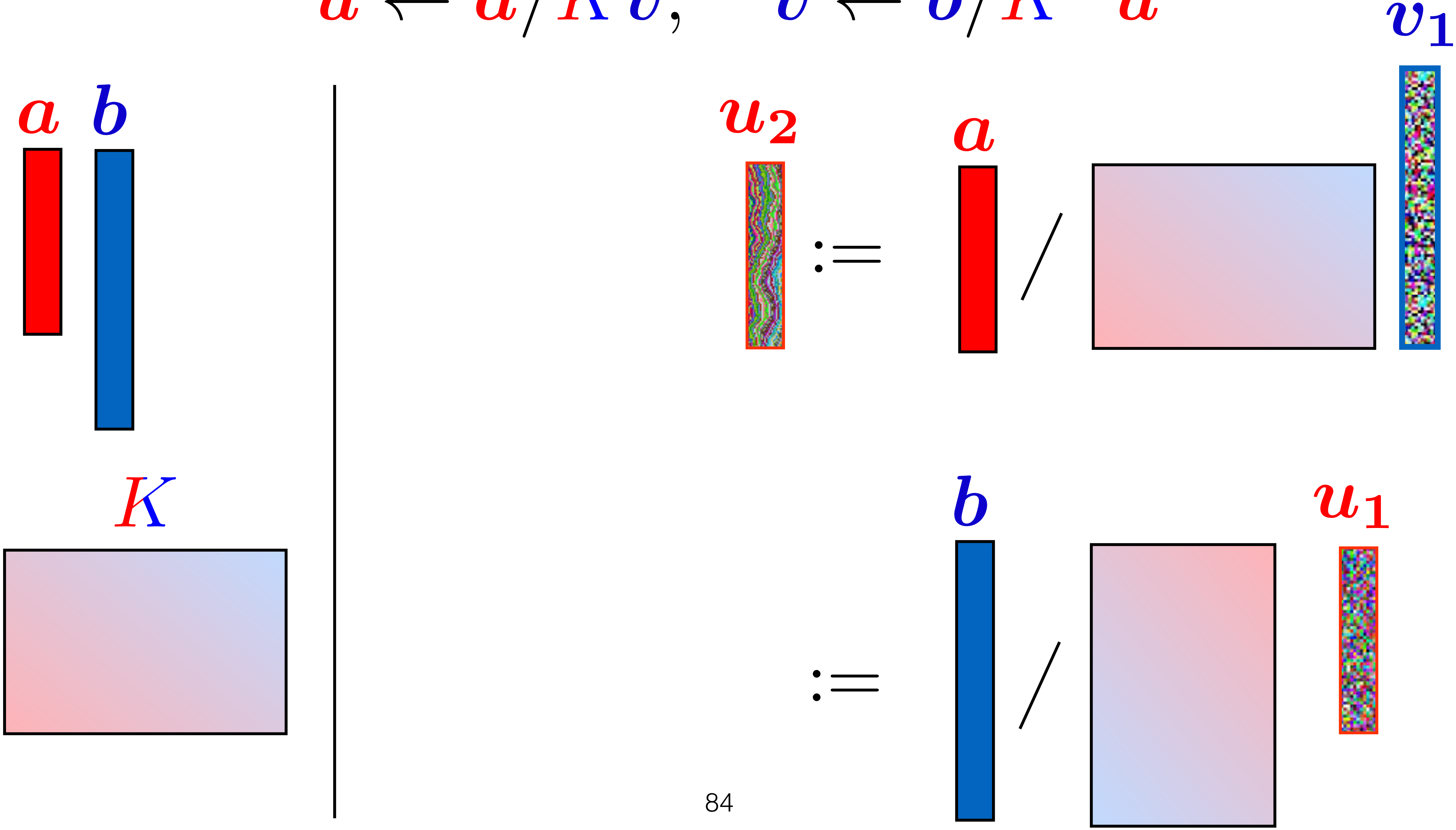
$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$



Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

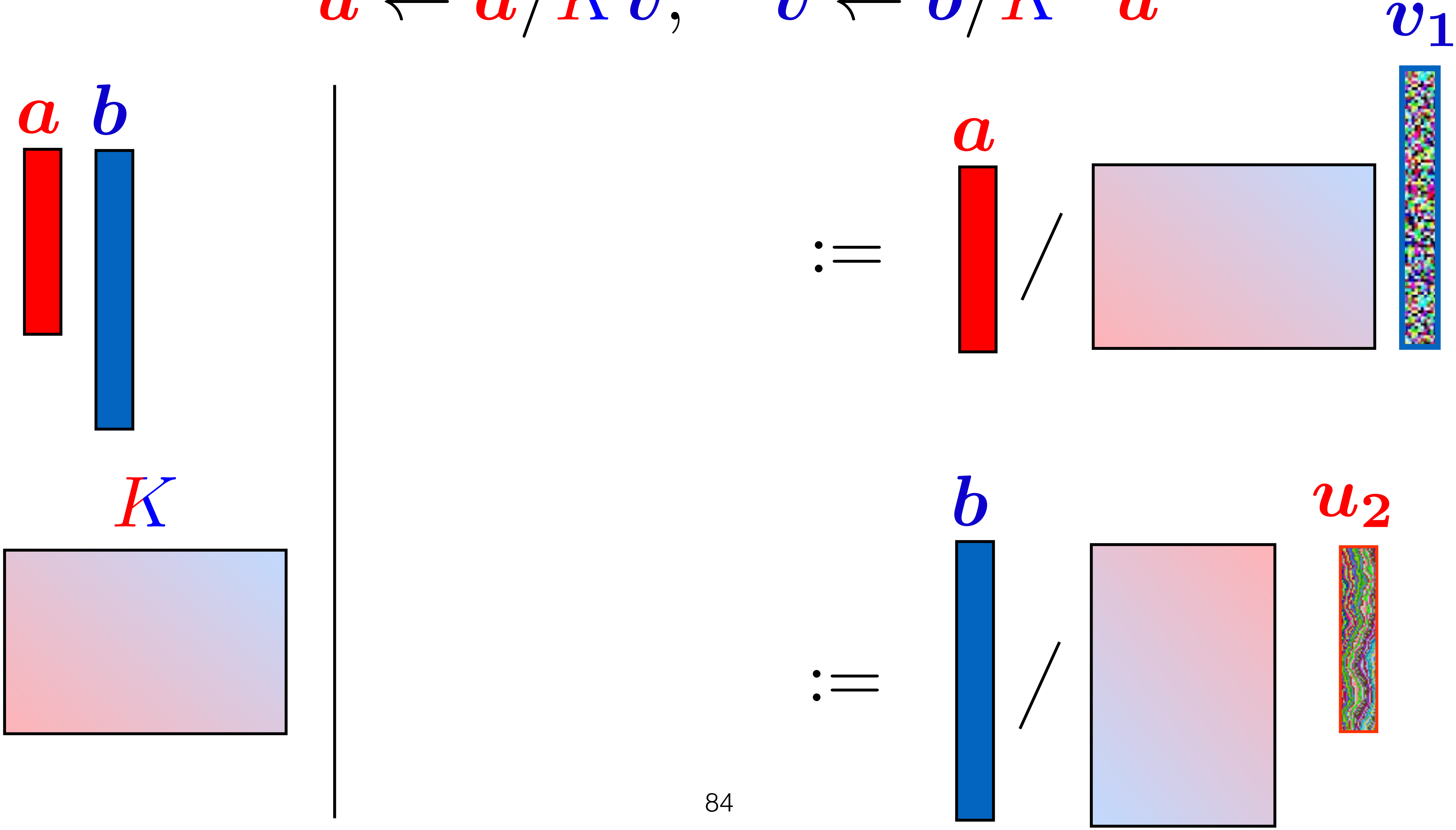
$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$



Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

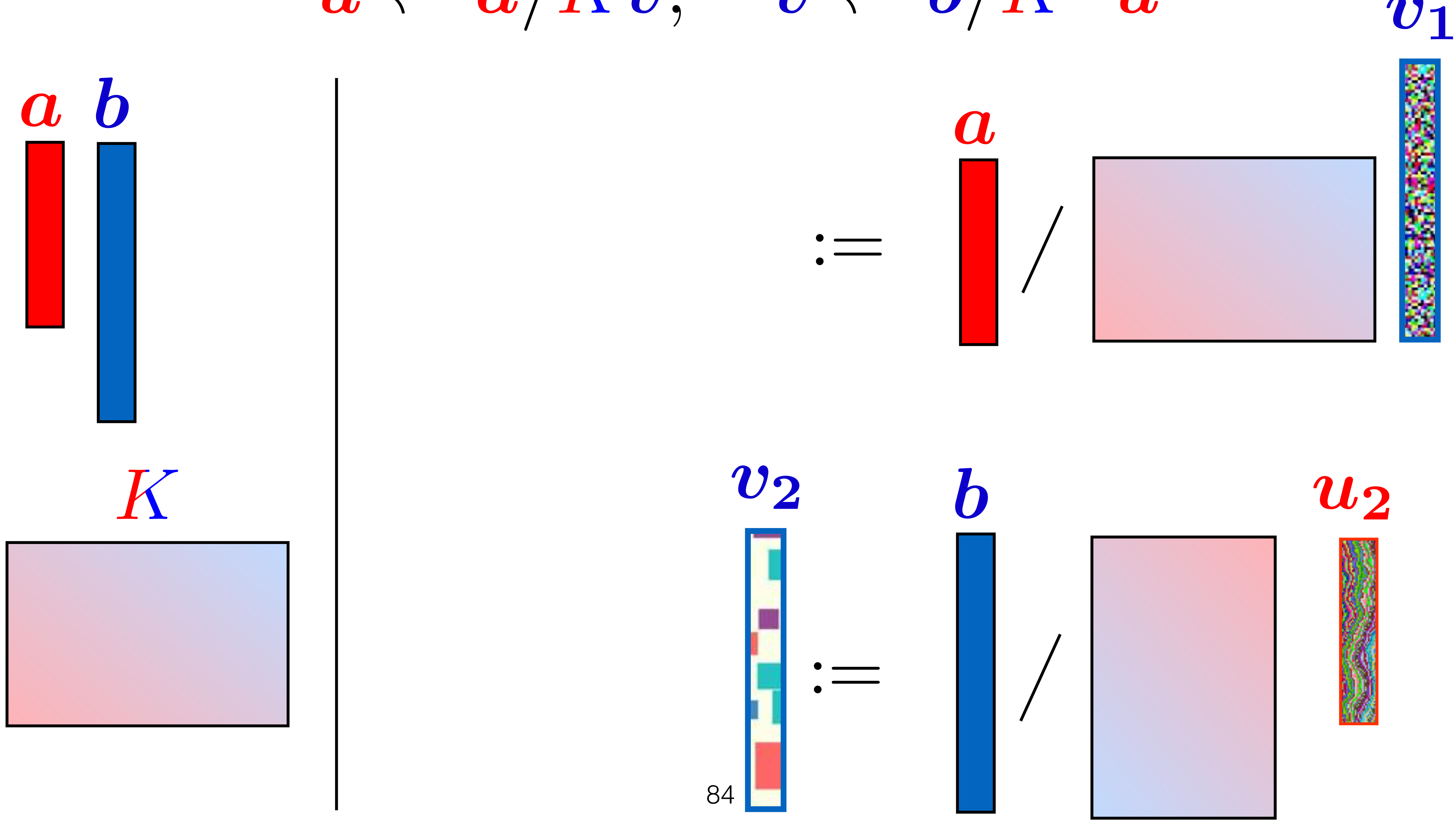
$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$



Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

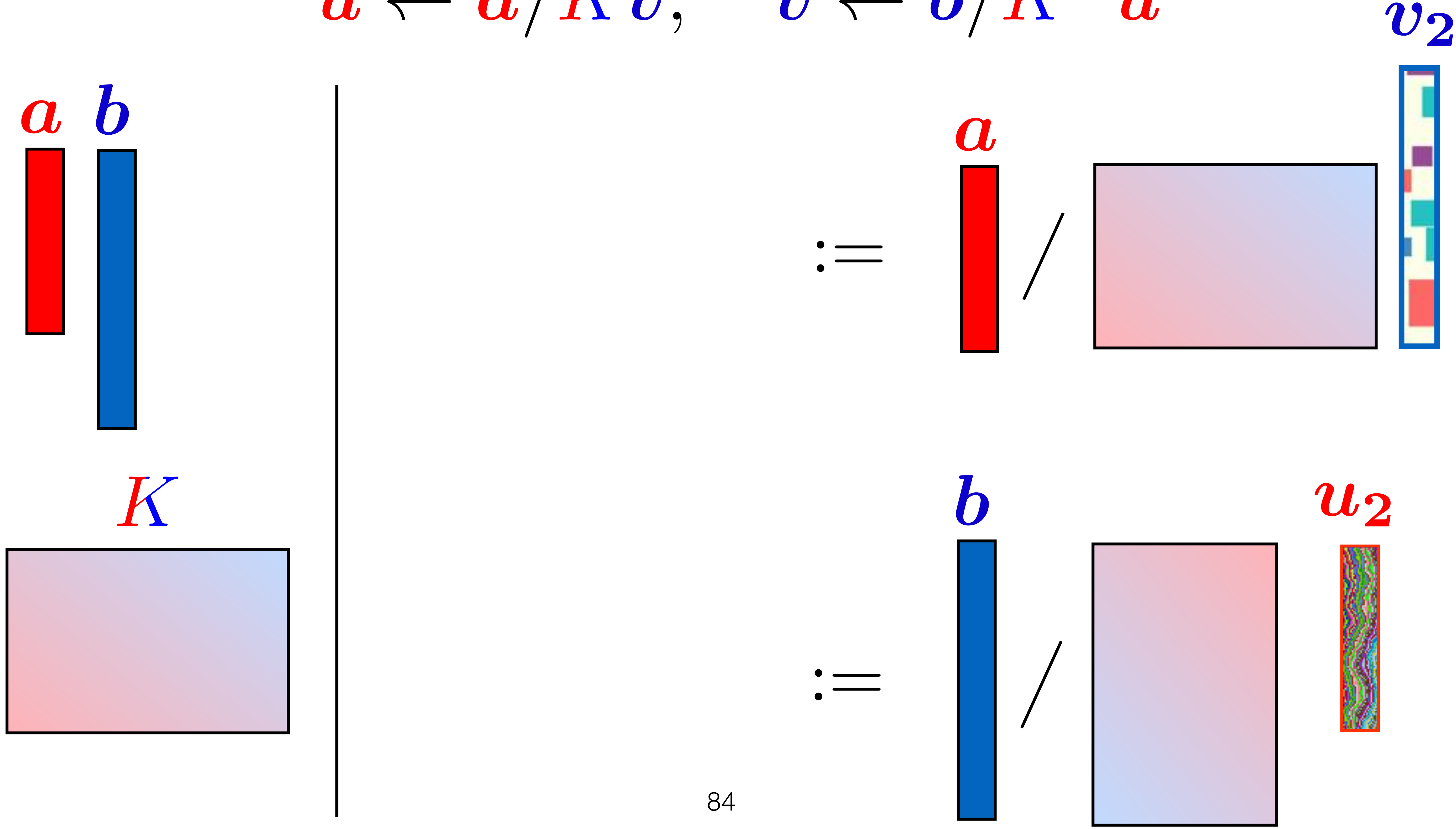
$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$



Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

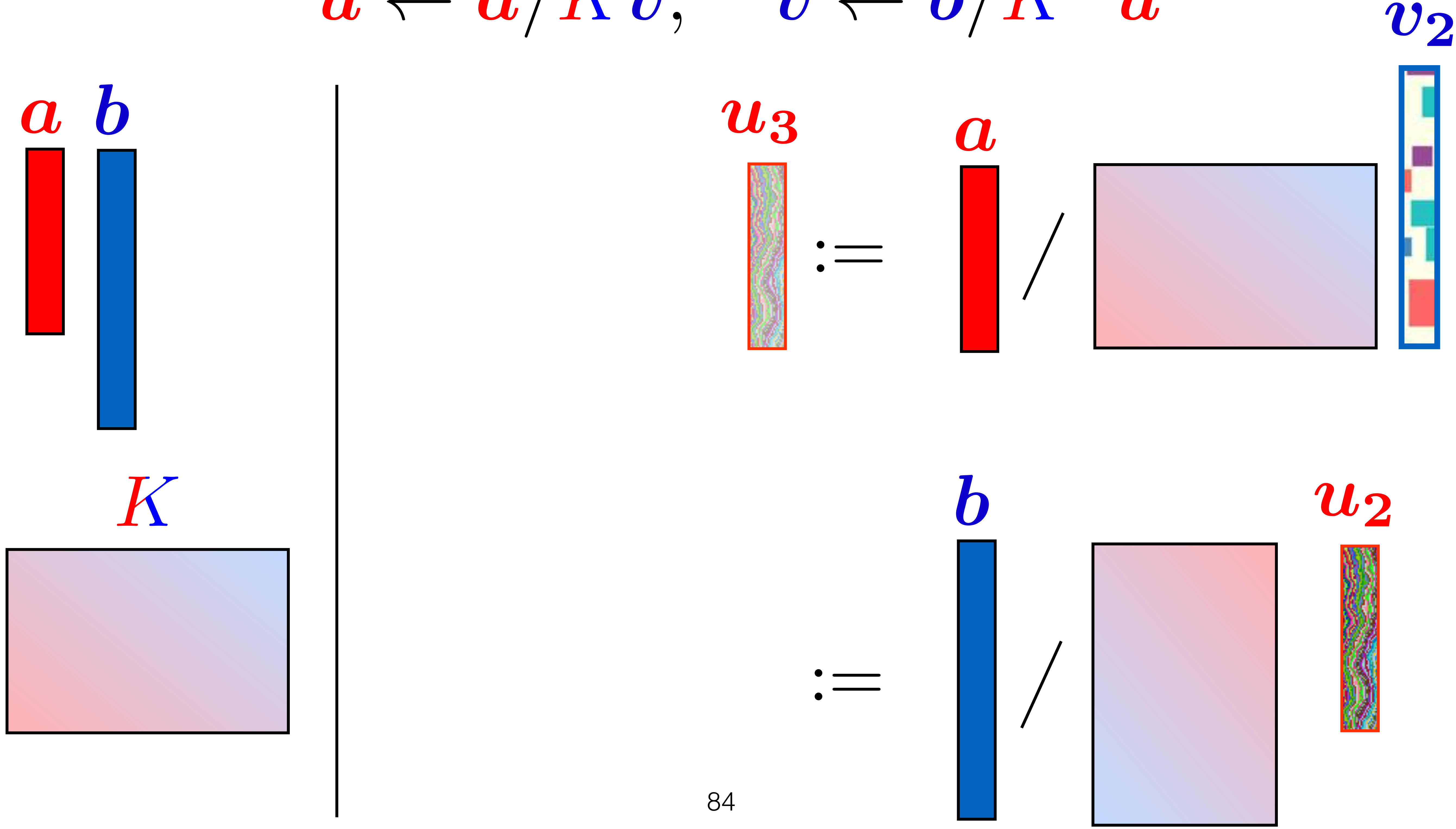
$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$



Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

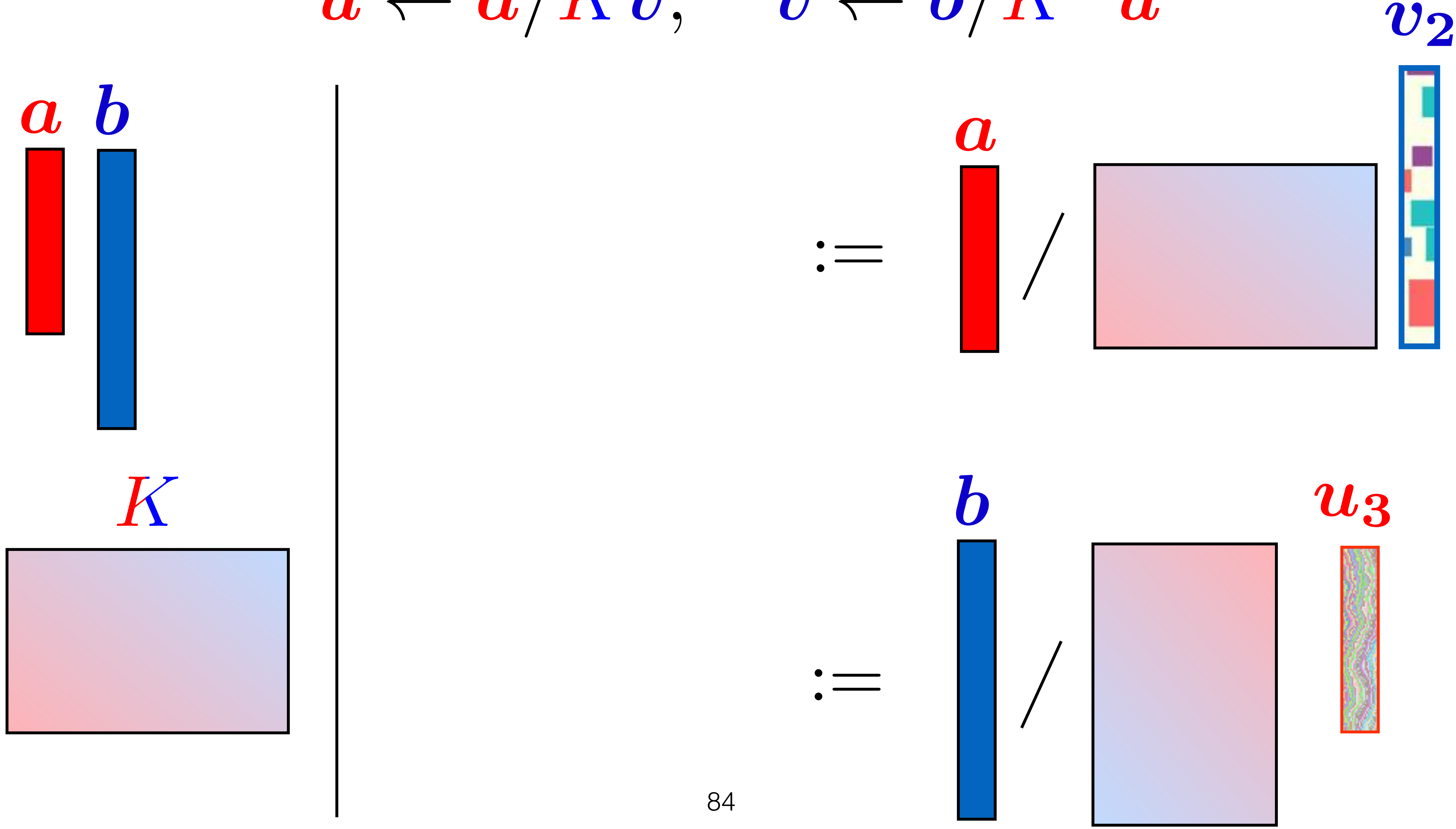
$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$



Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

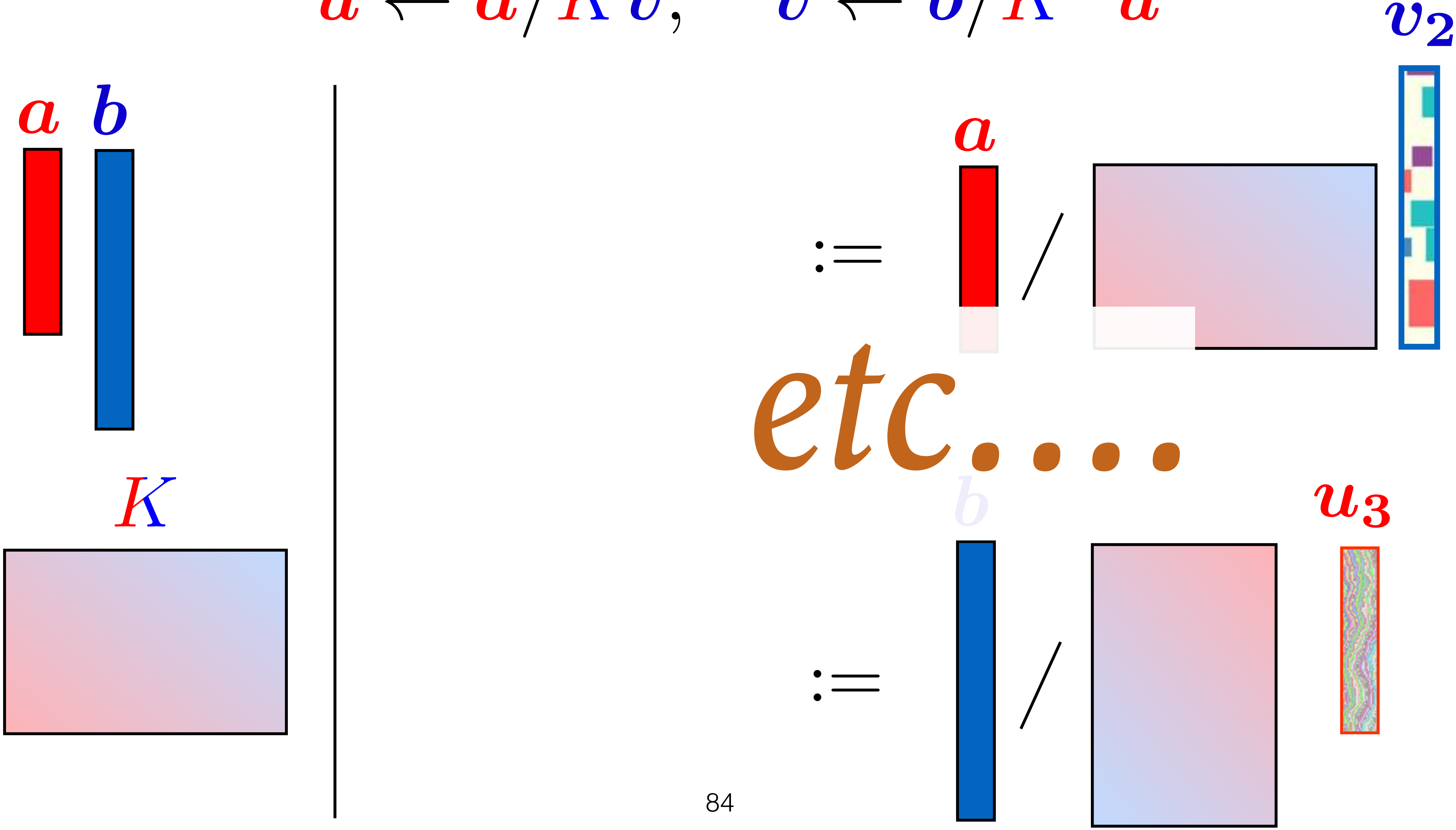
$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$



Fast & Scalable Algorithm

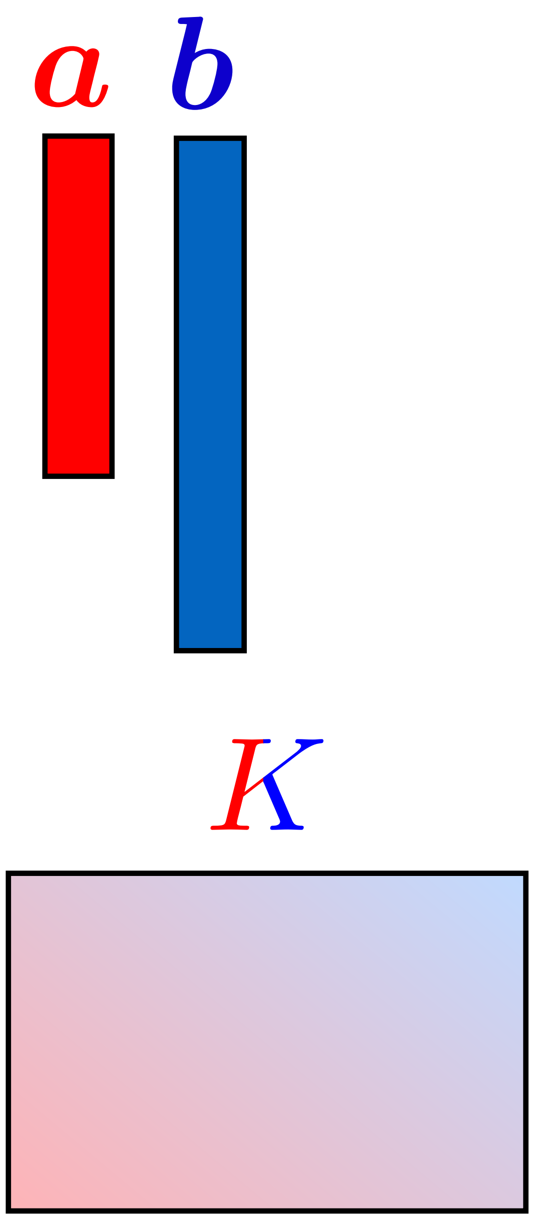
- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$

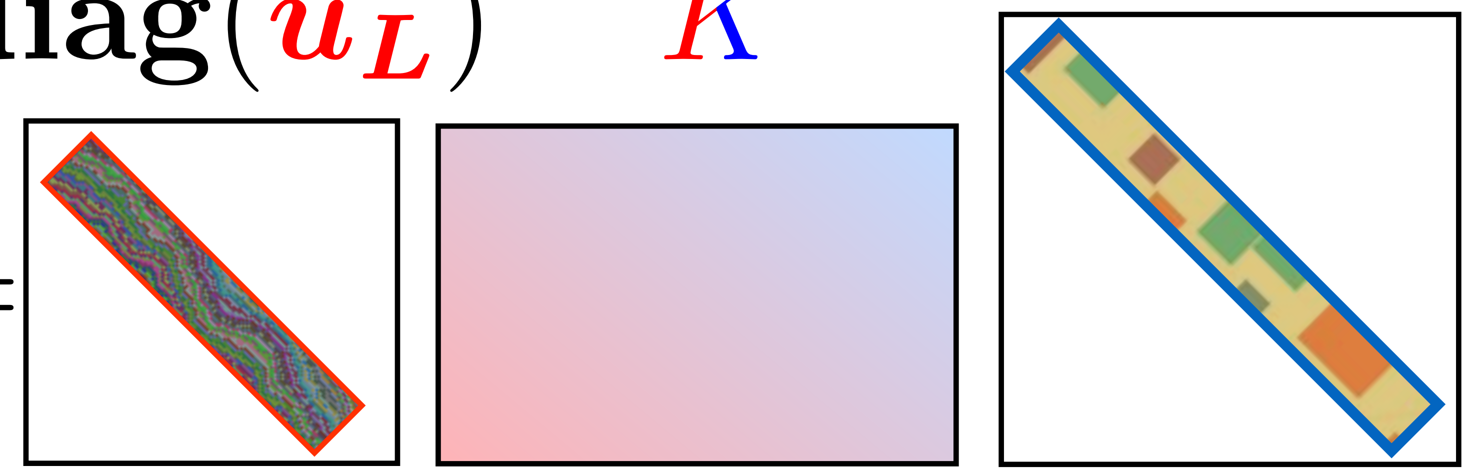


Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations.

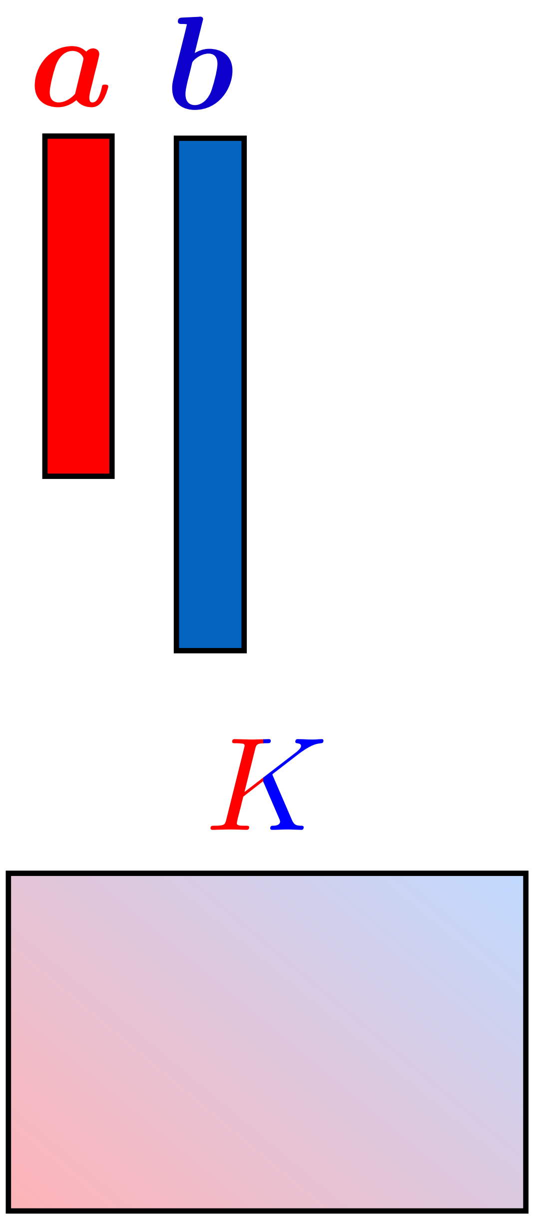


$$P_L := \mathbf{diag}(u_L)\, K\, \mathbf{diag}(v_L)$$

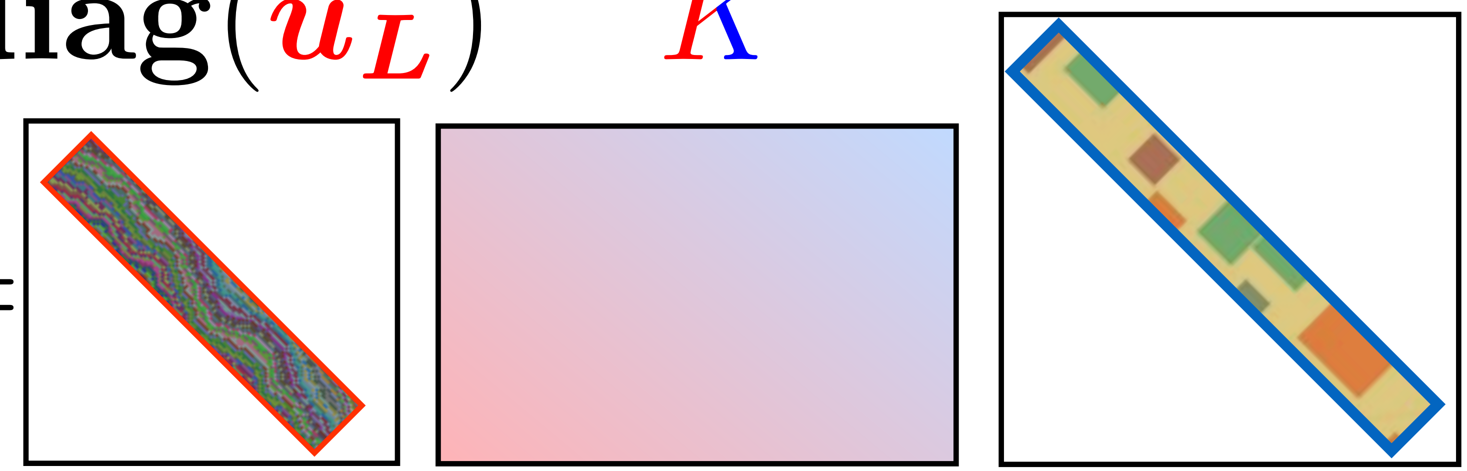


Fast & Scalable Algorithm

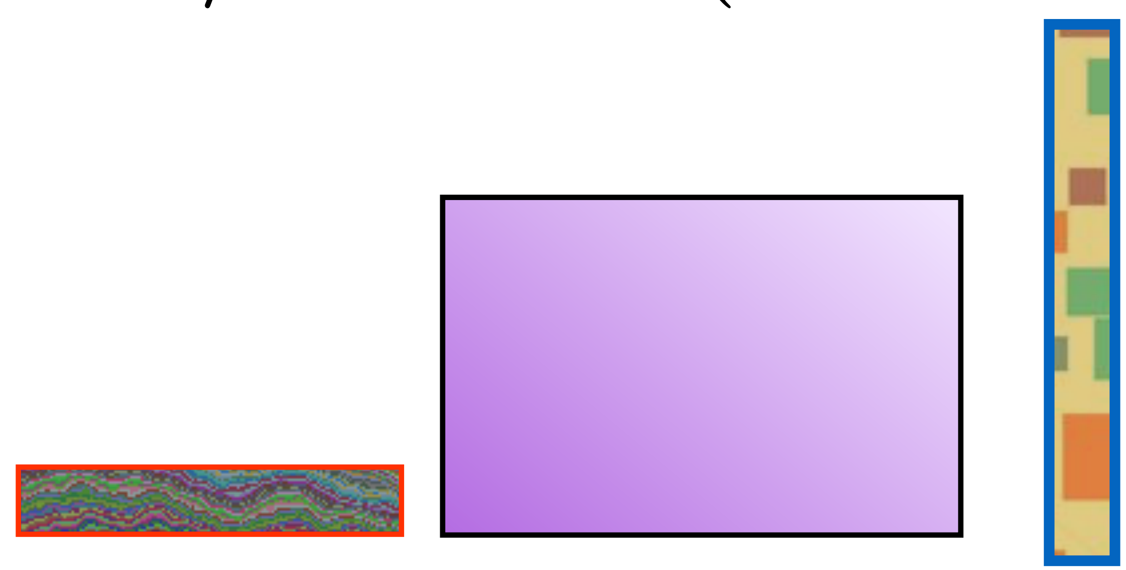
- **[Sinkhorn'64]** fixed-point iterations.



$$P_L := \mathbf{diag}(\boldsymbol{u_L})\, K\, \mathbf{diag}(\boldsymbol{v_L})$$

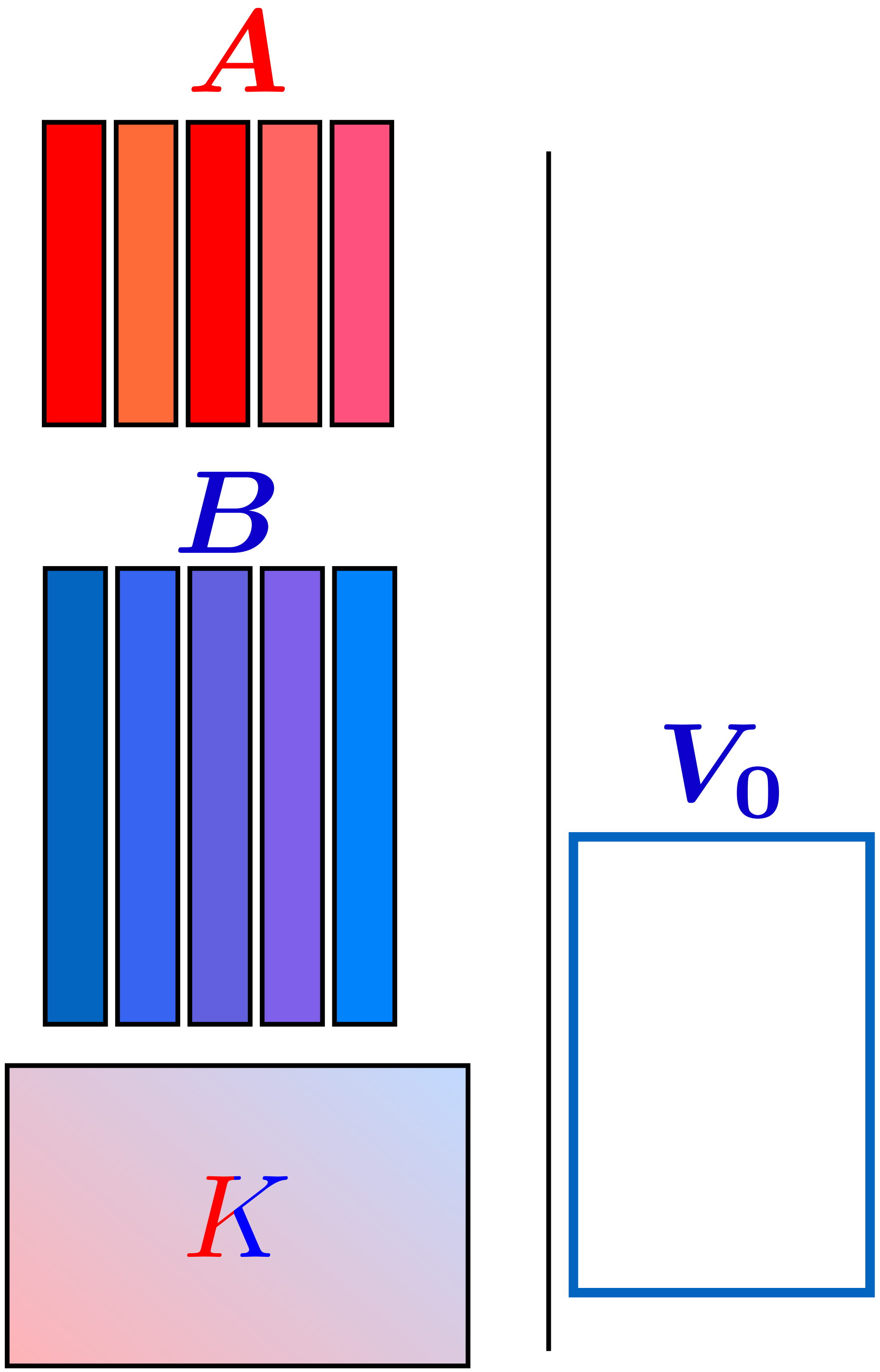


$$\langle P_L, M_{\boldsymbol{XY}} \rangle = \boldsymbol{u_L}^T (K \odot M_{\boldsymbol{XY}}) \boldsymbol{v_L}$$



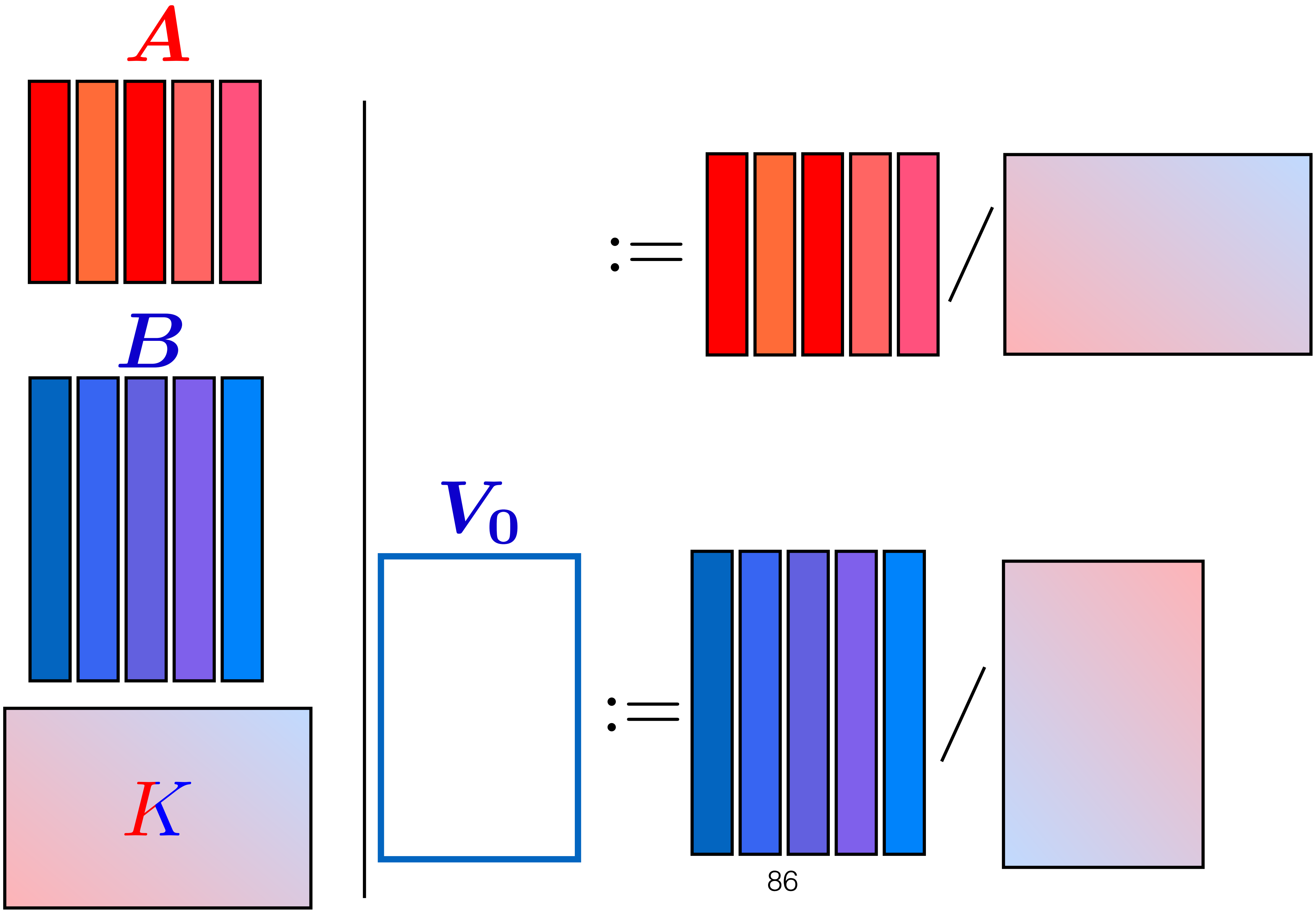
Also embarrassingly parallel

- **[Sinkhorn'64]** with *matrix* fixed-point iterations



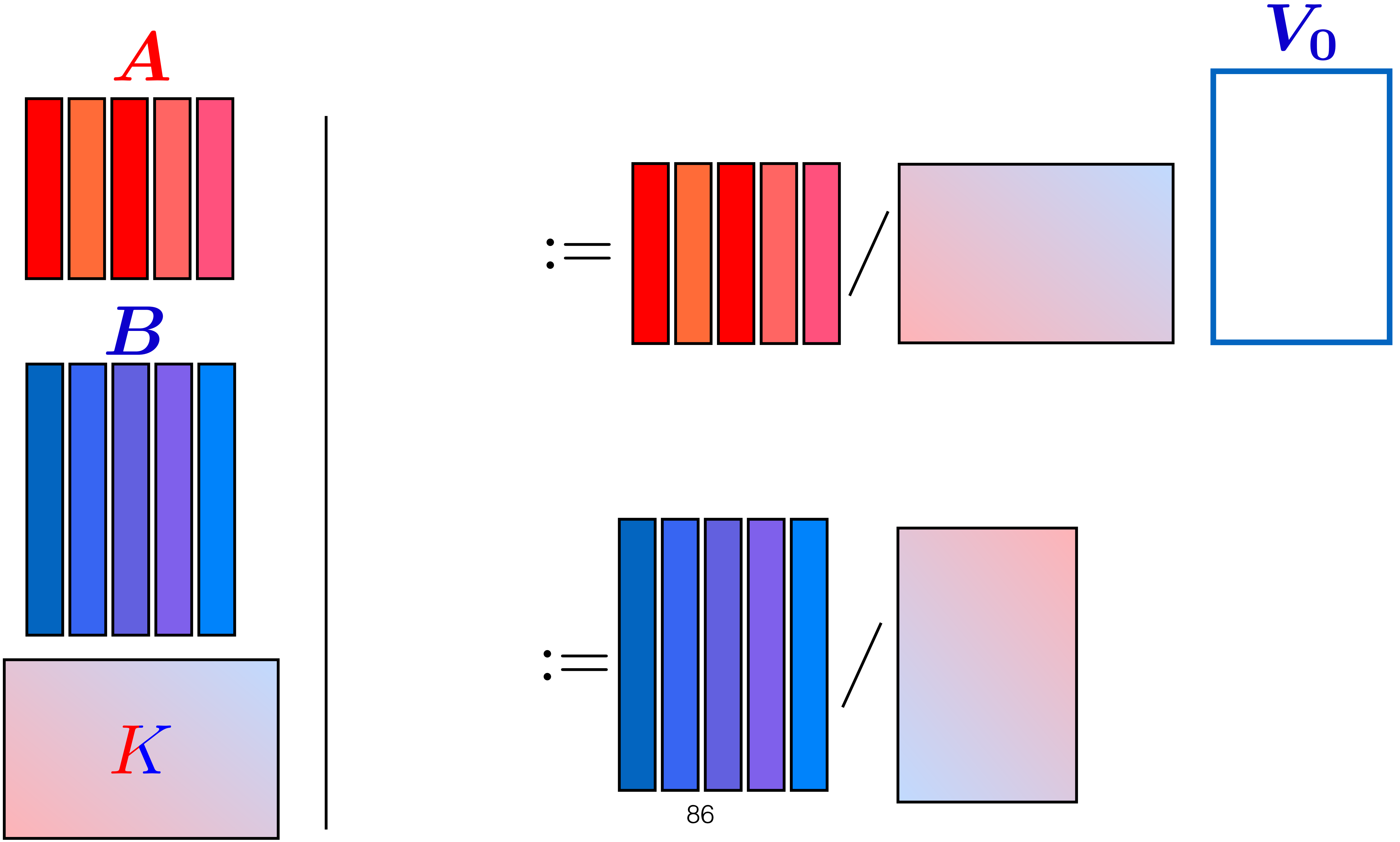
Also embarrassingly parallel

- **[Sinkhorn'64]** with *matrix* fixed-point iterations



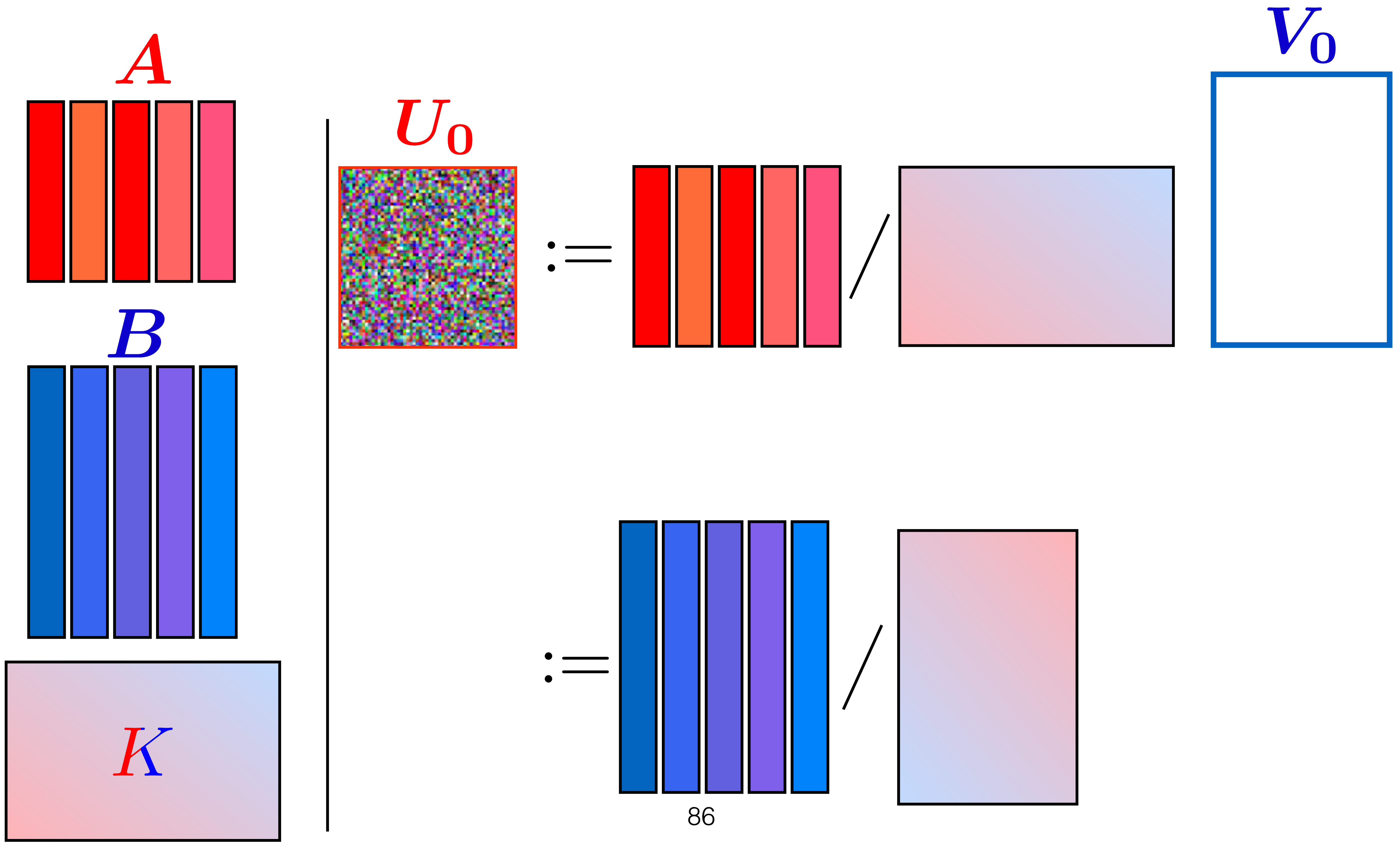
Also embarrassingly parallel

- **[Sinkhorn'64]** with *matrix* fixed-point iterations



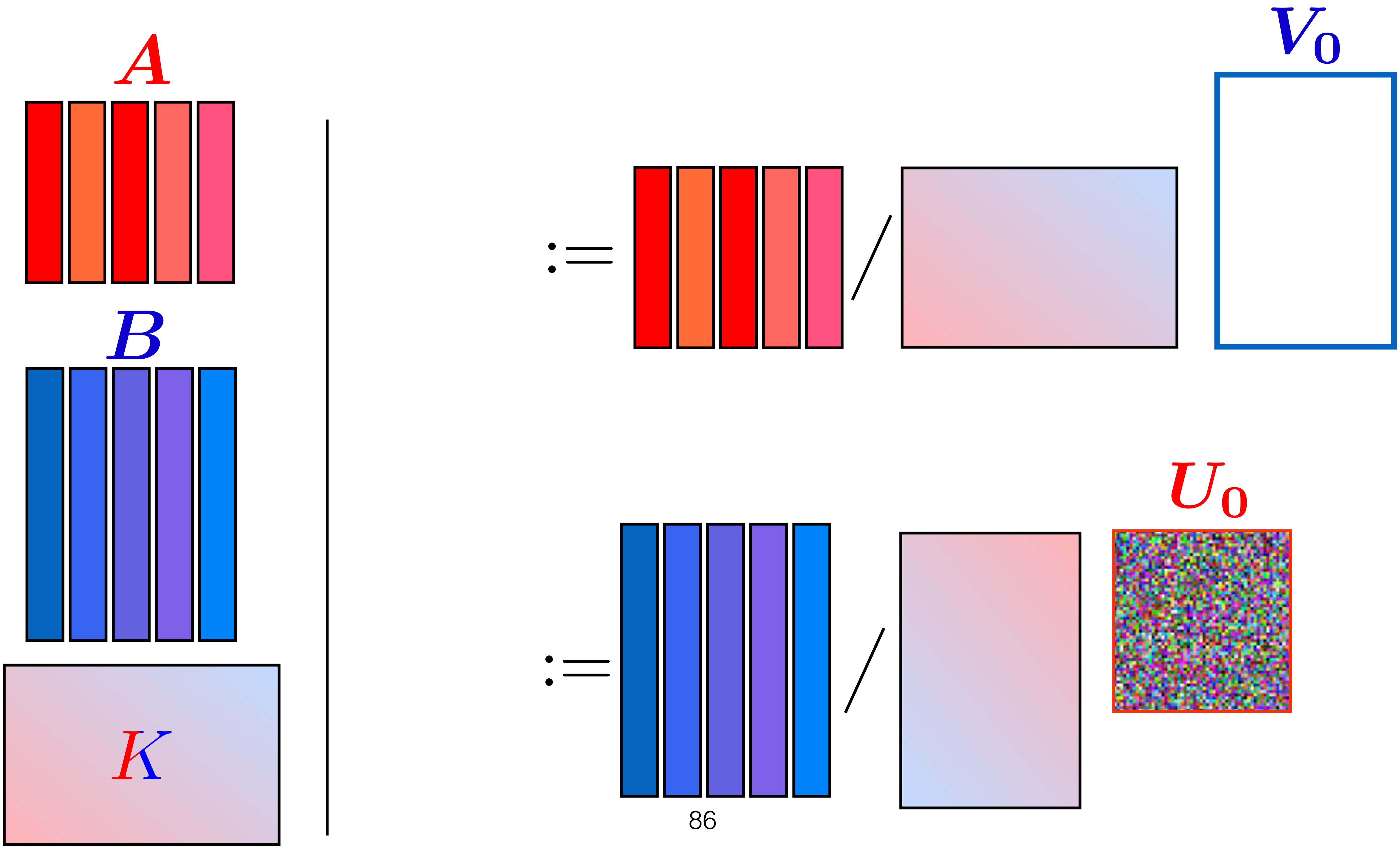
Also embarrassingly parallel

- **[Sinkhorn'64]** with *matrix* fixed-point iterations



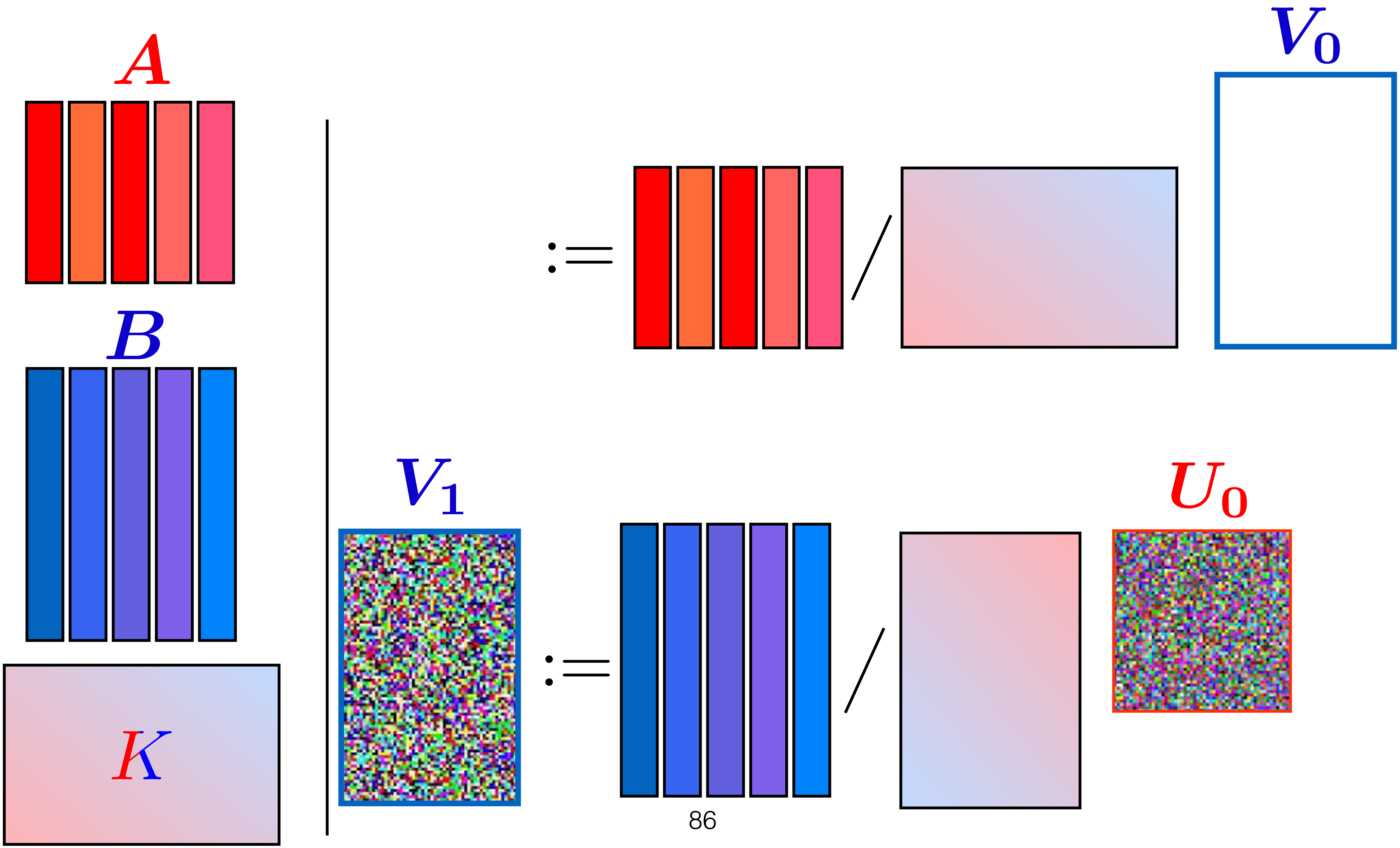
Also embarrassingly parallel

- **[Sinkhorn'64]** with *matrix* fixed-point iterations



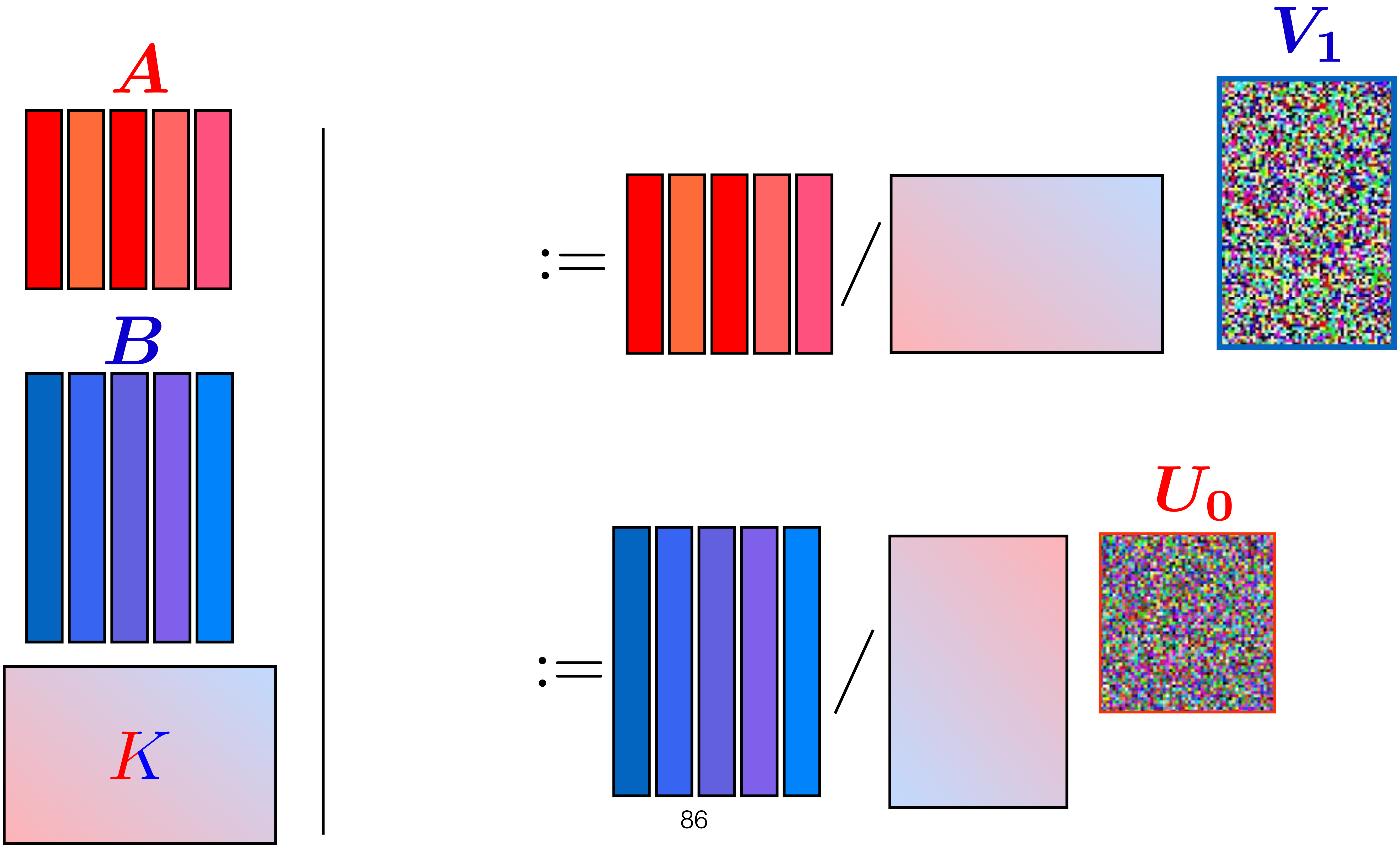
Also embarrassingly parallel

- **[Sinkhorn'64]** with *matrix* fixed-point iterations



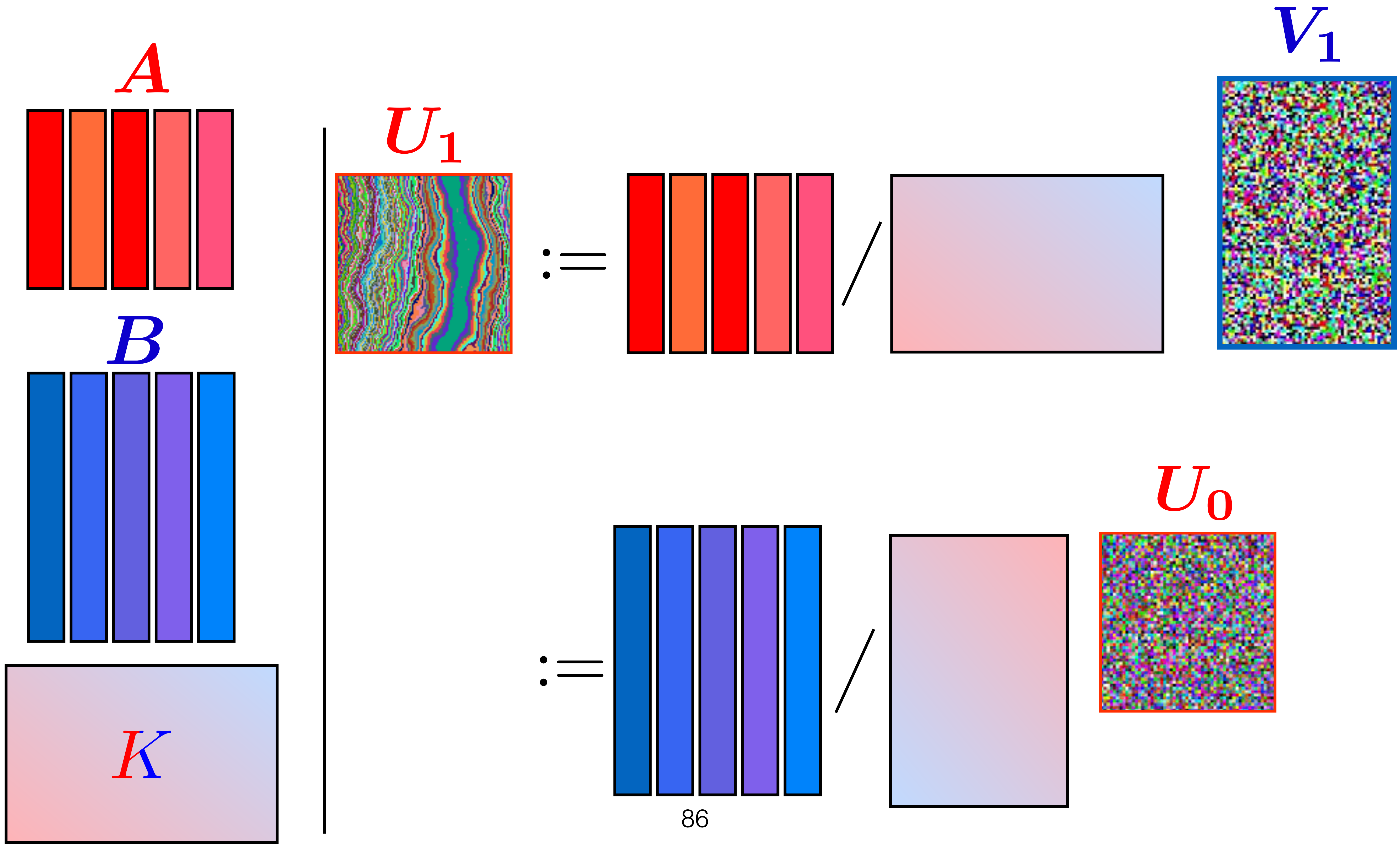
Also embarrassingly parallel

- **[Sinkhorn'64]** with *matrix* fixed-point iterations



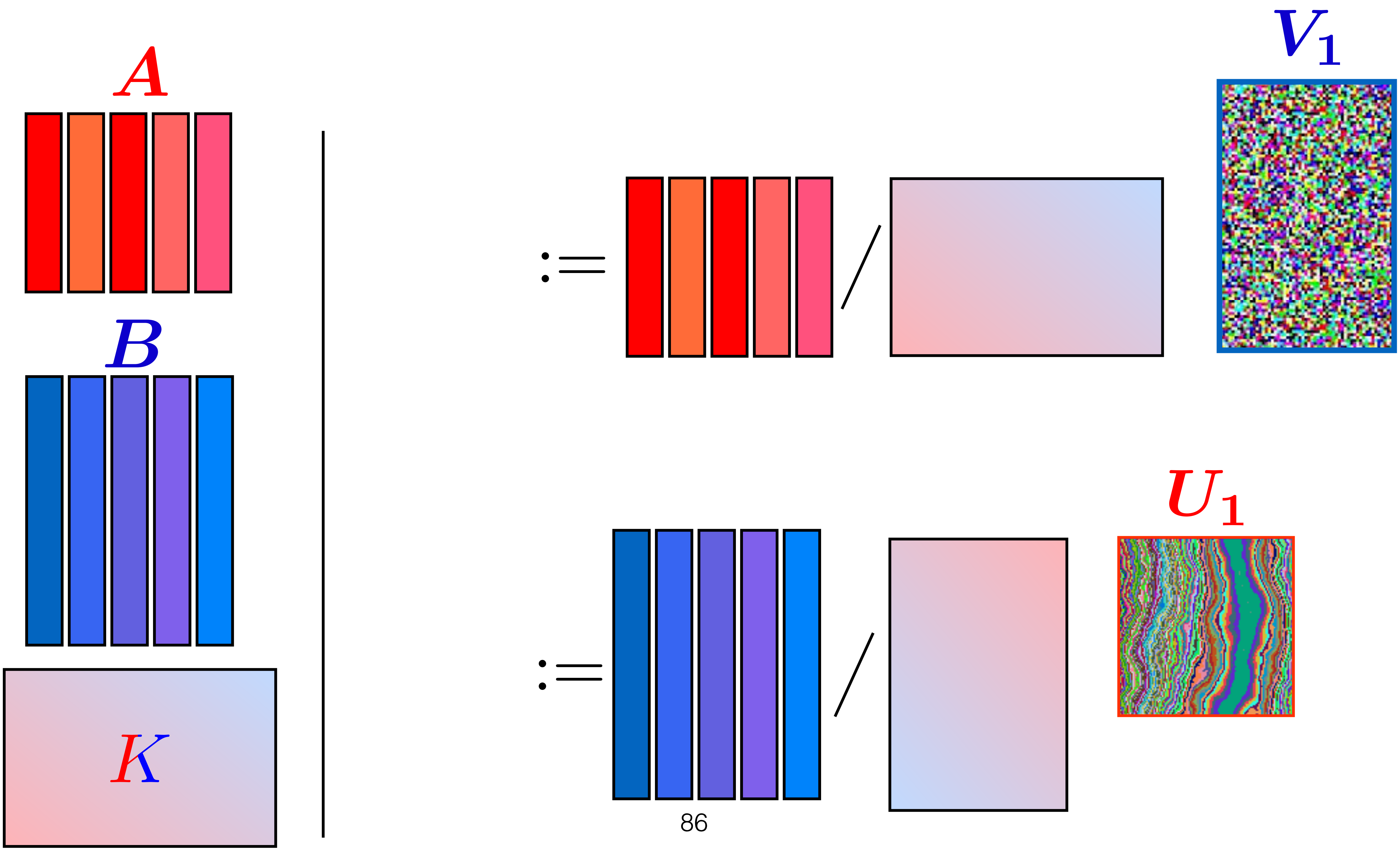
Also embarrassingly parallel

- **[Sinkhorn'64]** with *matrix* fixed-point iterations



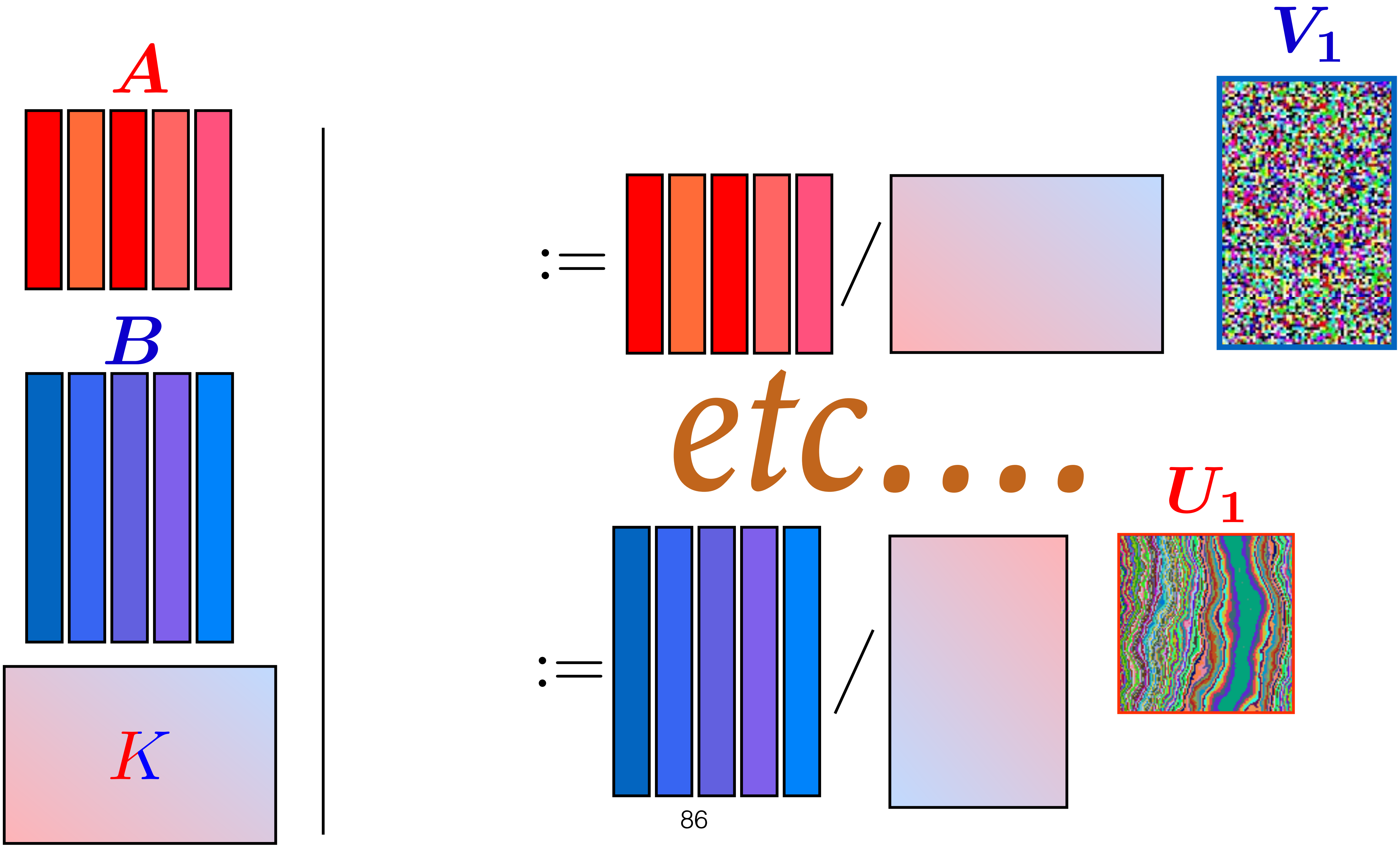
Also embarrassingly parallel

- **[Sinkhorn'64]** with *matrix* fixed-point iterations

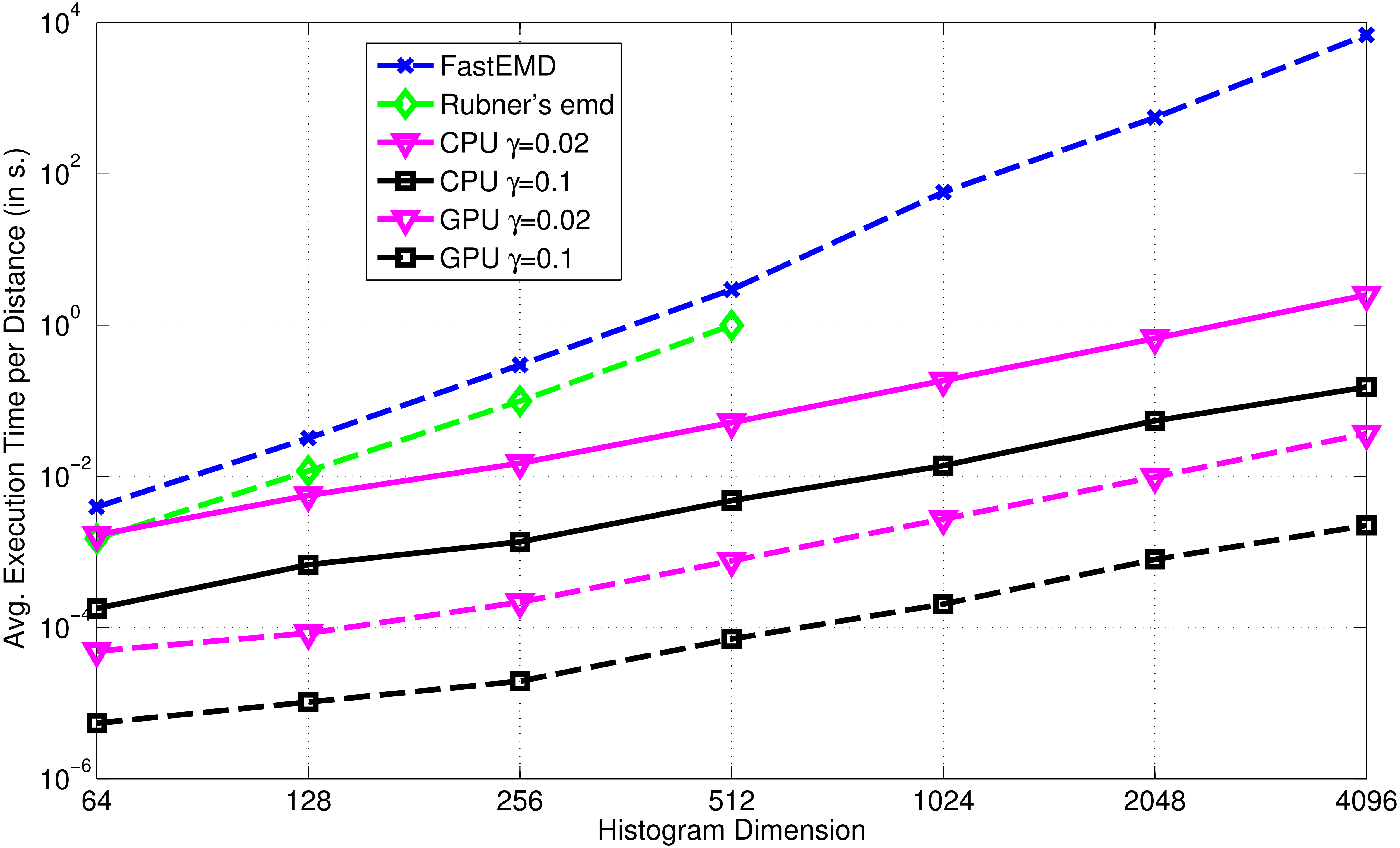


Also embarrassingly parallel

- **[Sinkhorn'64]** with *matrix* fixed-point iterations



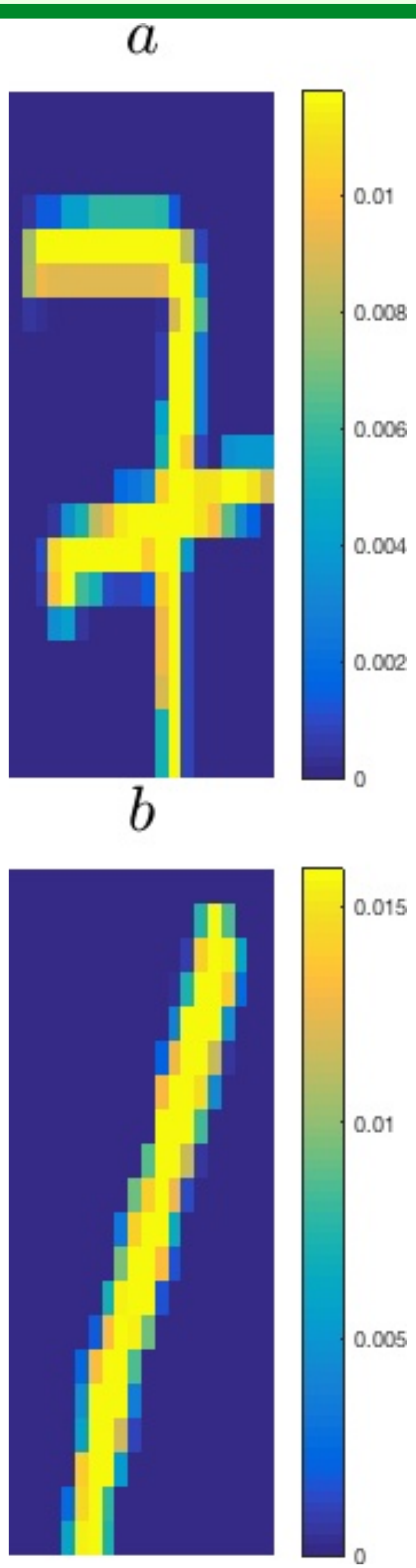
Very Fast EMD Approx. Solver



Note. $(\Omega, \boldsymbol{D})$ is a random graph with shortest path metric, histograms sampled uniformly on simplex, Sinkhorn tolerance 10^{-2}.

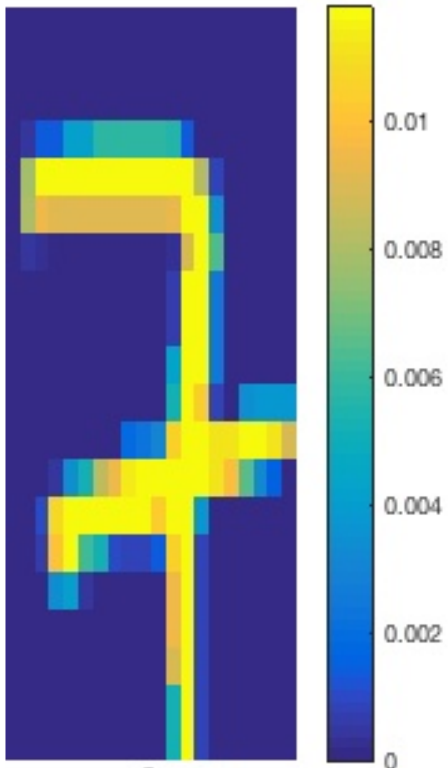
Very Fast EMD Approx. Solver

Very Fast EMD Approx. Solver

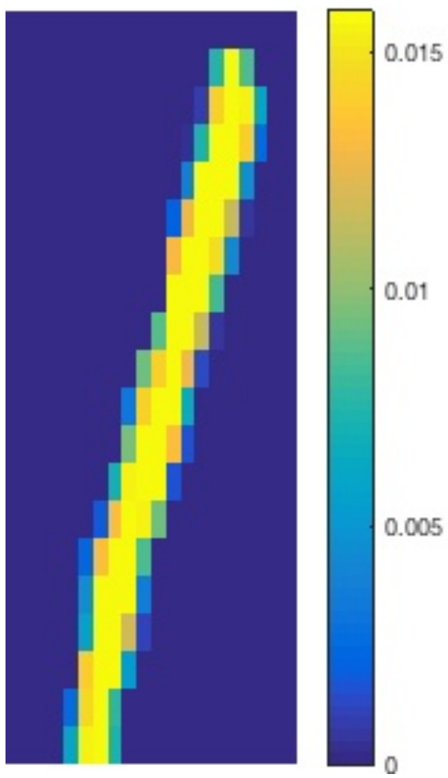


Very Fast EMD Approx. Solver

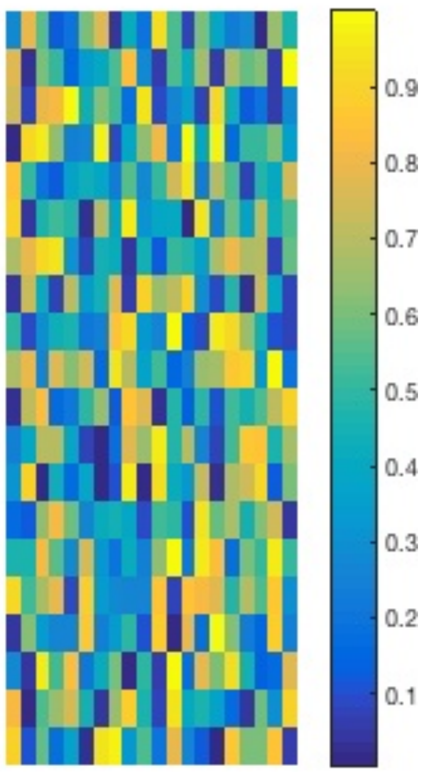
a



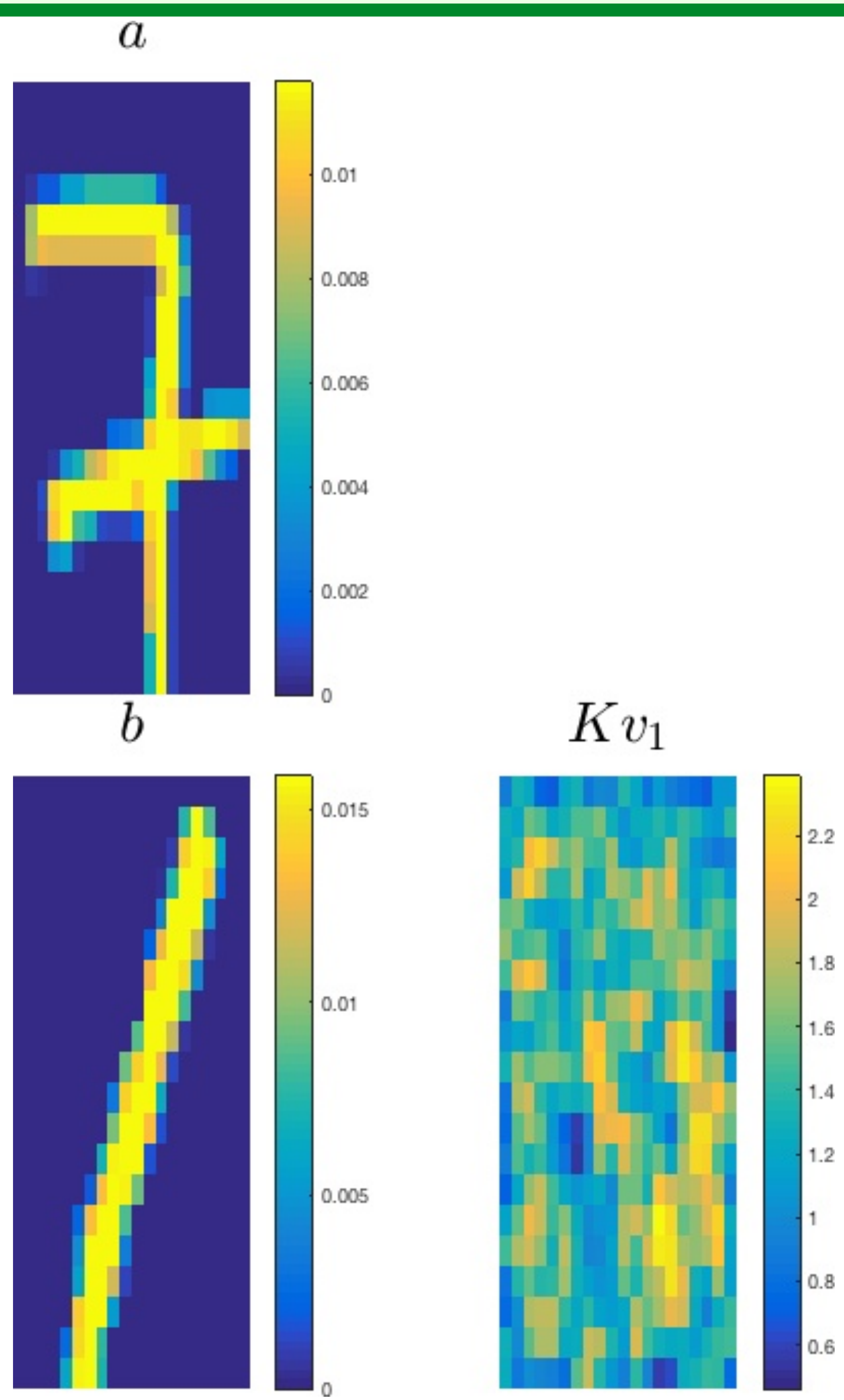
b



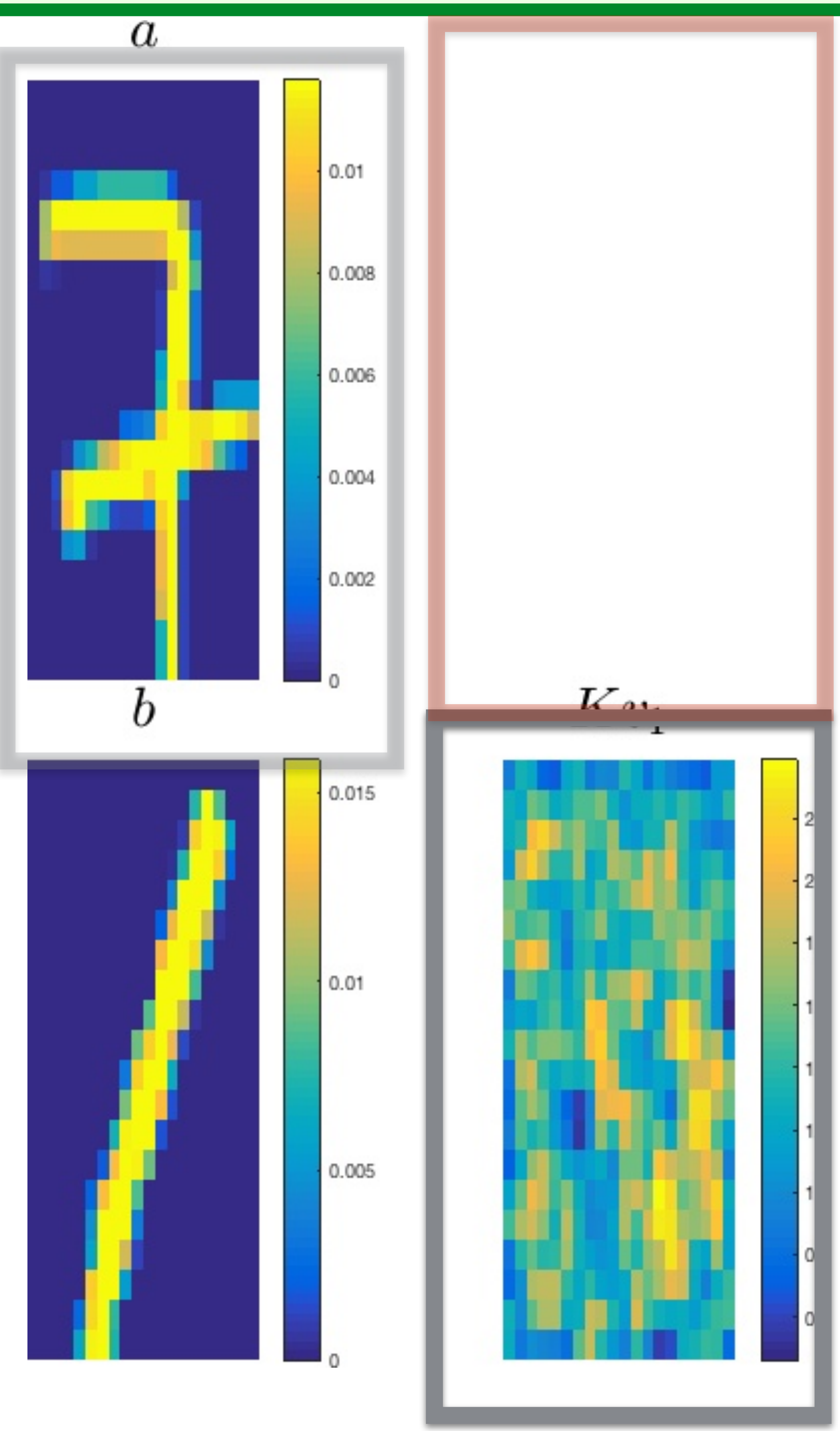
$v_1 \leftarrow \text{noise}$



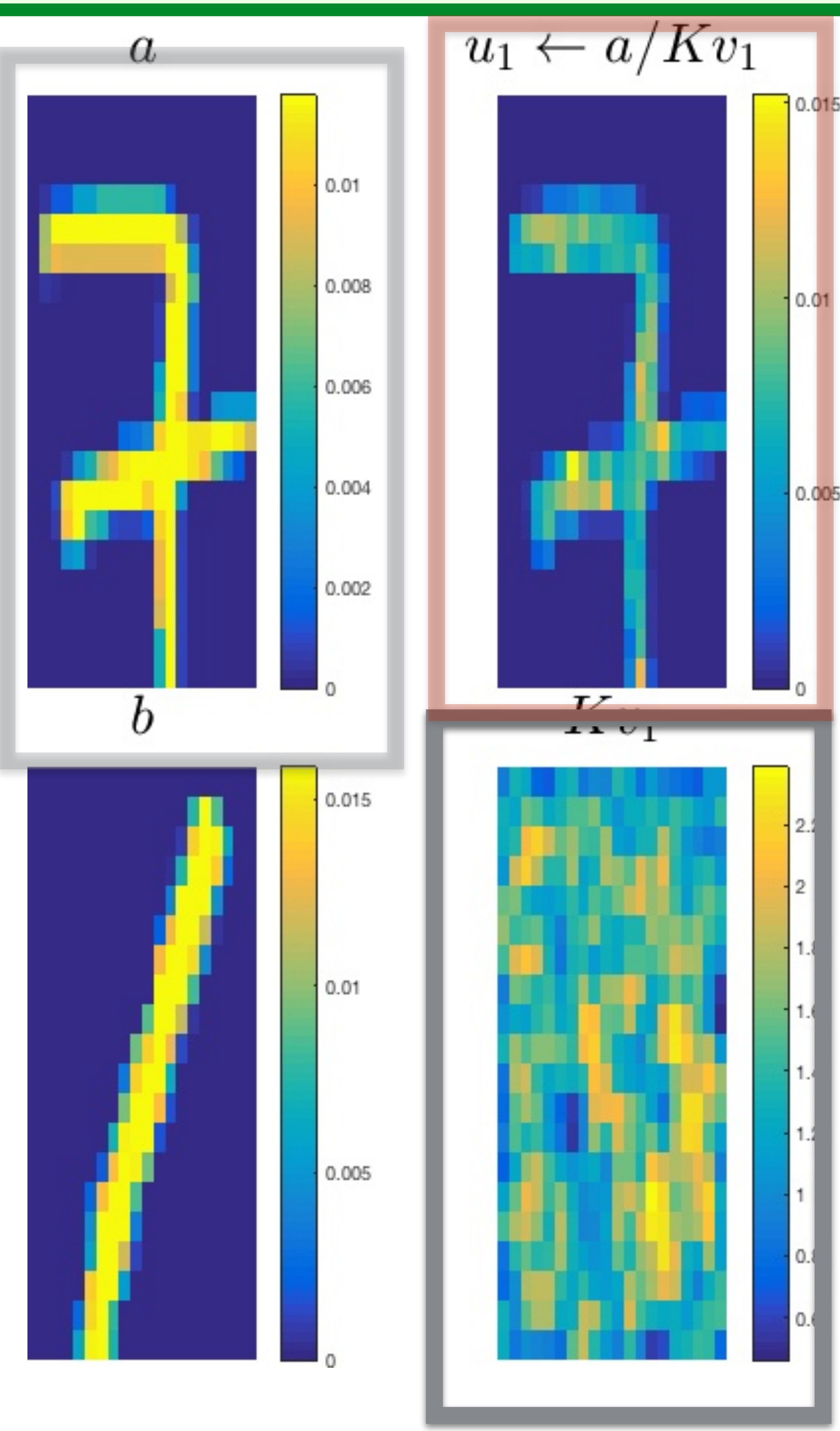
Very Fast EMD Approx. Solver



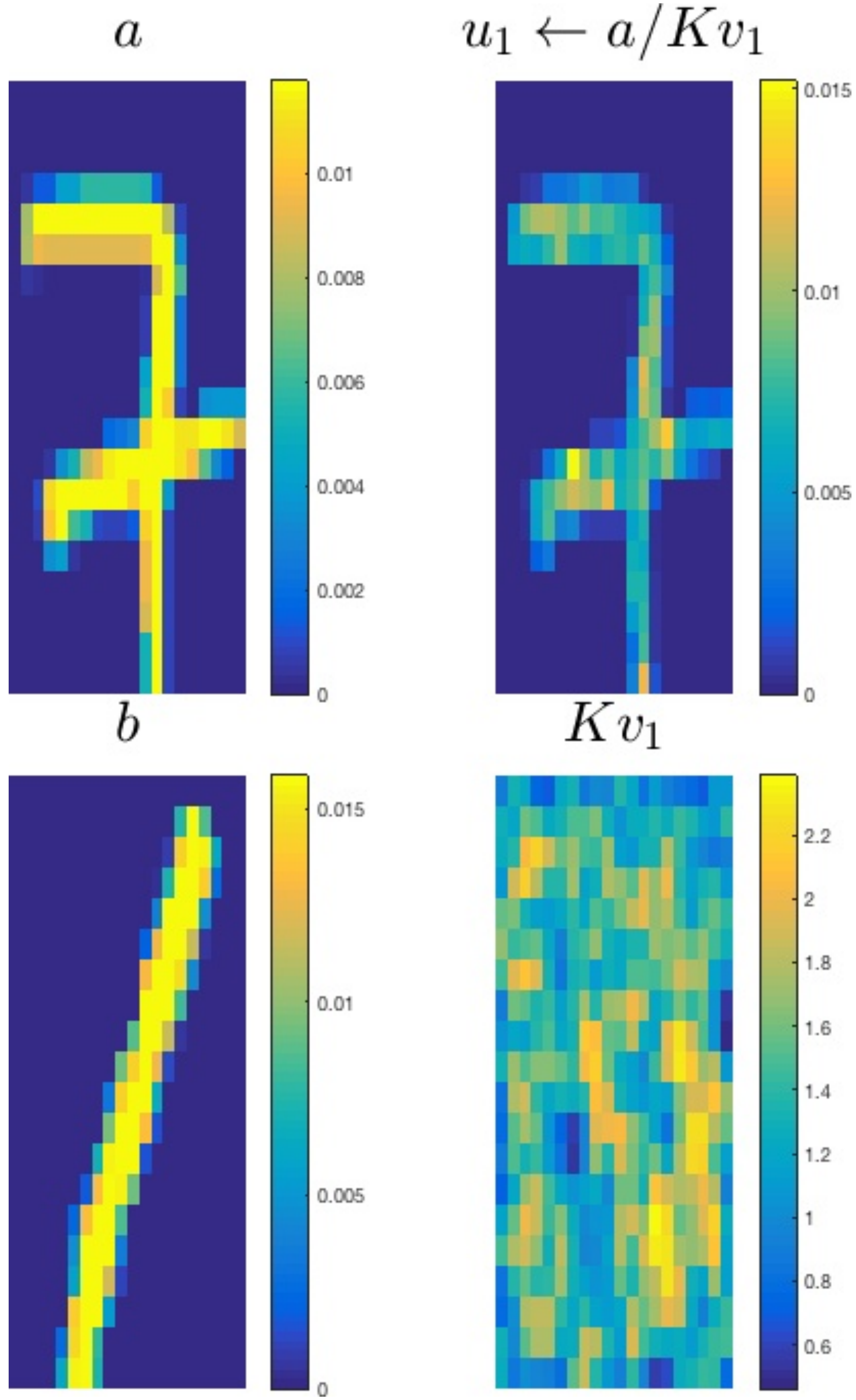
Very Fast EMD Approx. Solver



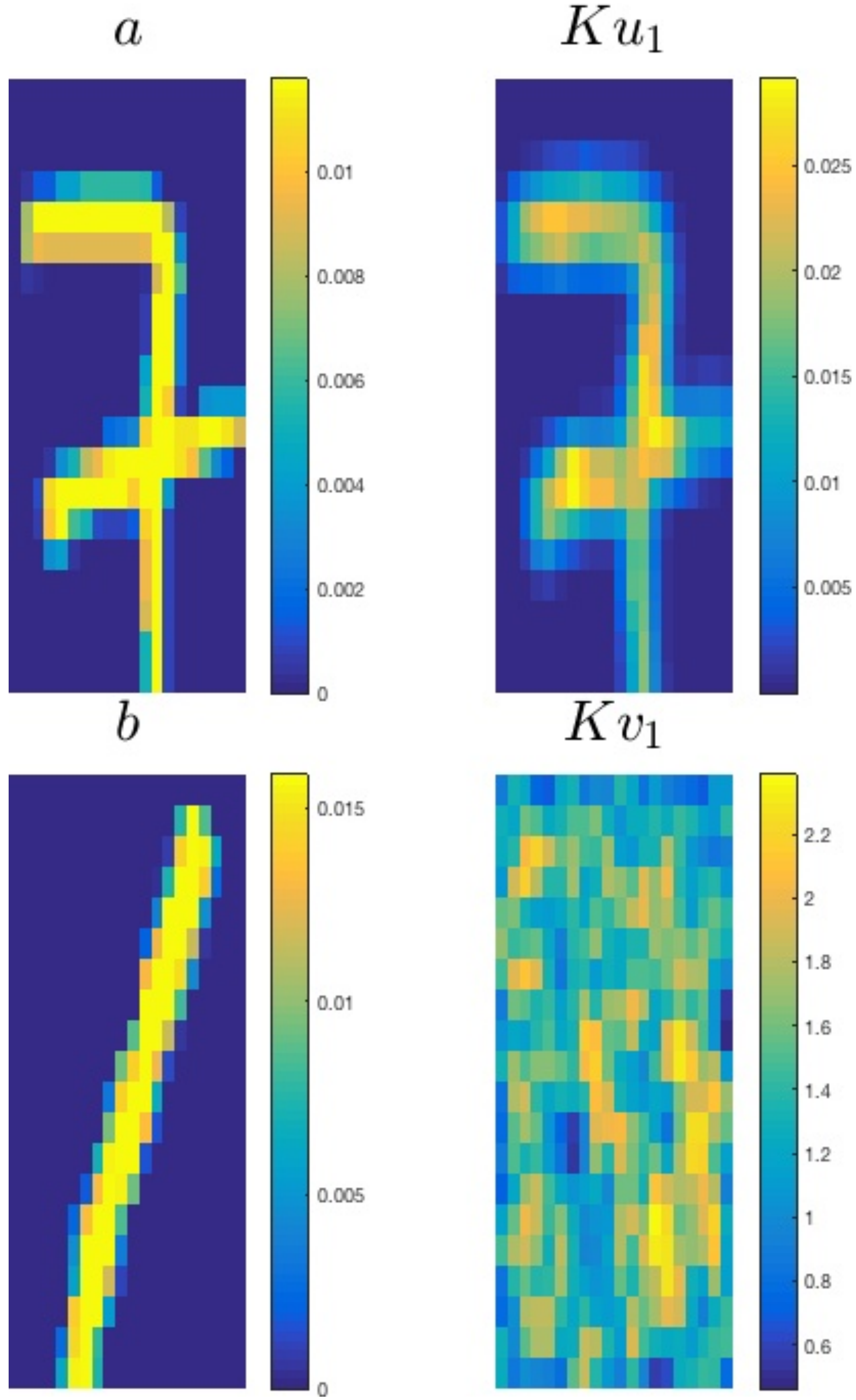
Very Fast EMD Approx. Solver



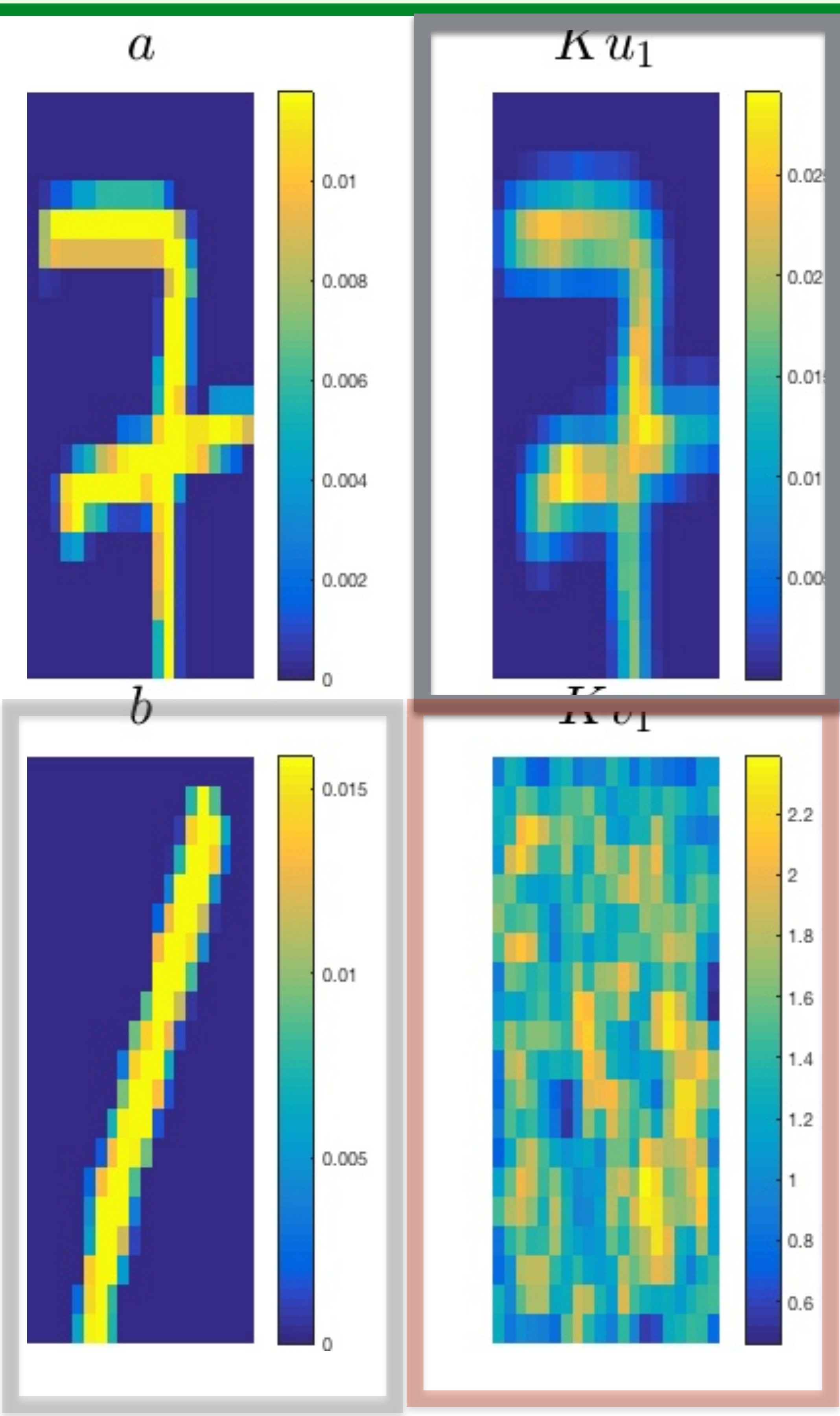
Very Fast EMD Approx. Solver



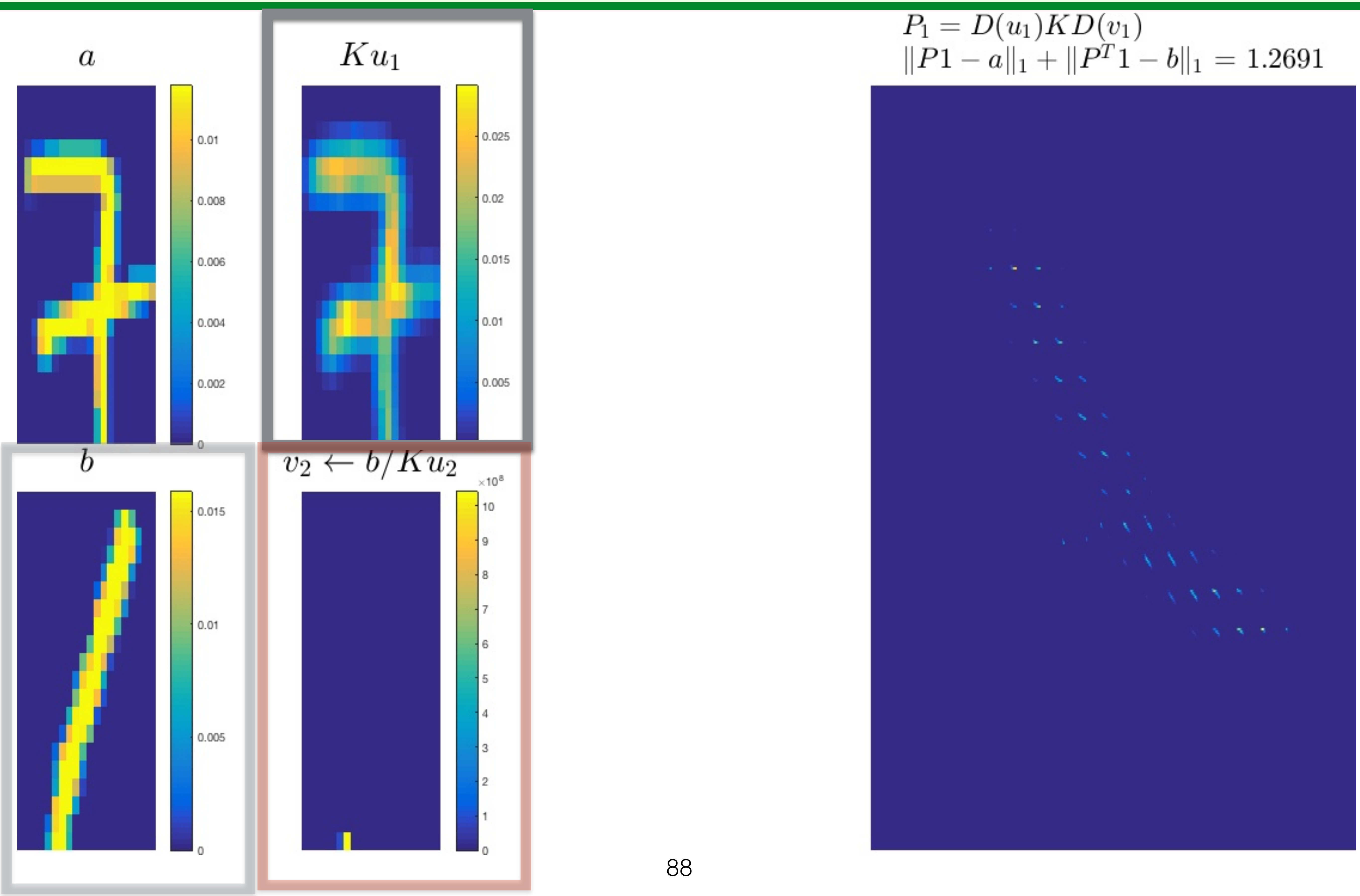
Very Fast EMD Approx. Solver



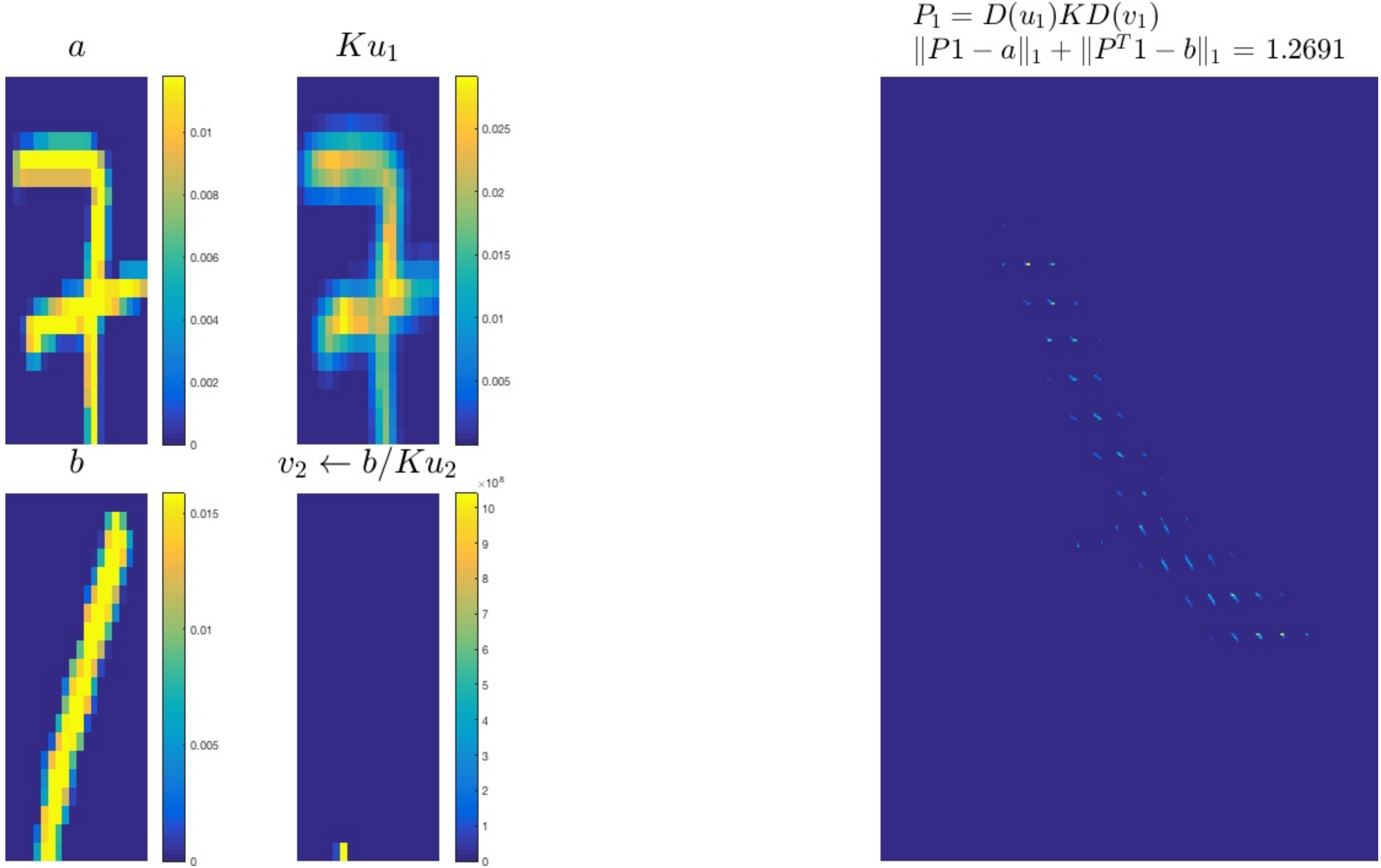
Very Fast EMD Approx. Solver



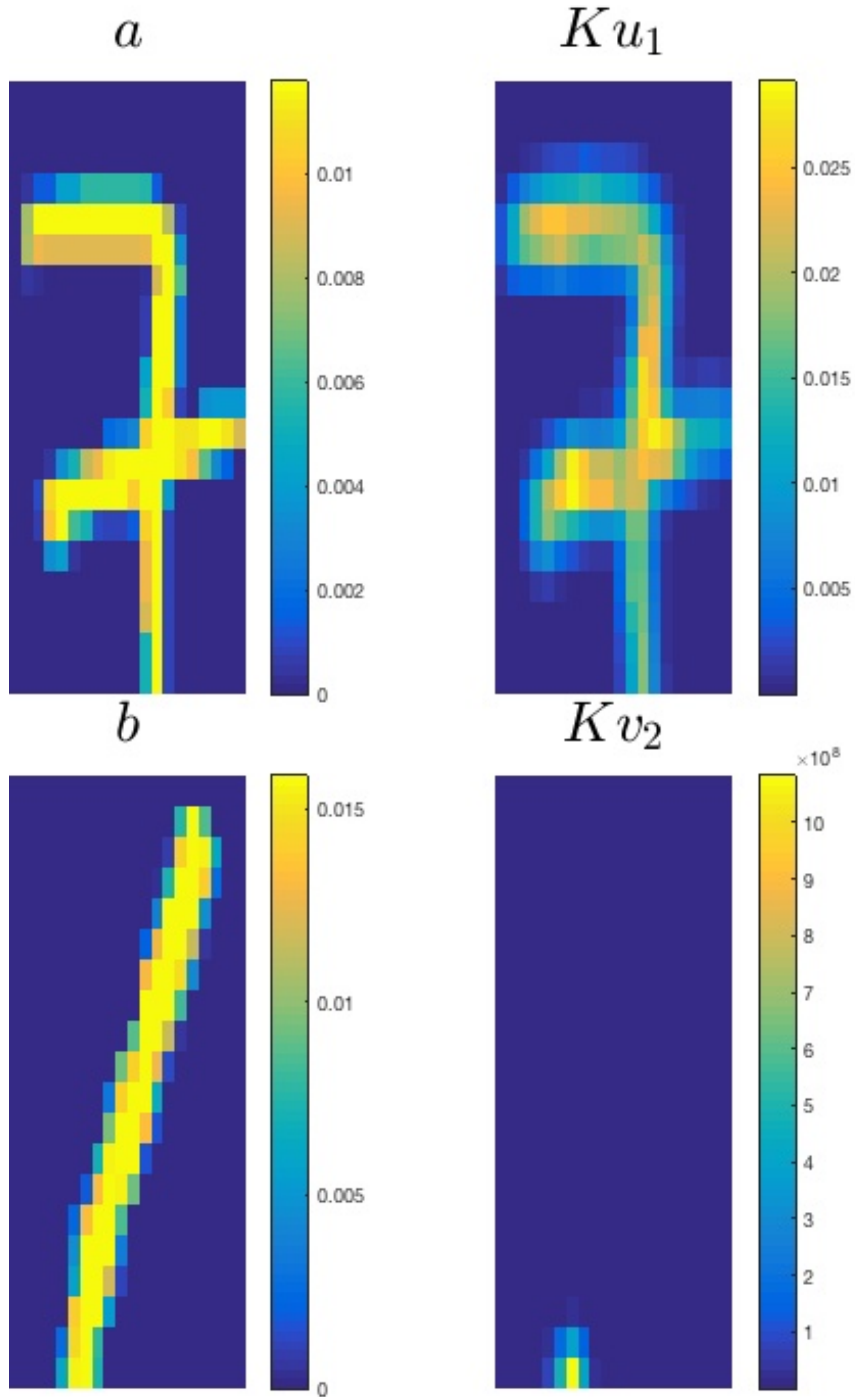
Very Fast EMD Approx. Solver

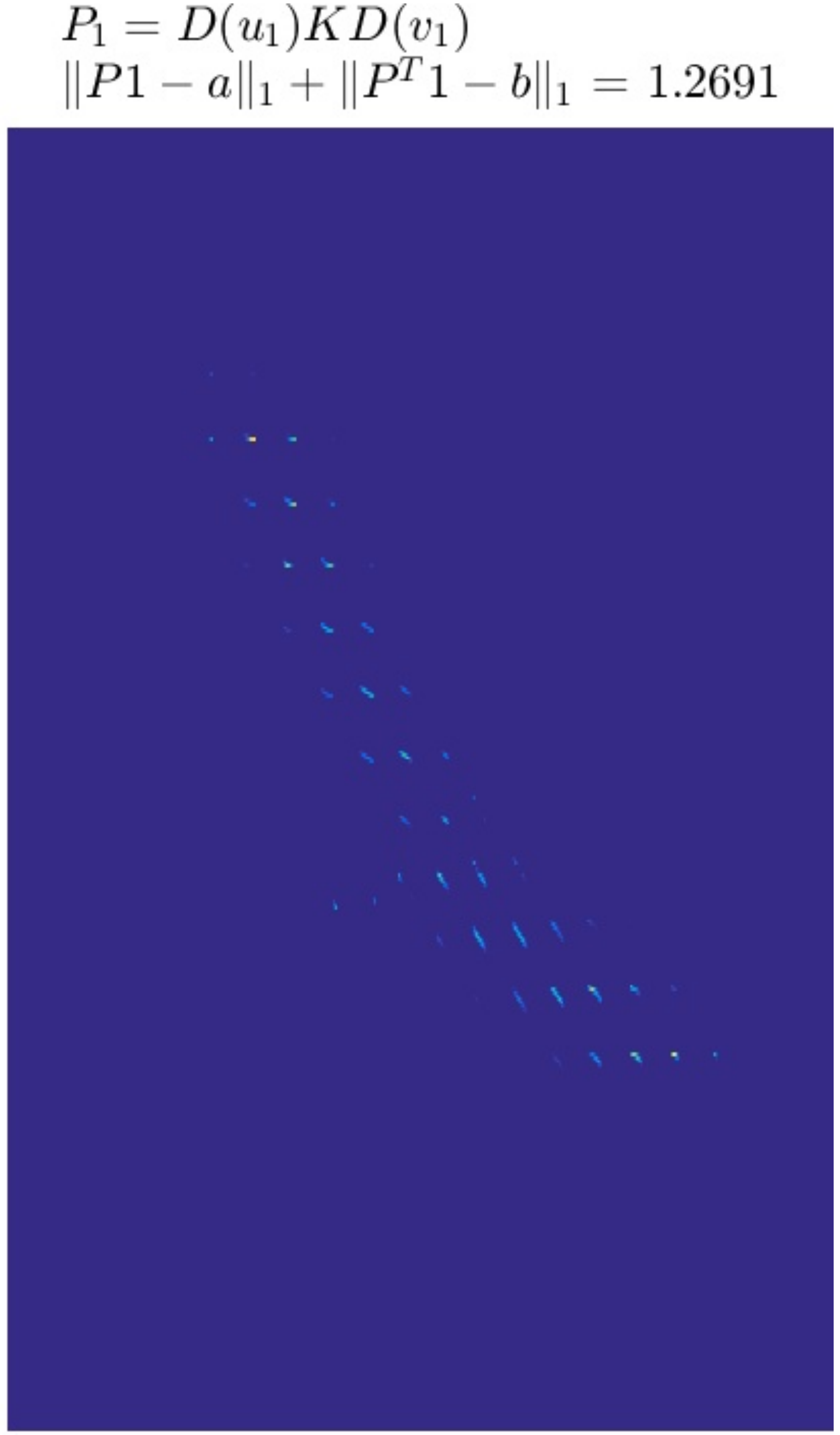


Very Fast EMD Approx. Solver

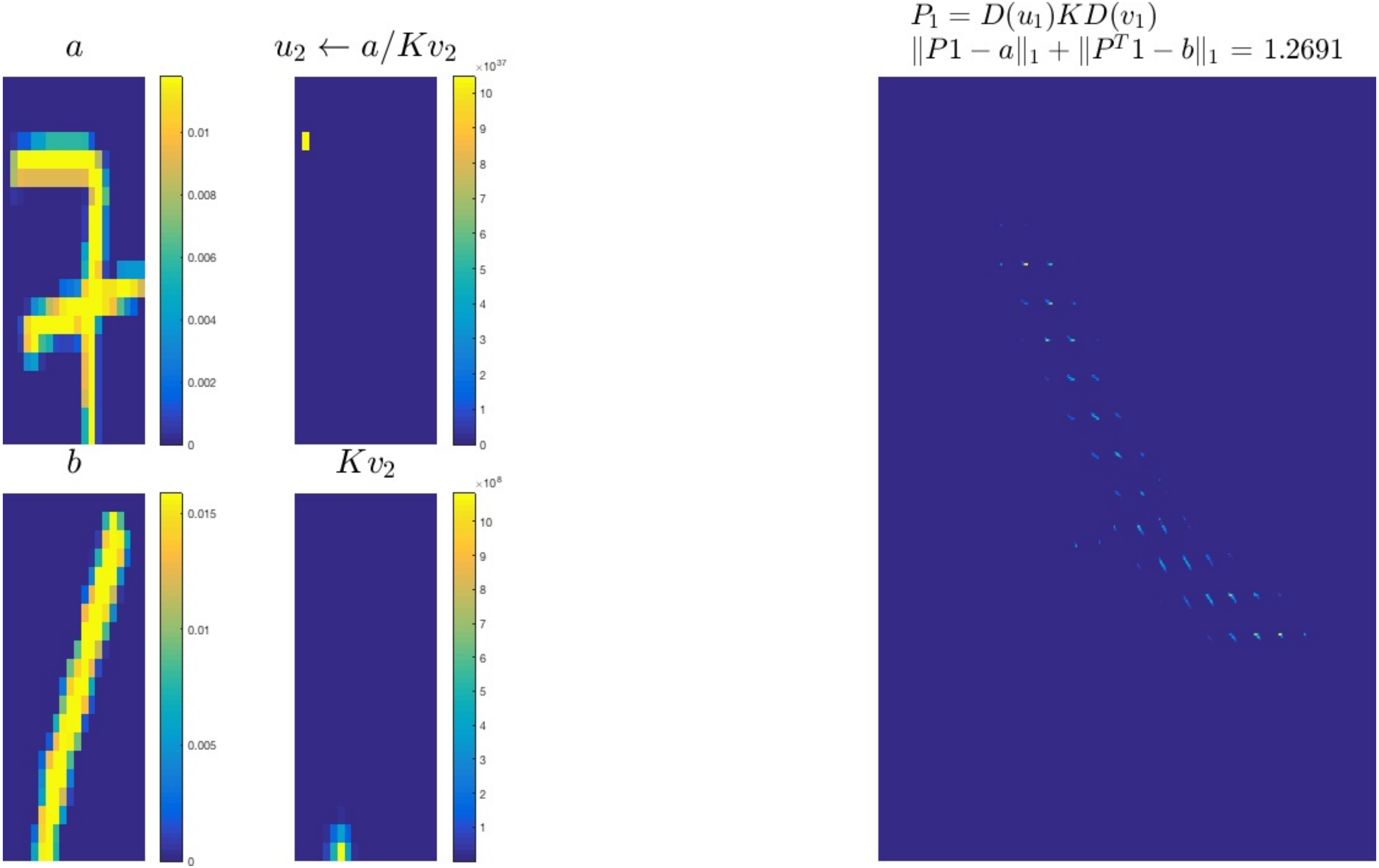


Very Fast EMD Approx. Solver

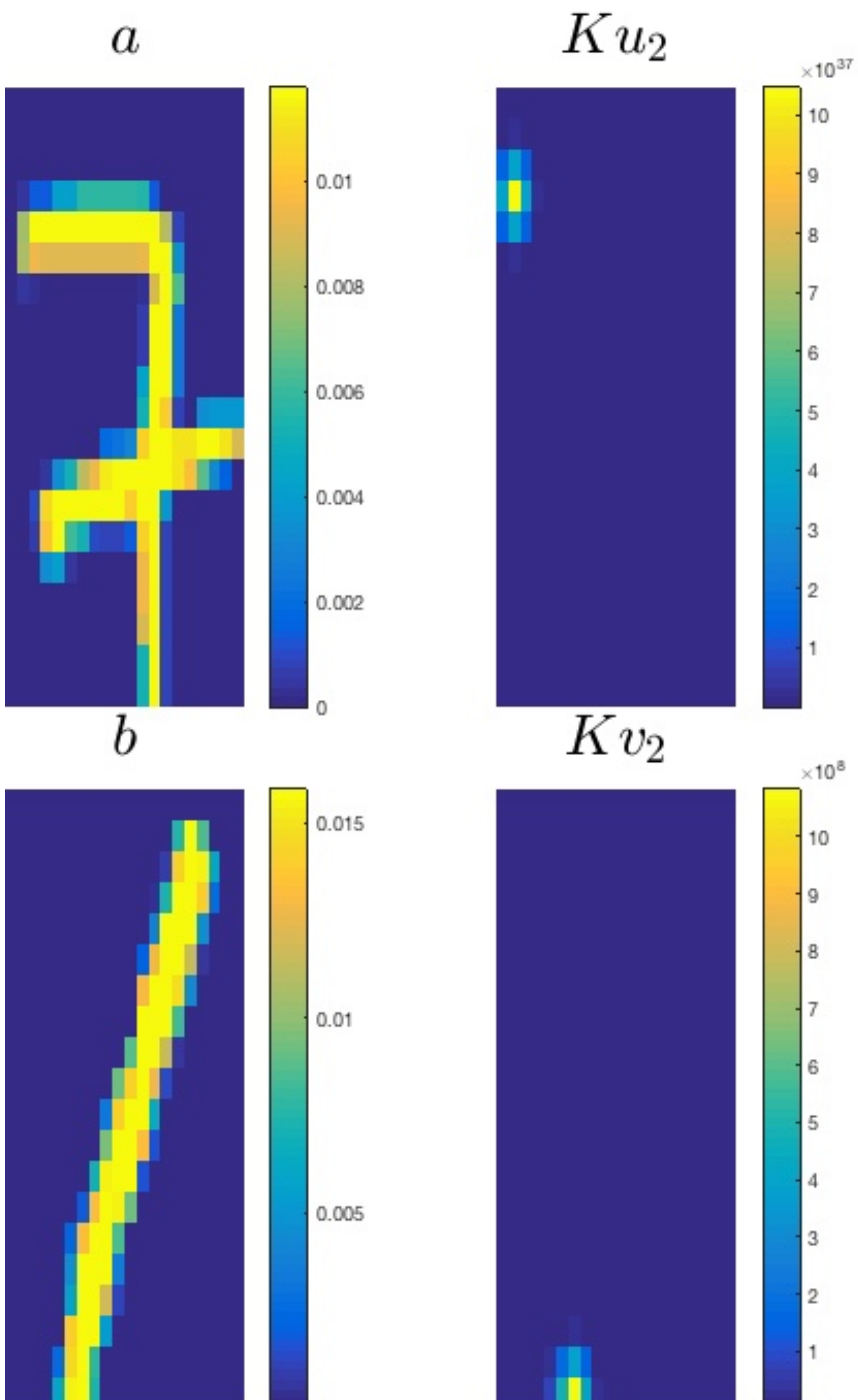




Very Fast EMD Approx. Solver



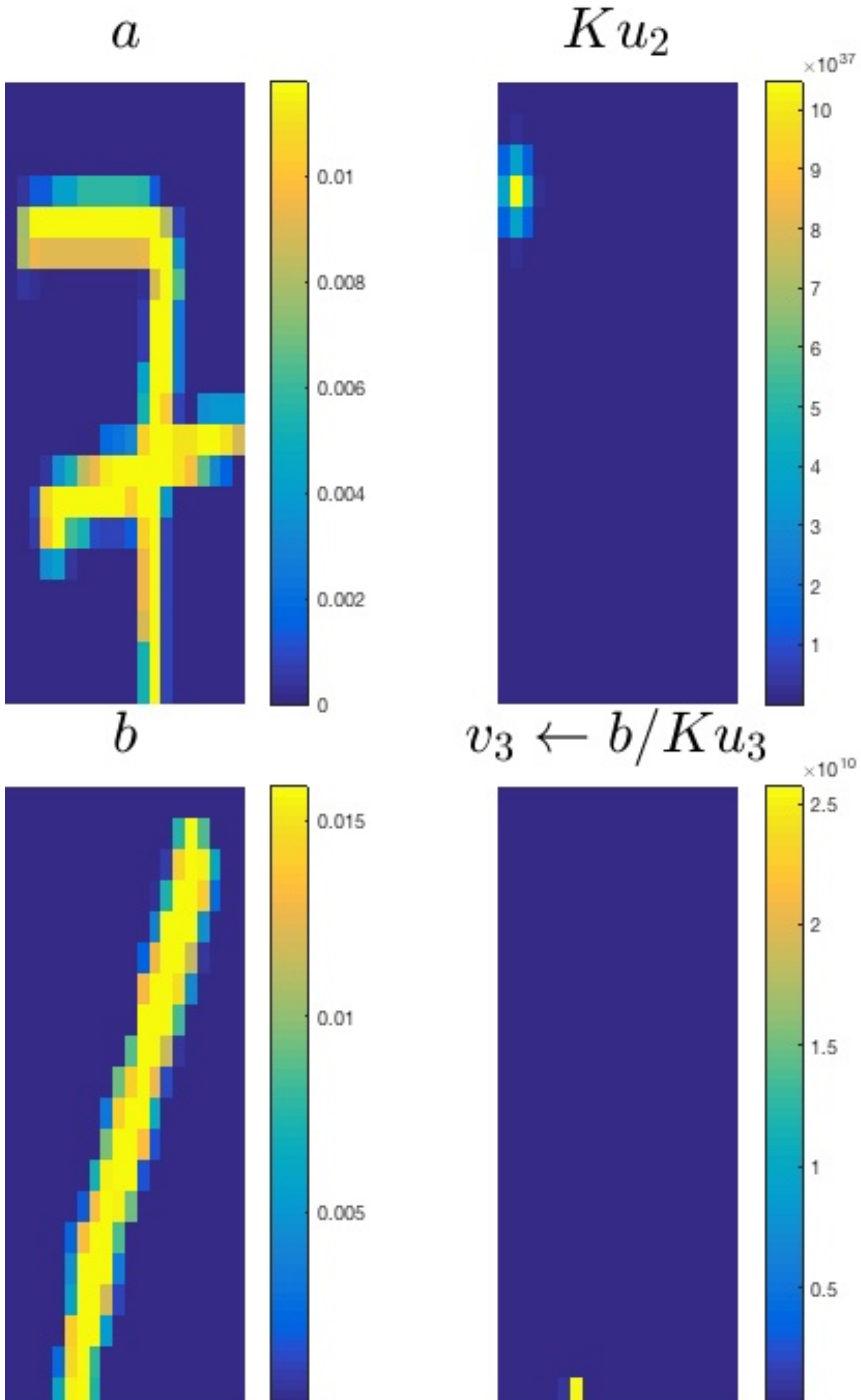
Very Fast EMD Approx. Solver



$P_1 = D(u_1)KD(v_1)$

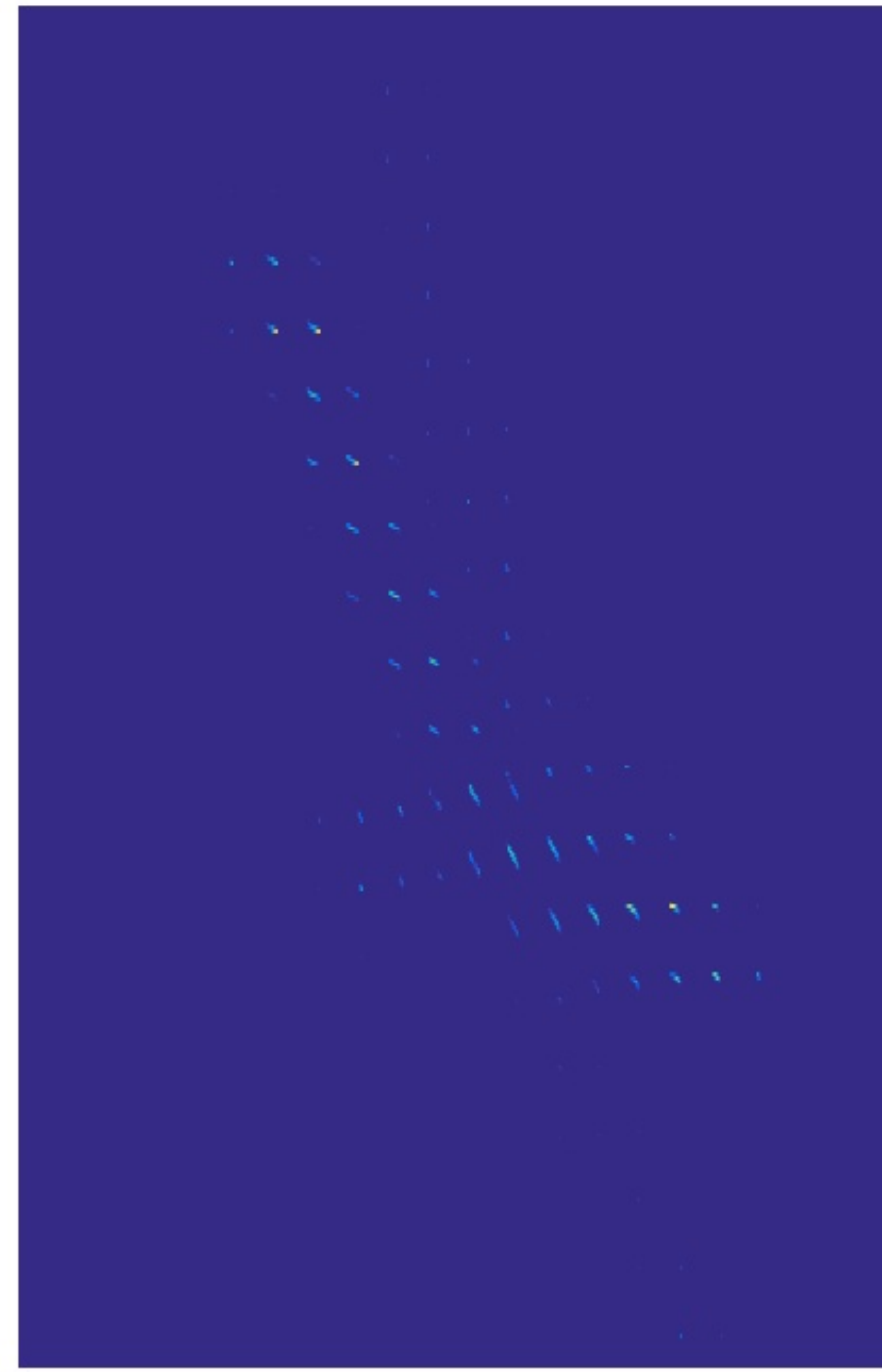
$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 1.2691$

Very Fast EMD Approx. Solver

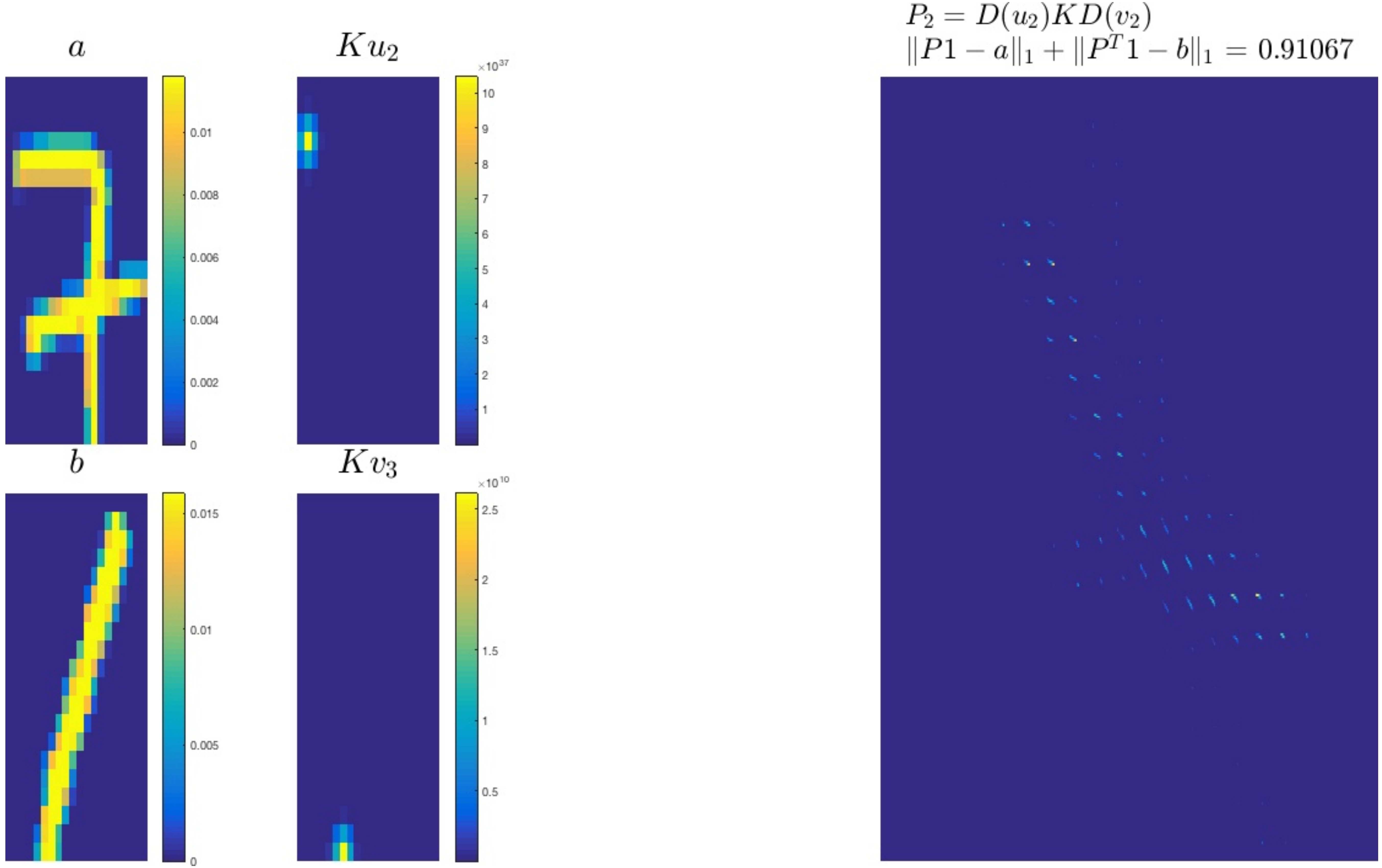


$P_2 = D(u_2)KD(v_2)$

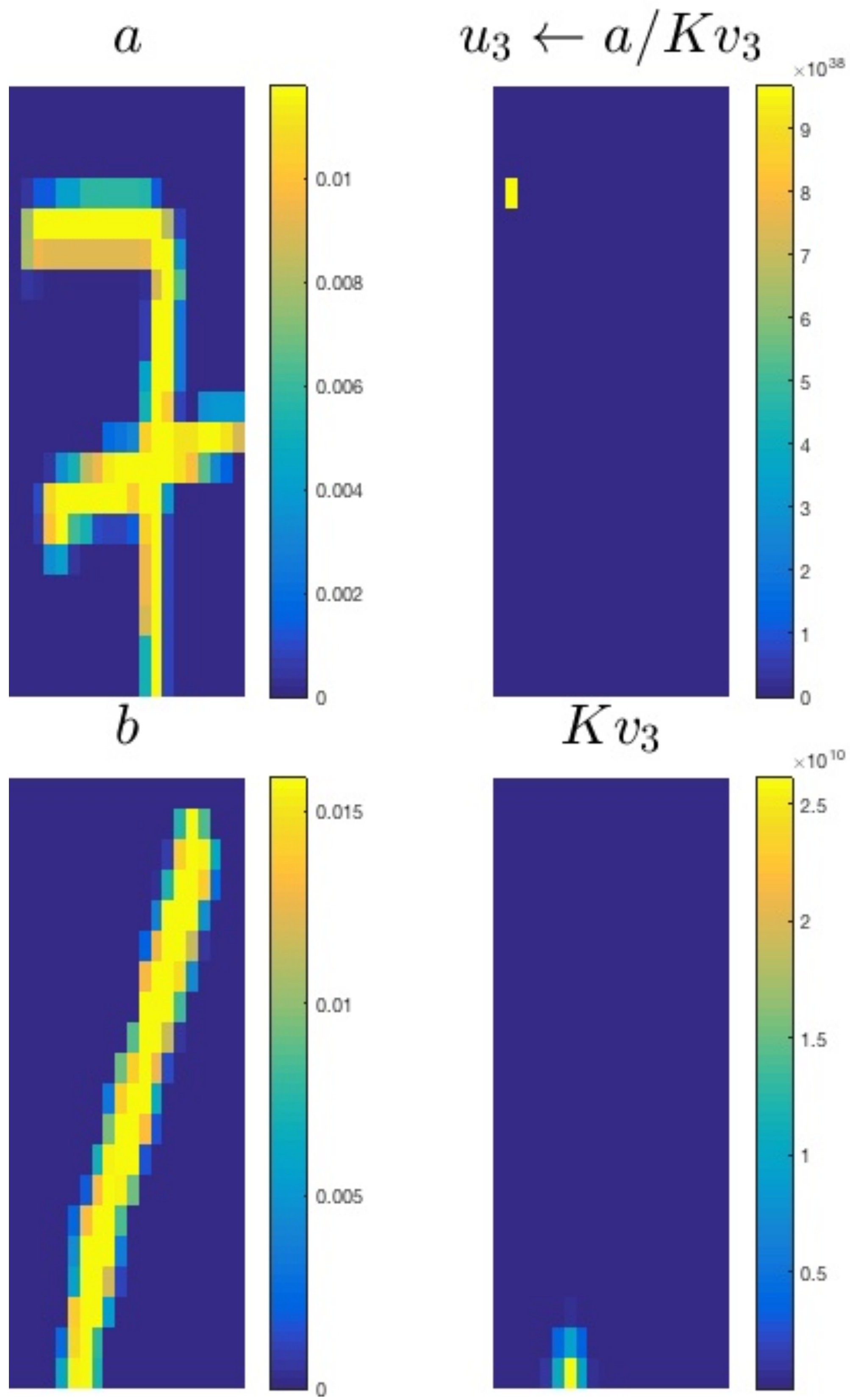
$\|P\mathbb{1} - a\|_1 + \|P^T\mathbb{1} - b\|_1 = 0.91067$

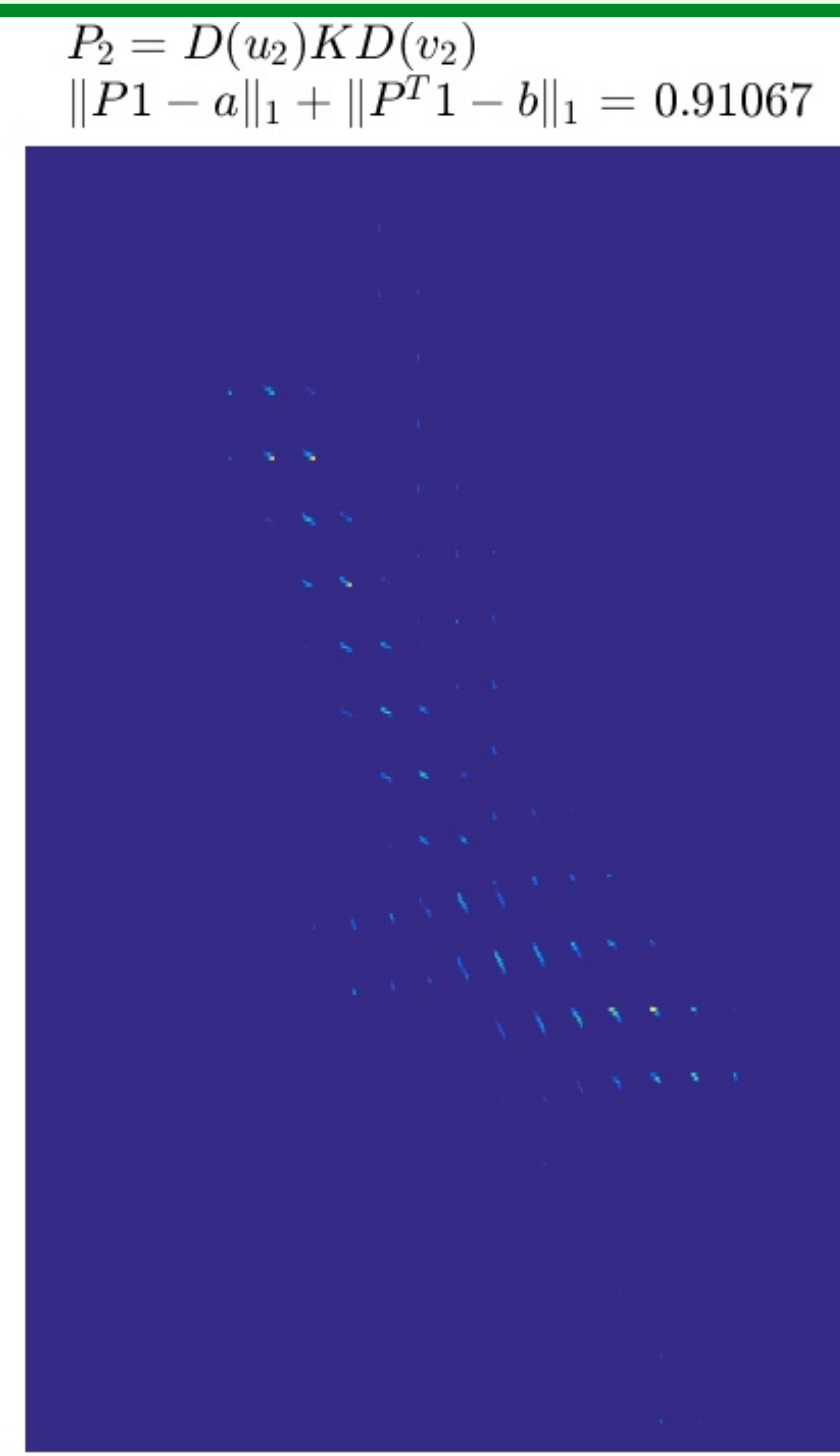


Very Fast EMD Approx. Solver

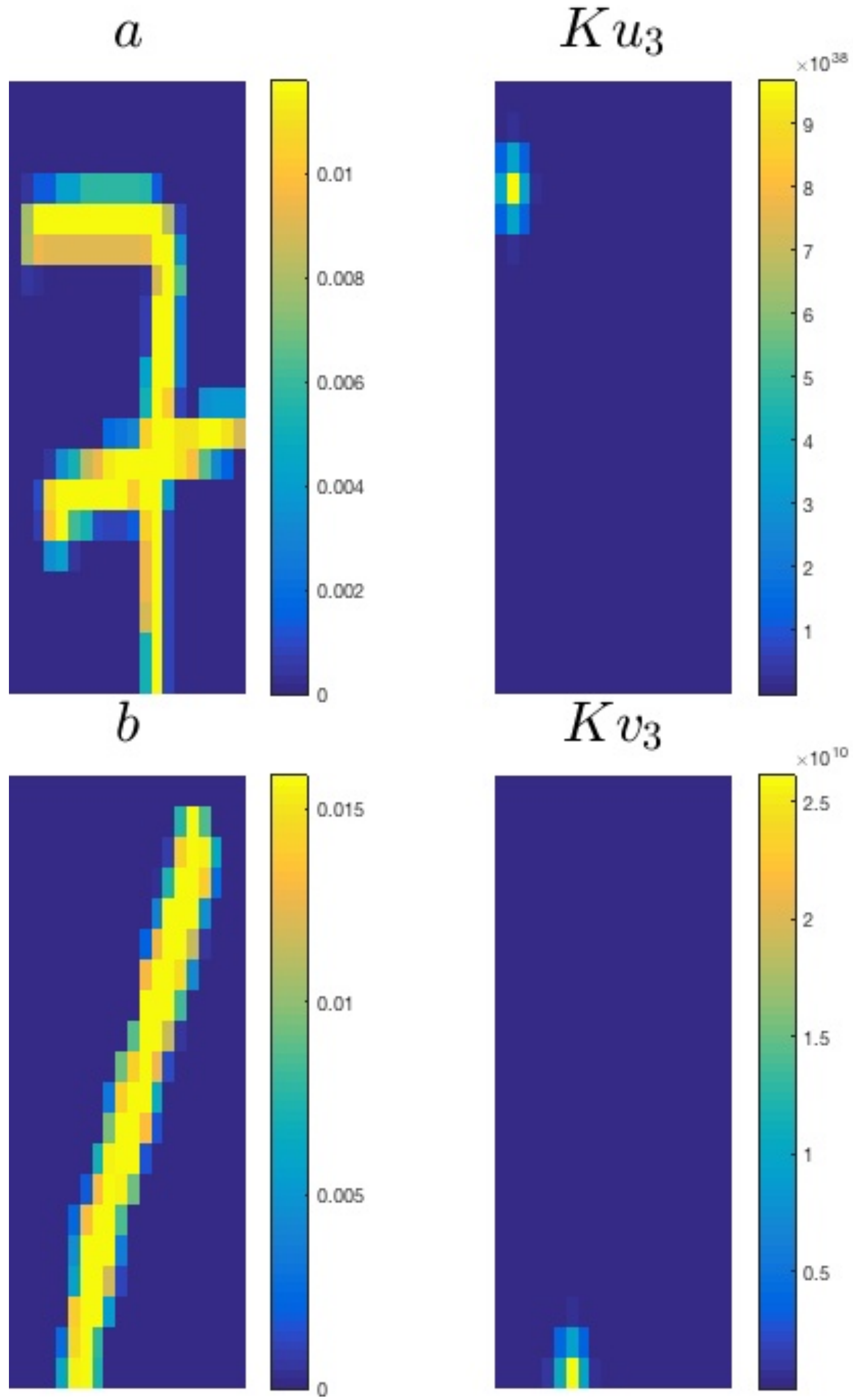


Very Fast EMD Approx. Solver



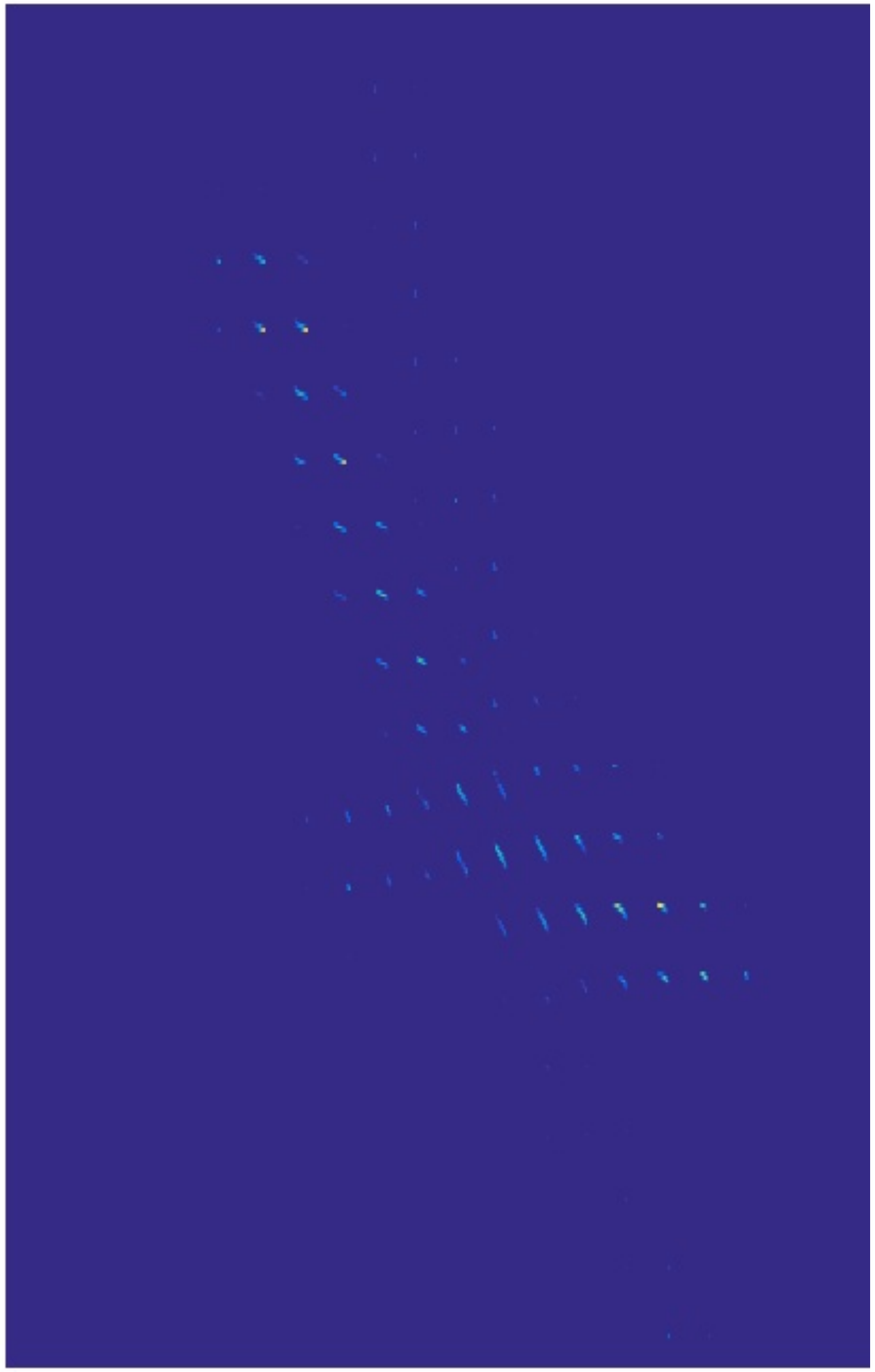


Very Fast EMD Approx. Solver

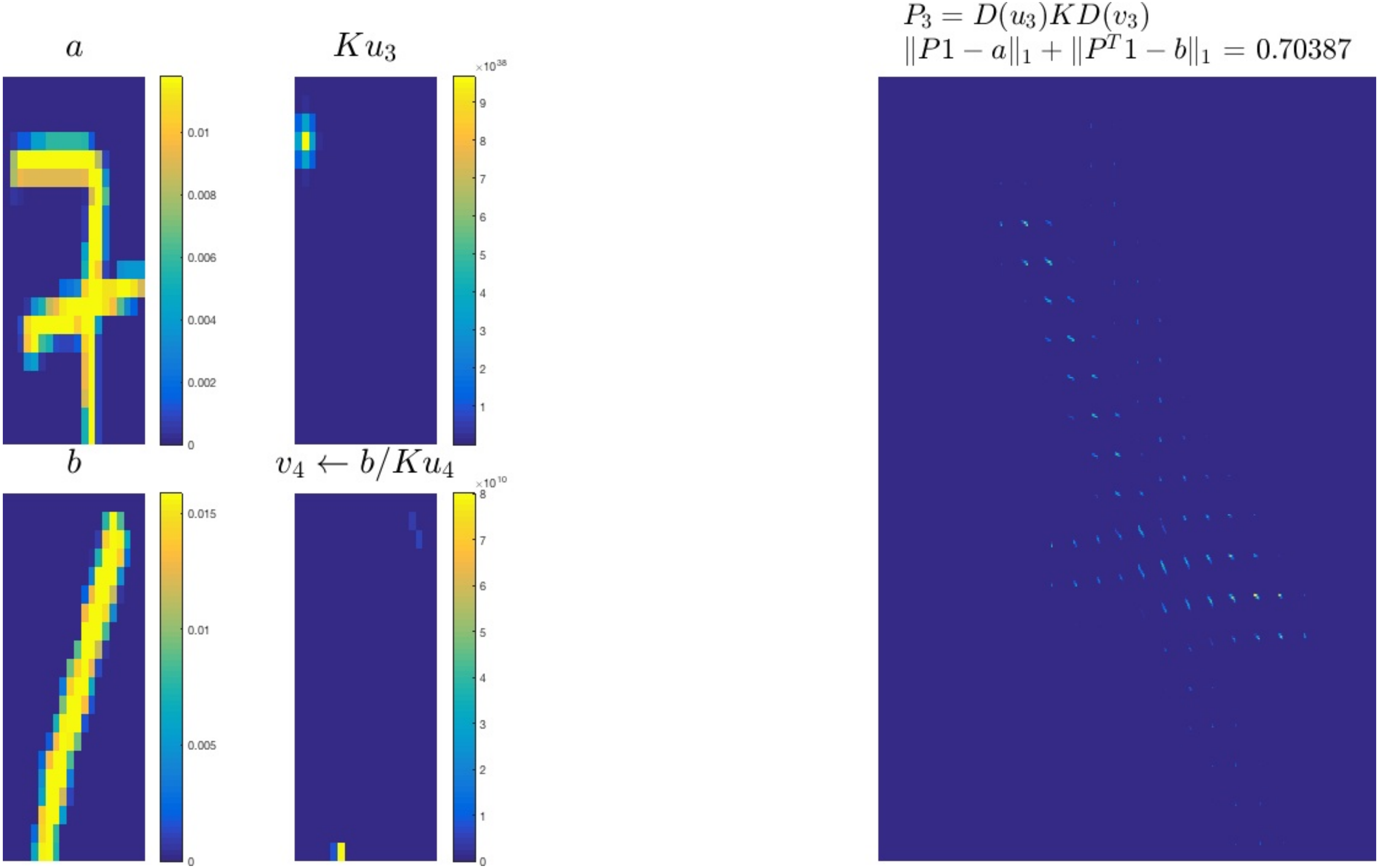


$P_2 = D(u_2)KD(v_2)$

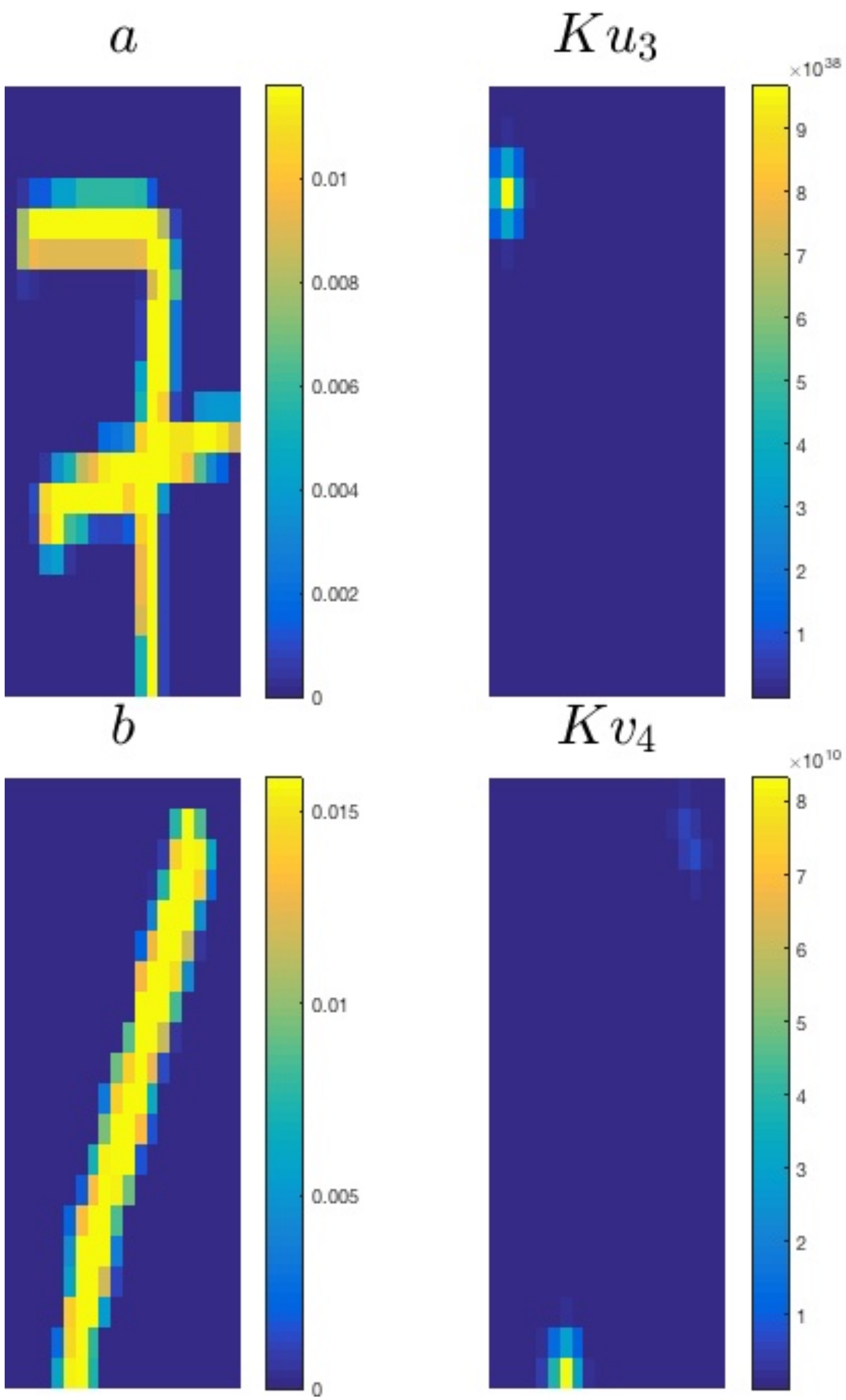
$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.91067$



Very Fast EMD Approx. Solver

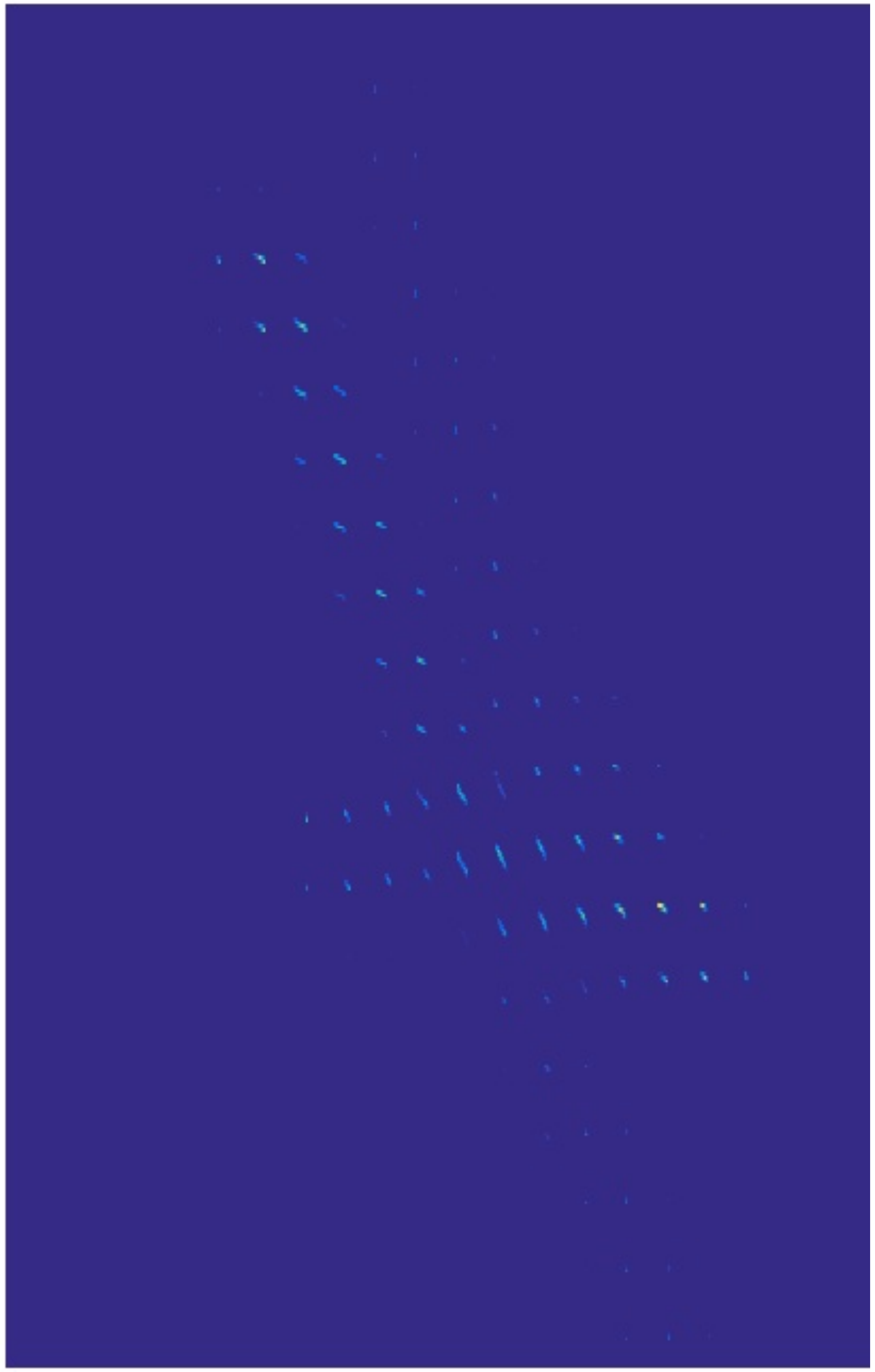


Very Fast EMD Approx. Solver

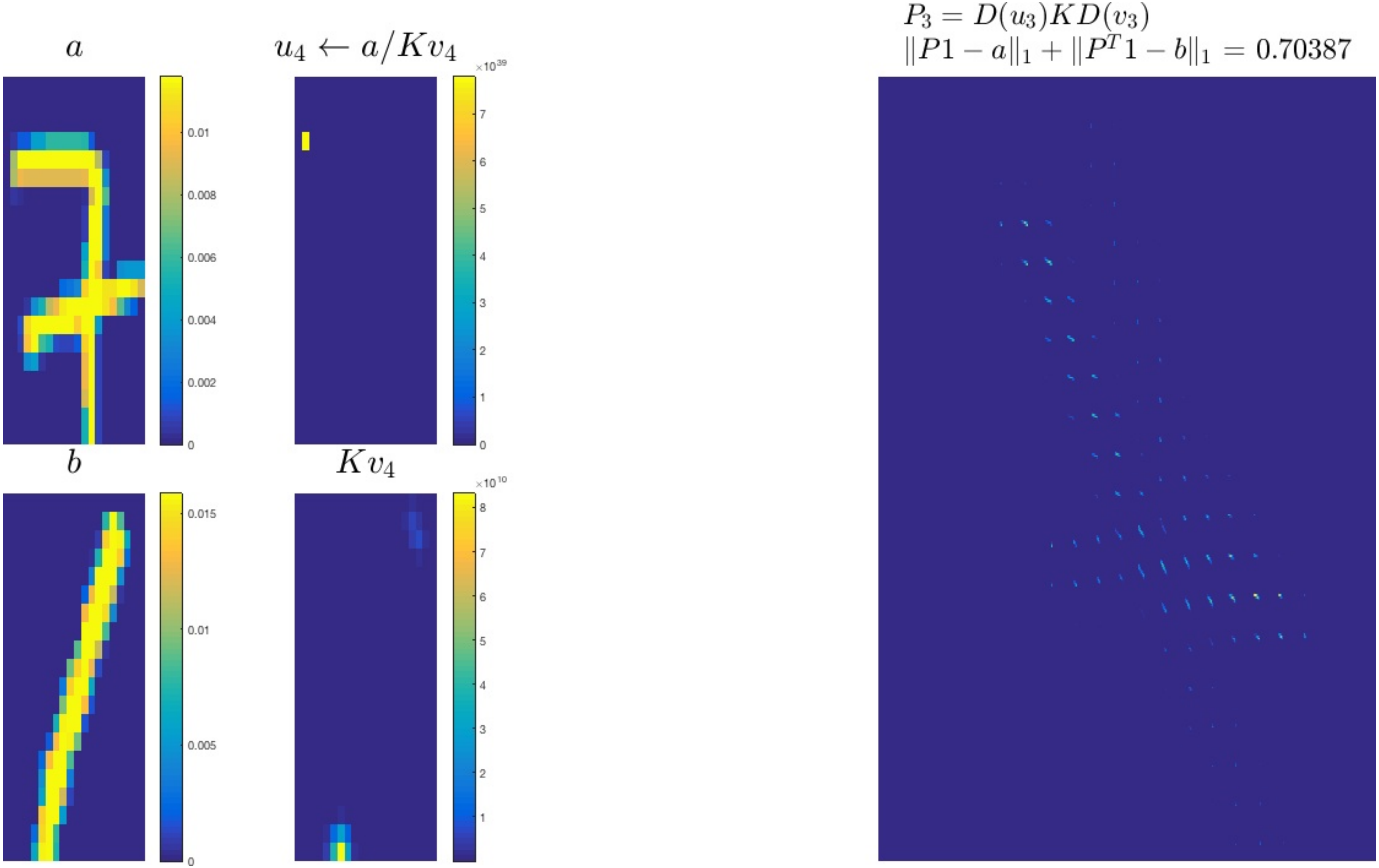


$P_3 = D(u_3)KD(v_3)$

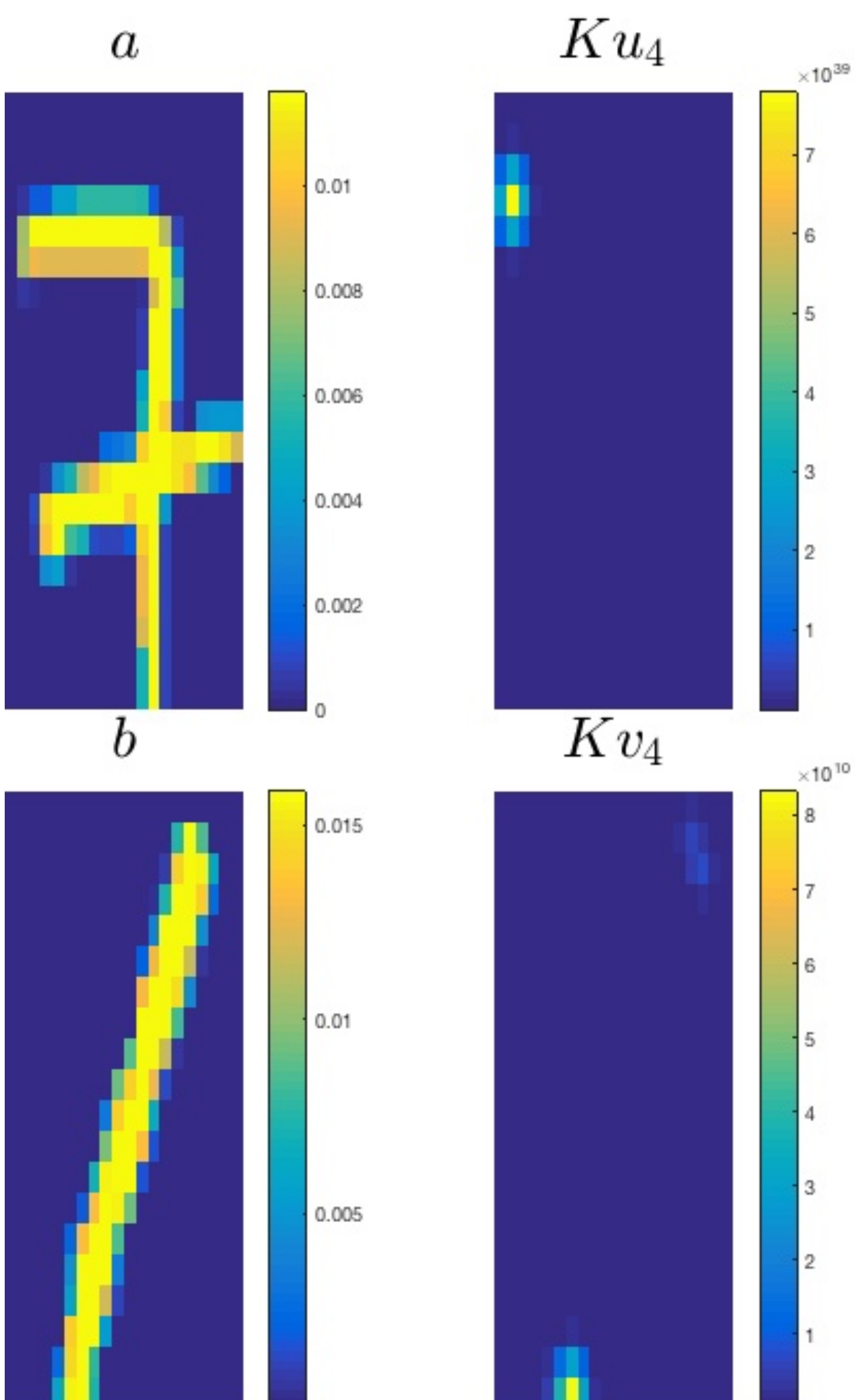
$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.70387$



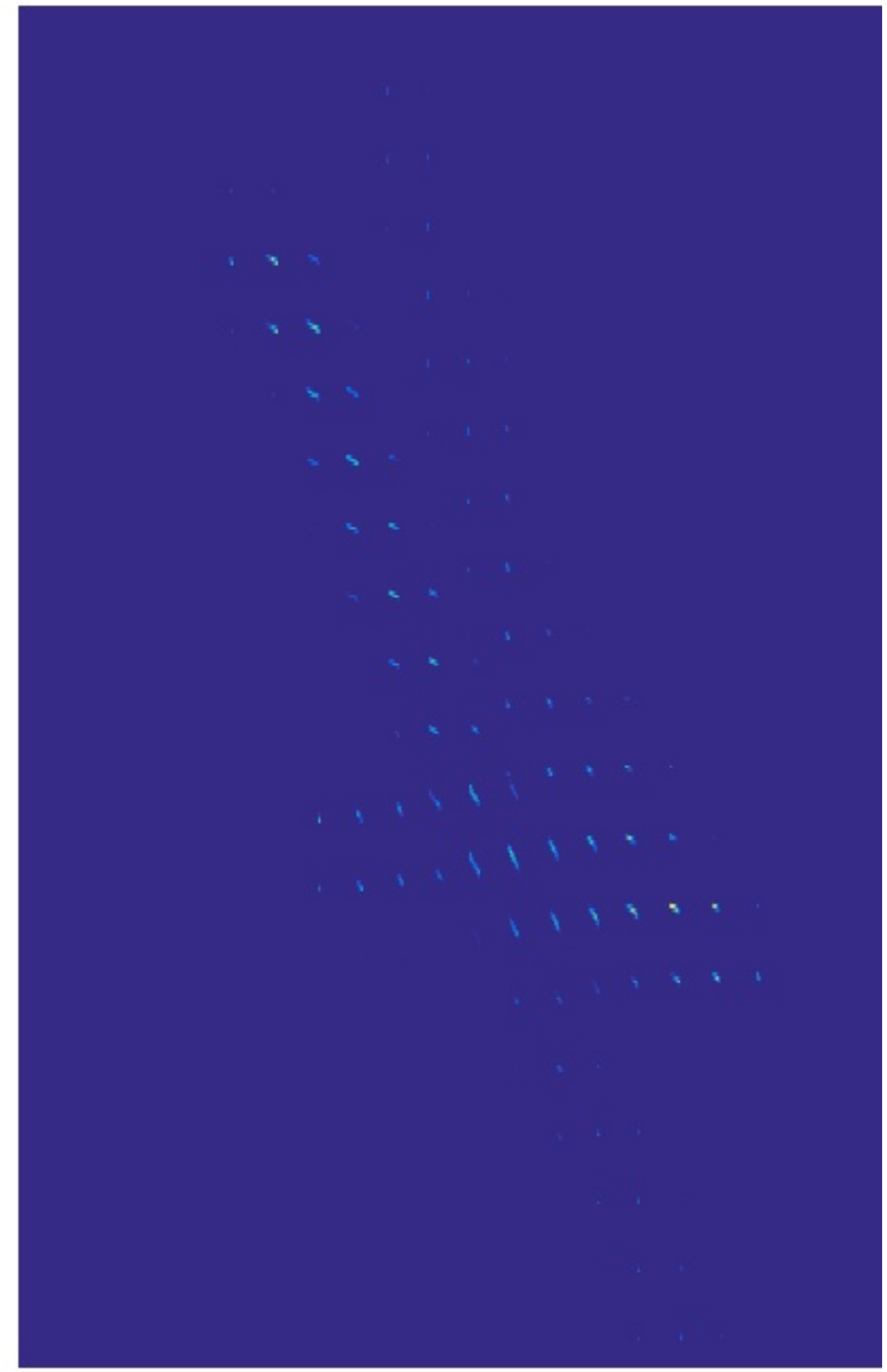
Very Fast EMD Approx. Solver



Very Fast EMD Approx. Solver



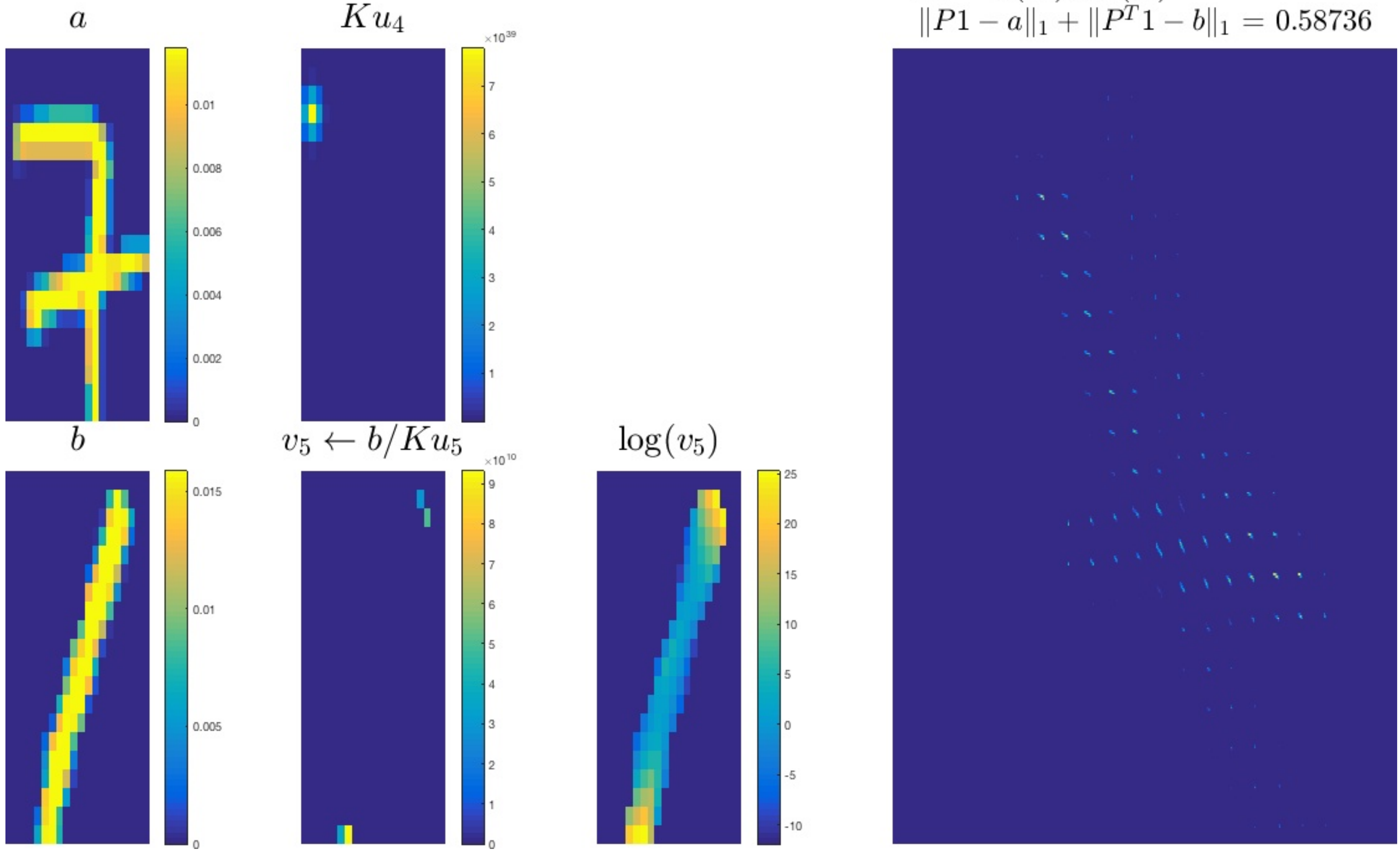
$P_3 = D(u_3)KD(v_3)$
$\|P\mathbb{1} - a\|_1 + \|P^T\mathbb{1} - b\|_1 = 0.70387$



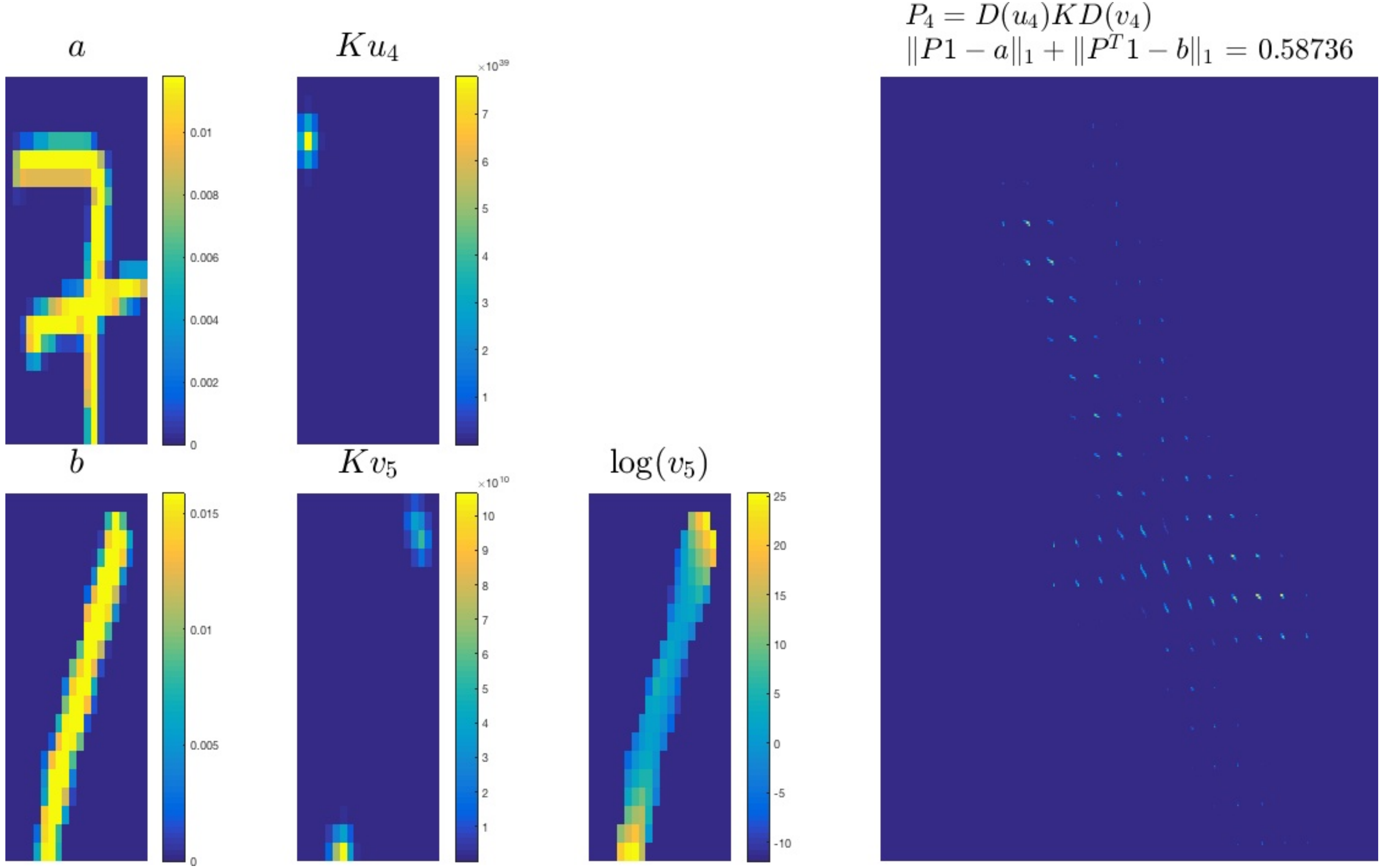
Very Fast EMD Approx. Solver

$P_4 = D(u_4)KD(v_4)$

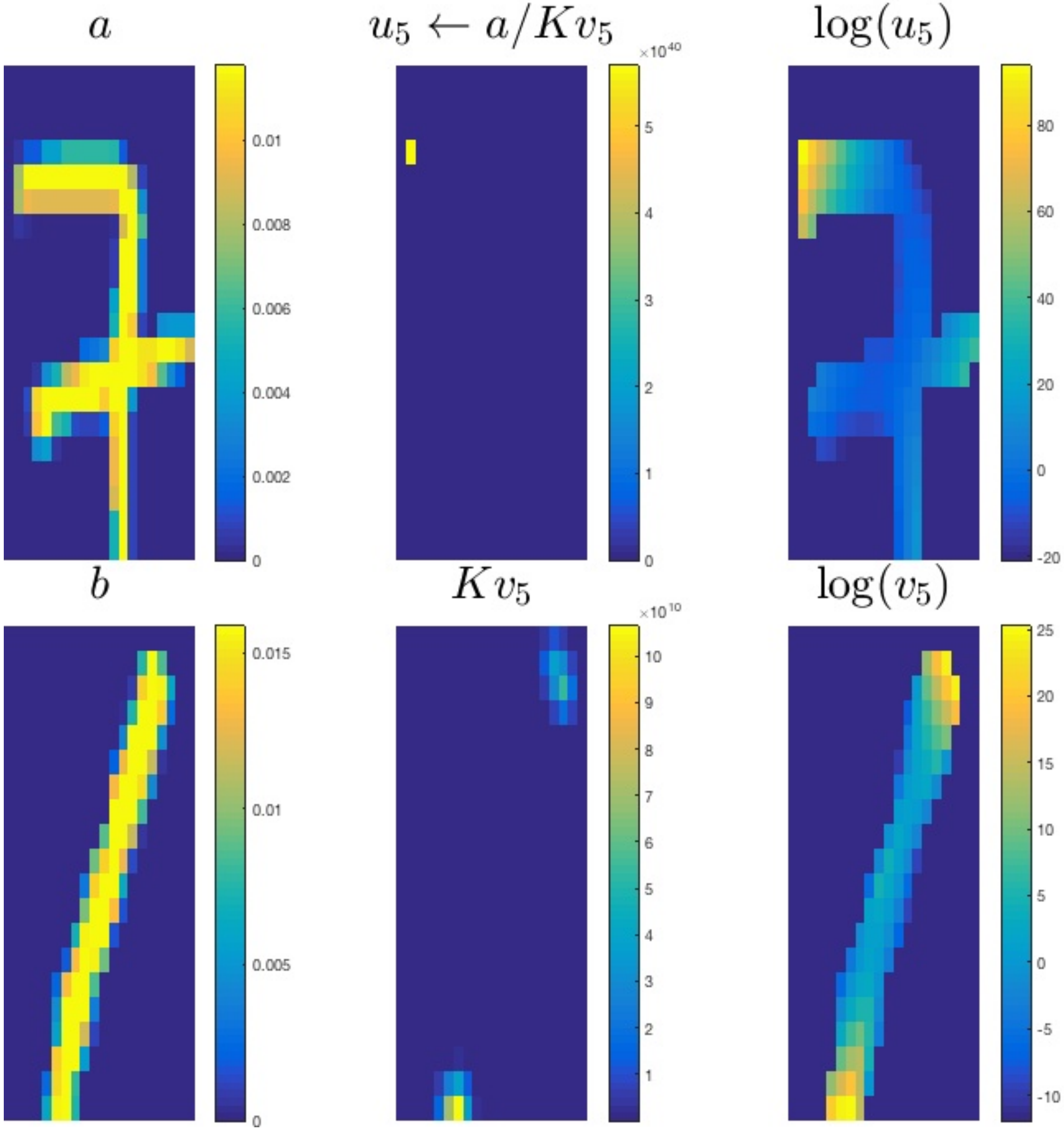
$\|P1 - a\|_1 + \|P^T1 - b\|_1 = 0.58736$



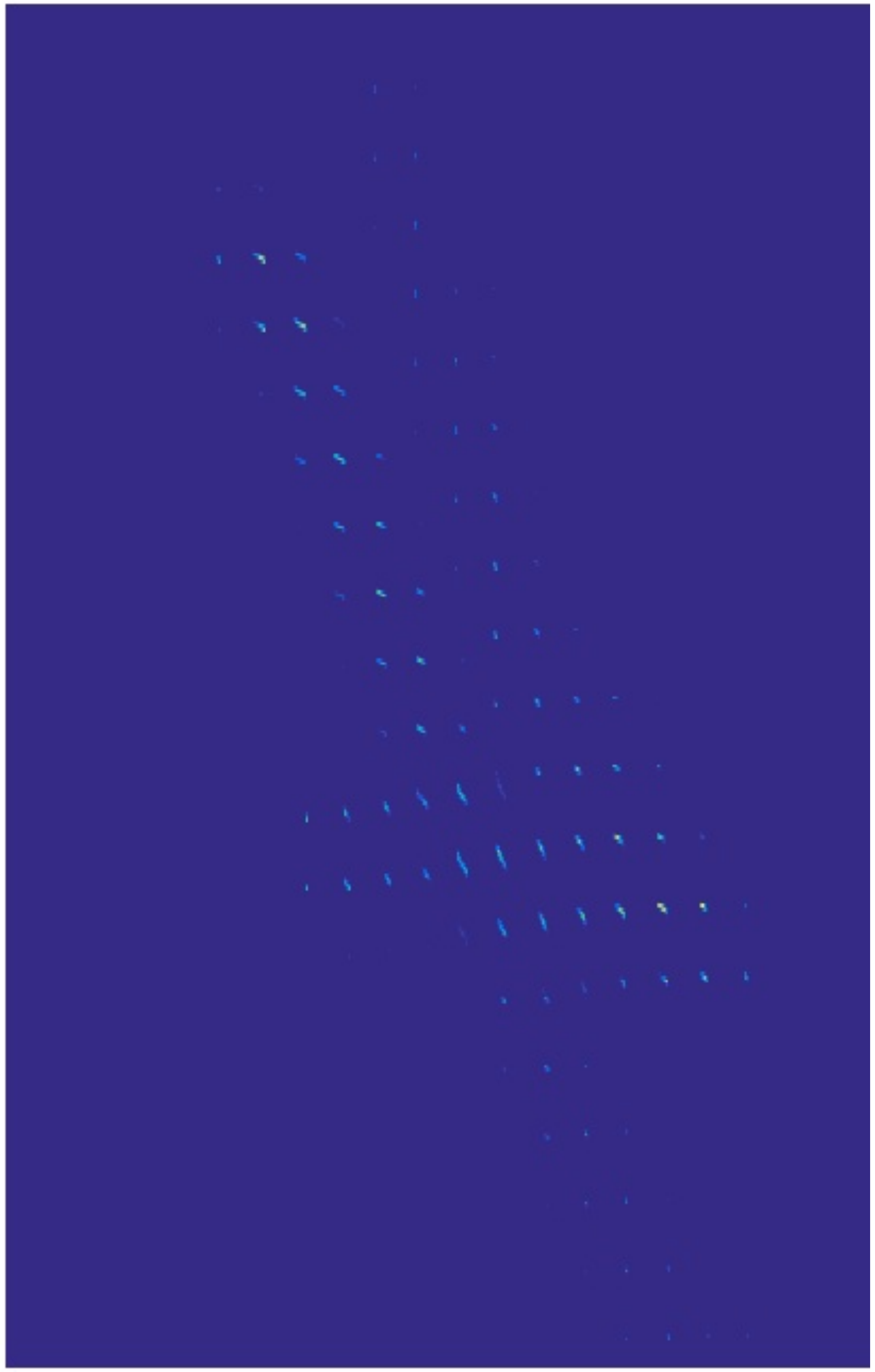
Very Fast EMD Approx. Solver



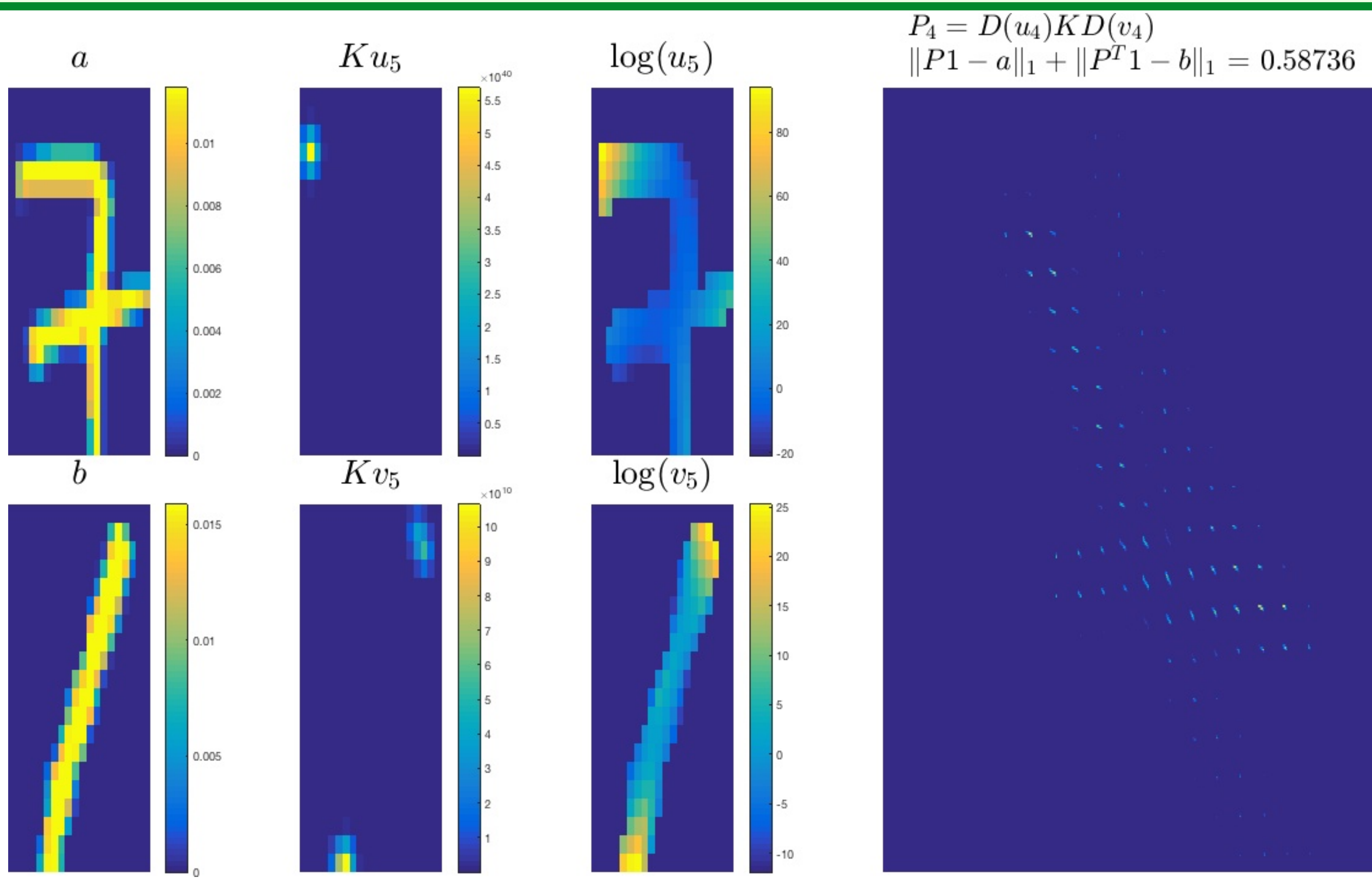
Very Fast EMD Approx. Solver



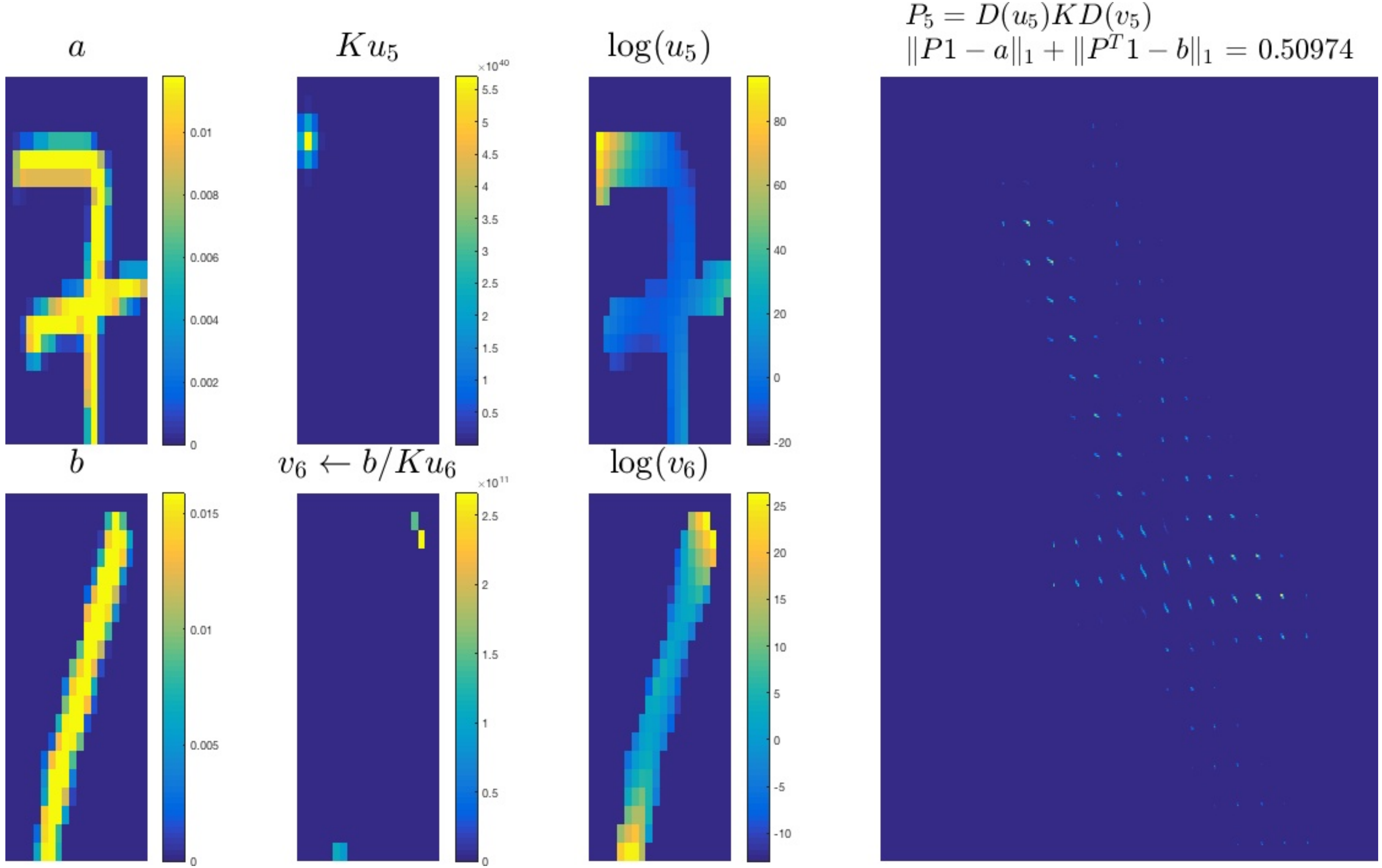
$P_4 = D(u_4)KD(v_4)$
$\|P\mathbb{1} - a\|_1 + \|P^T\mathbb{1} - b\|_1 = 0.58736$



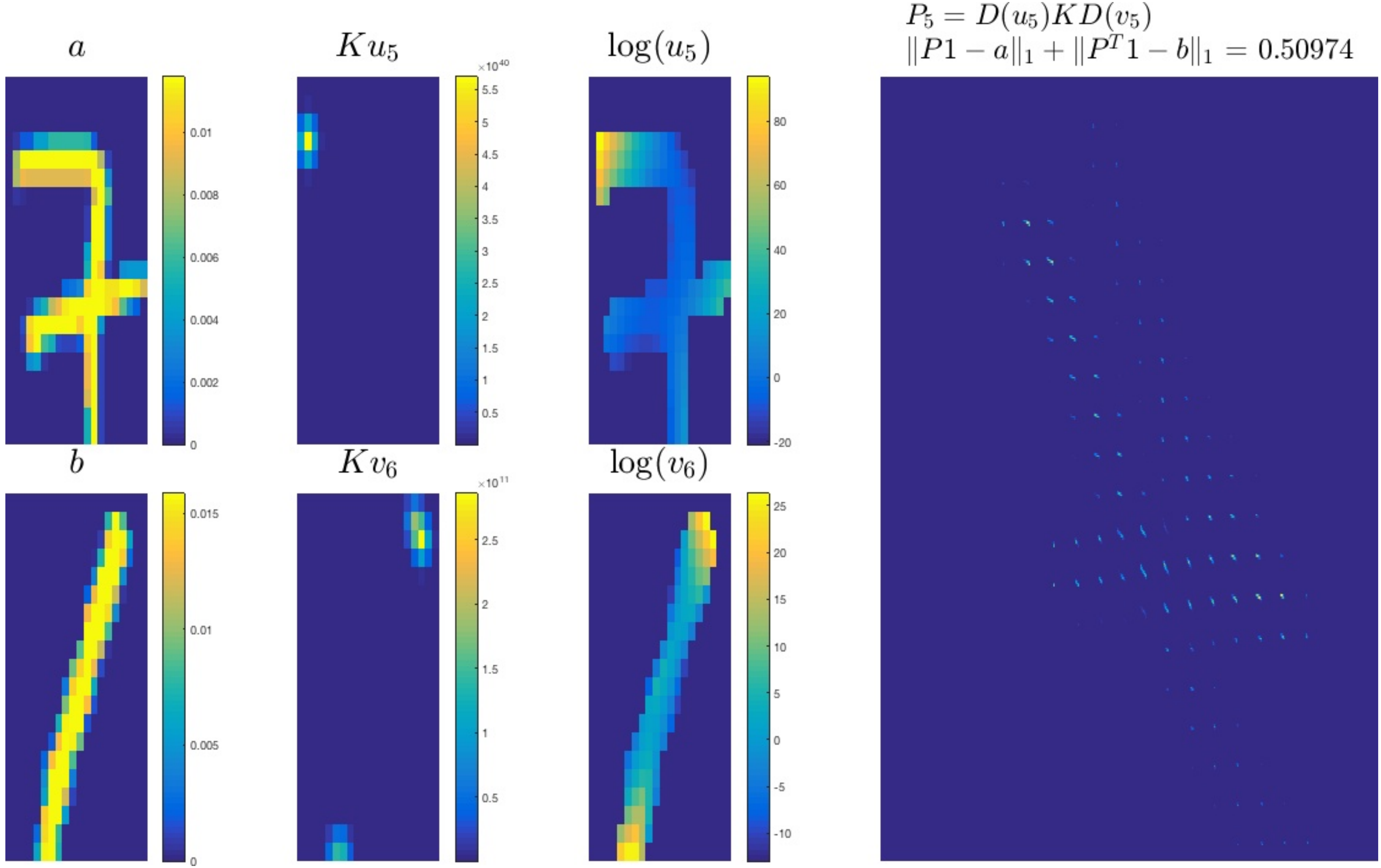
Very Fast EMD Approx. Solver



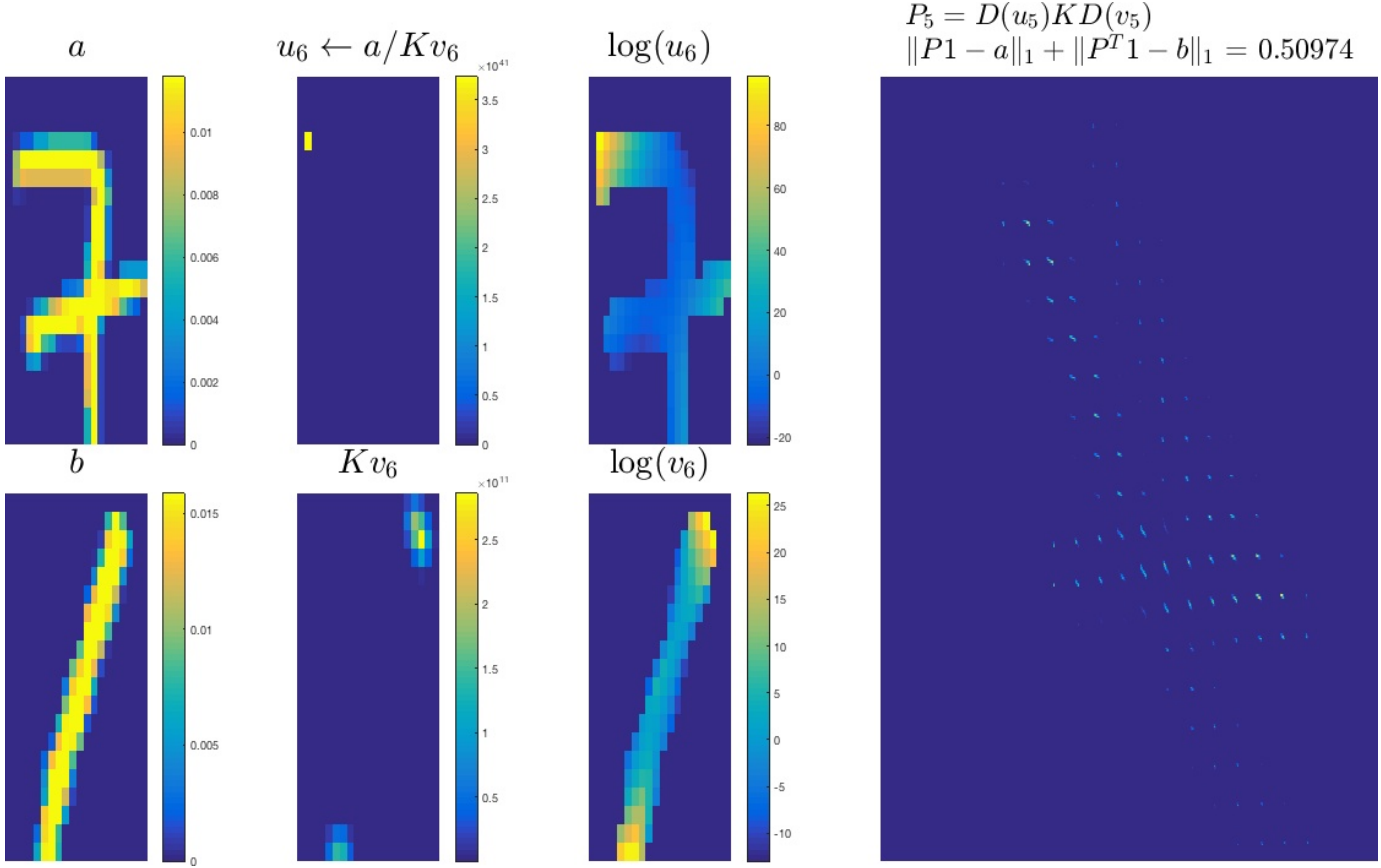
Very Fast EMD Approx. Solver



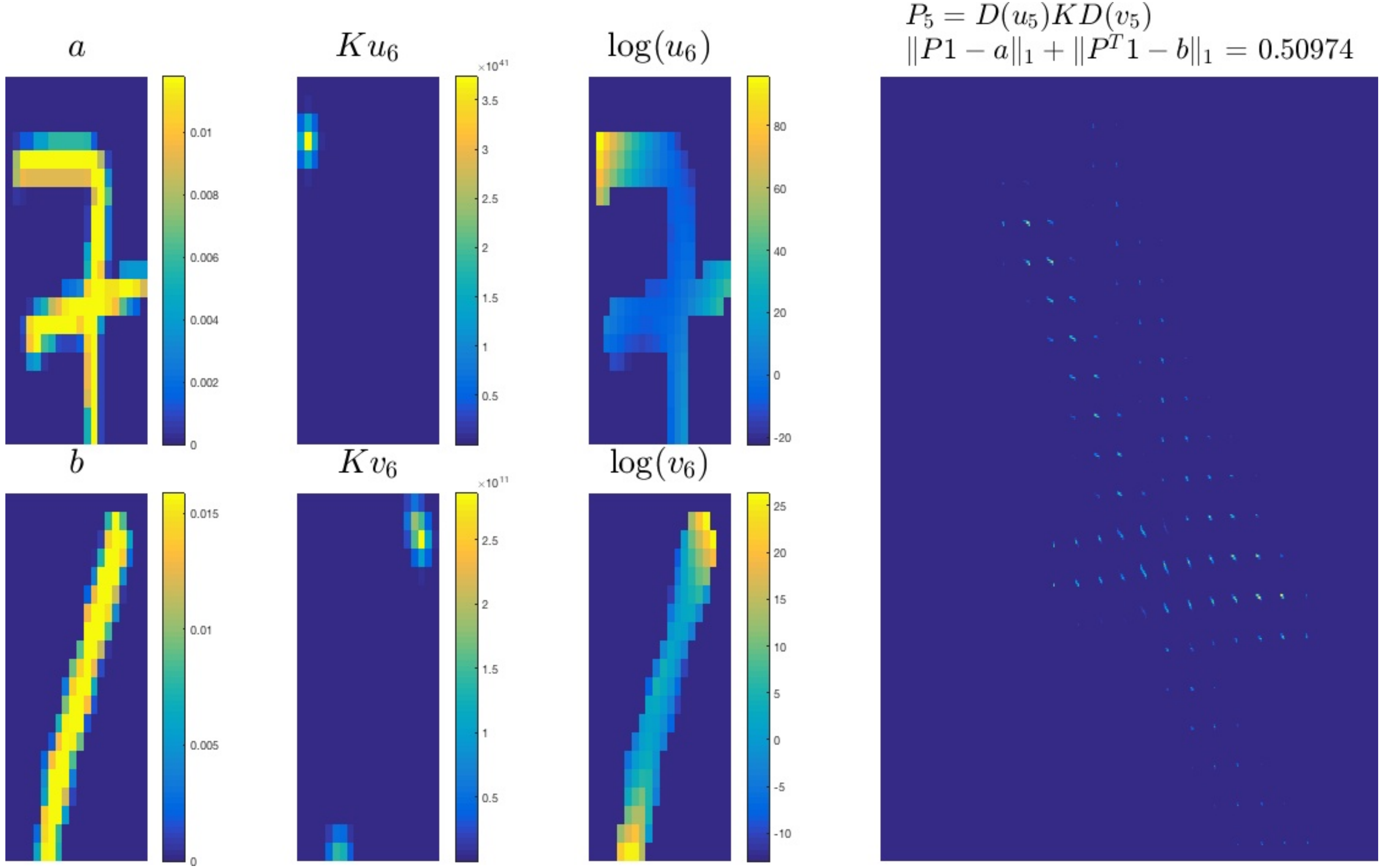
Very Fast EMD Approx. Solver



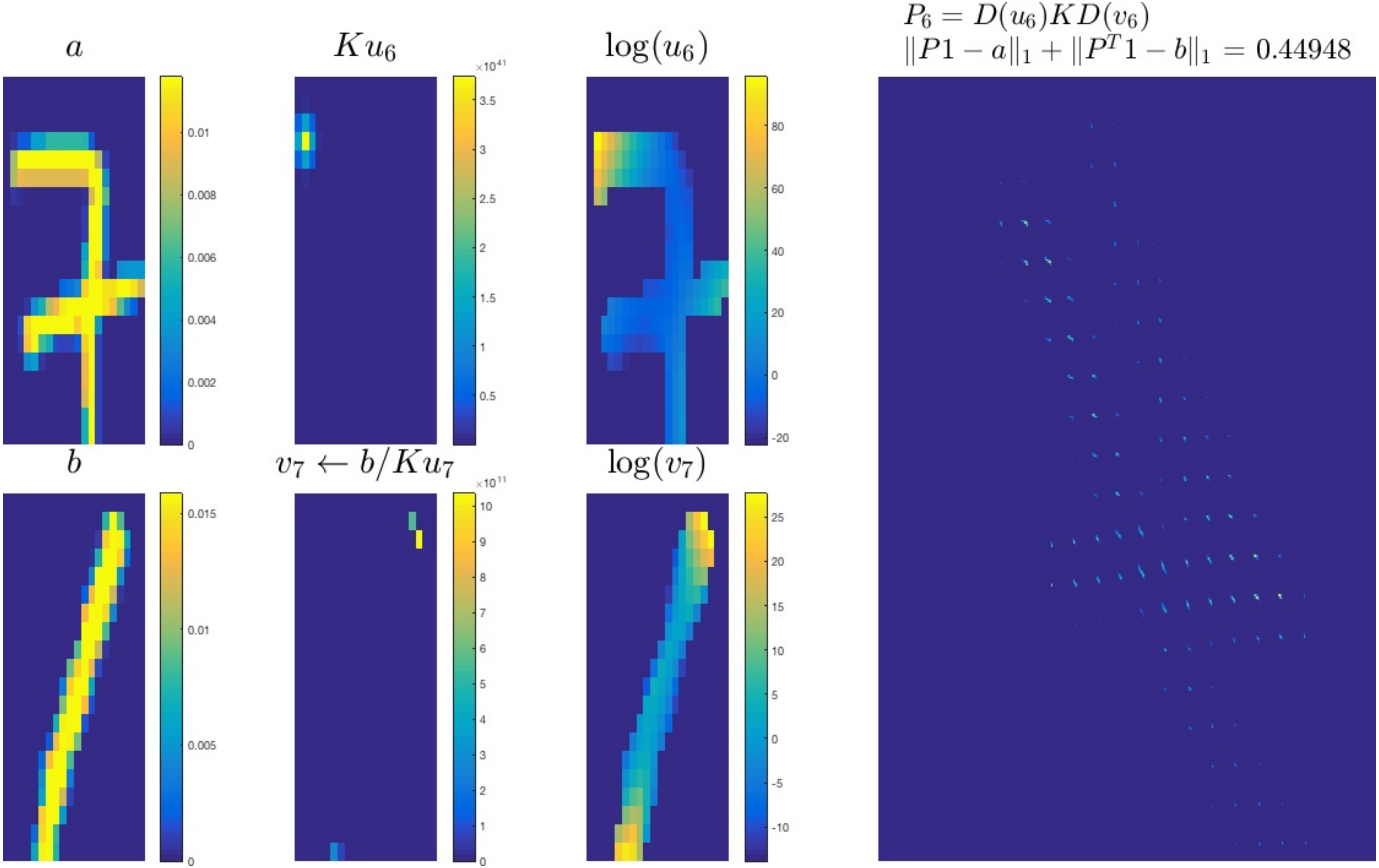
Very Fast EMD Approx. Solver



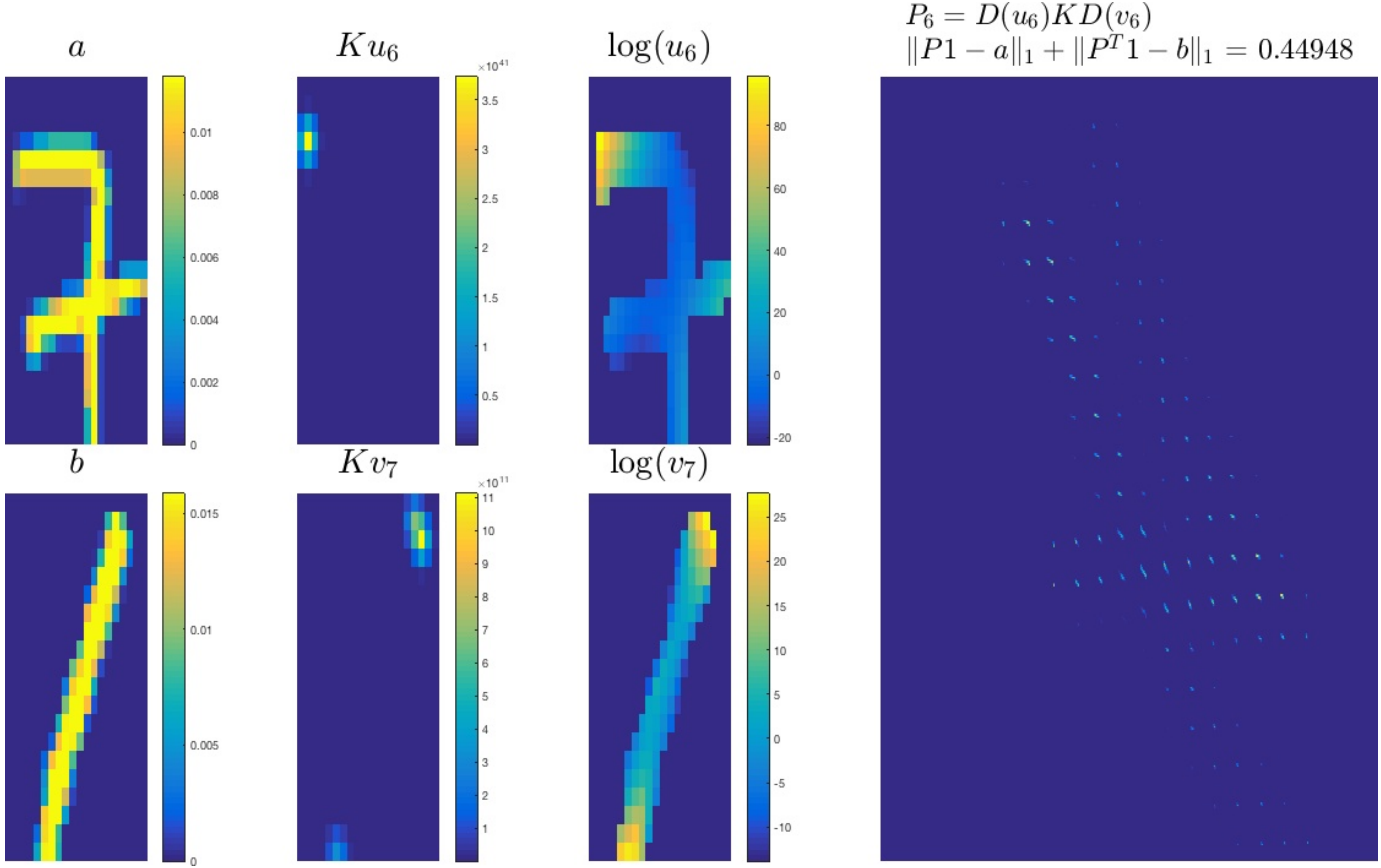
Very Fast EMD Approx. Solver



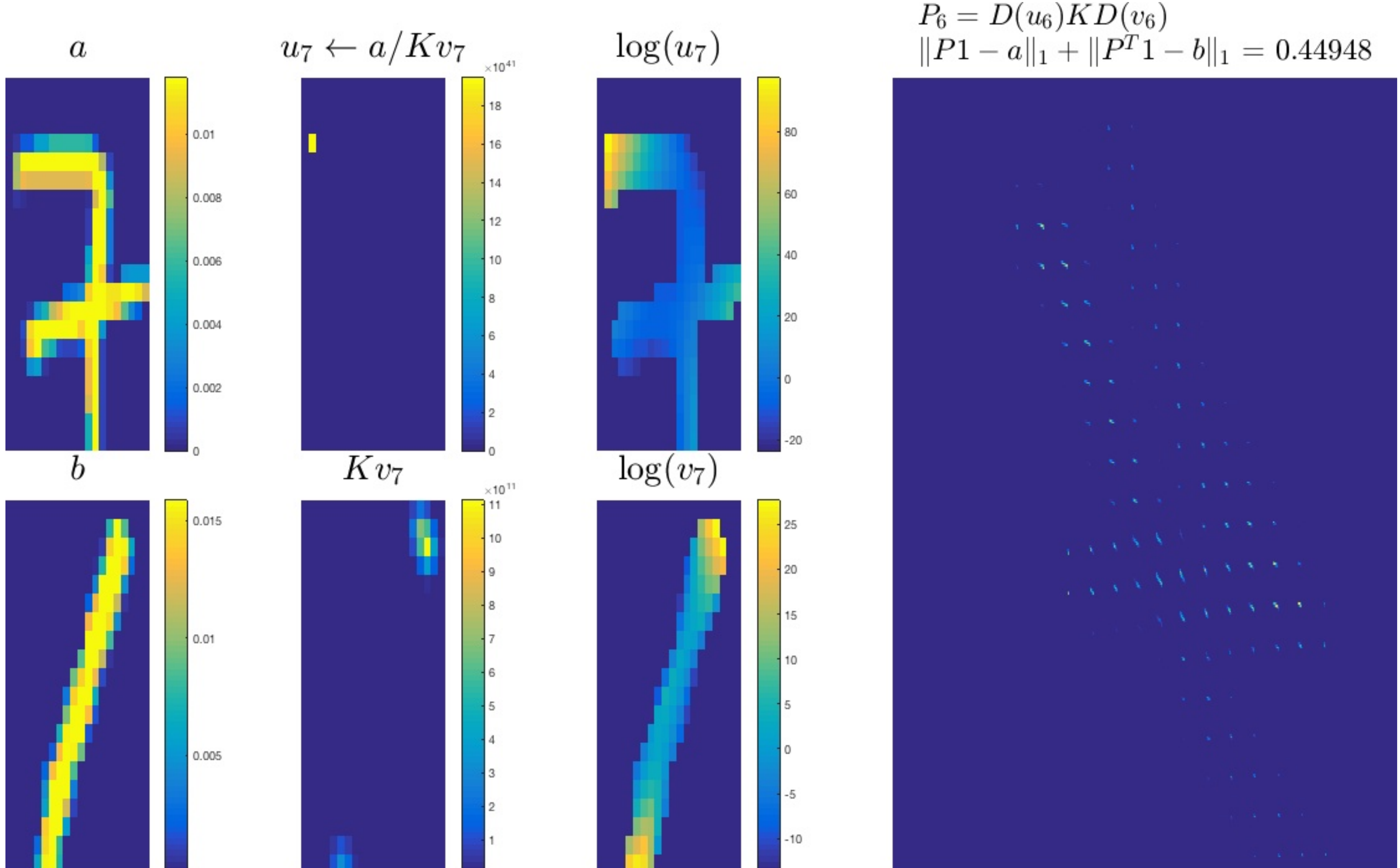
Very Fast EMD Approx. Solver



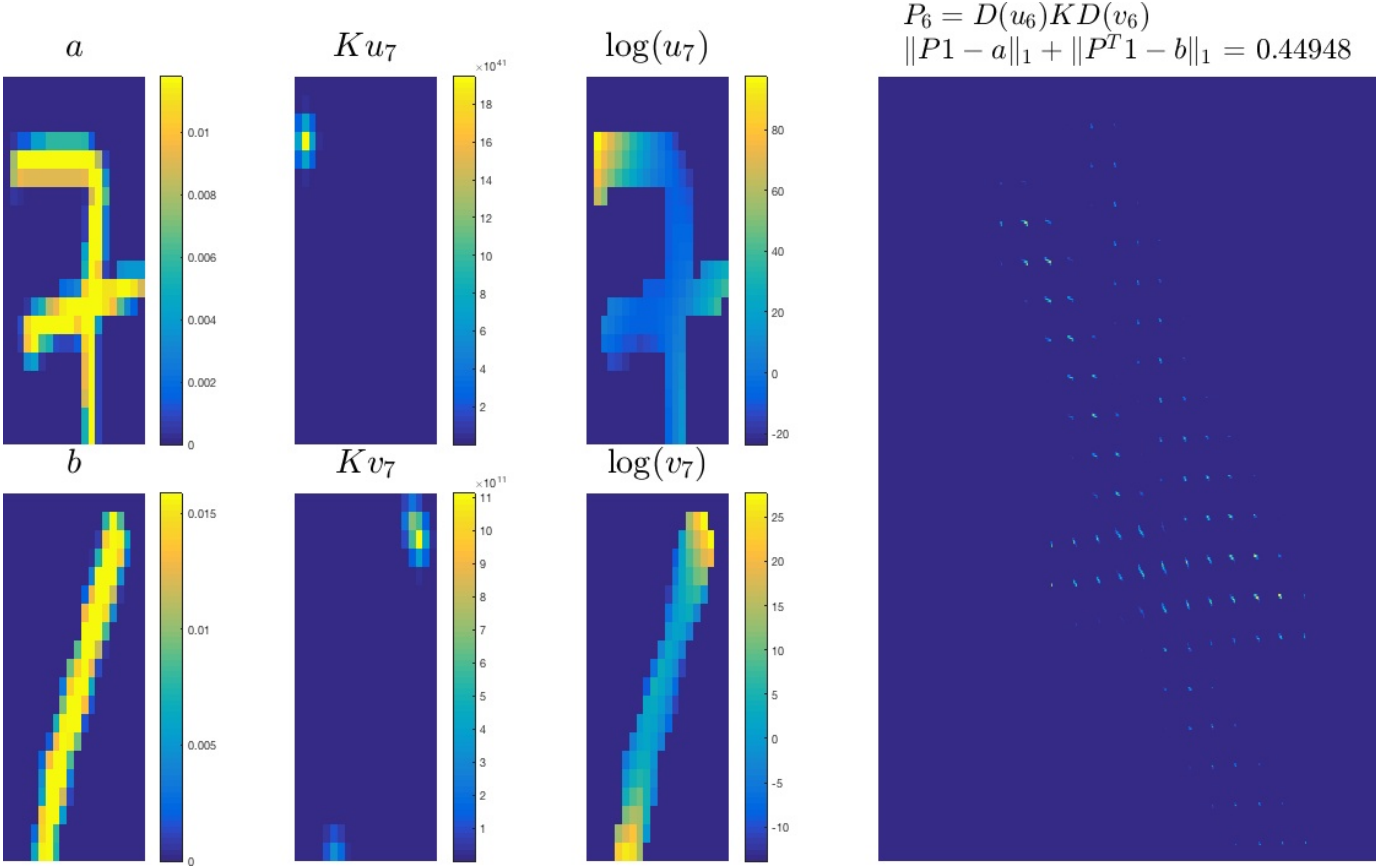
Very Fast EMD Approx. Solver



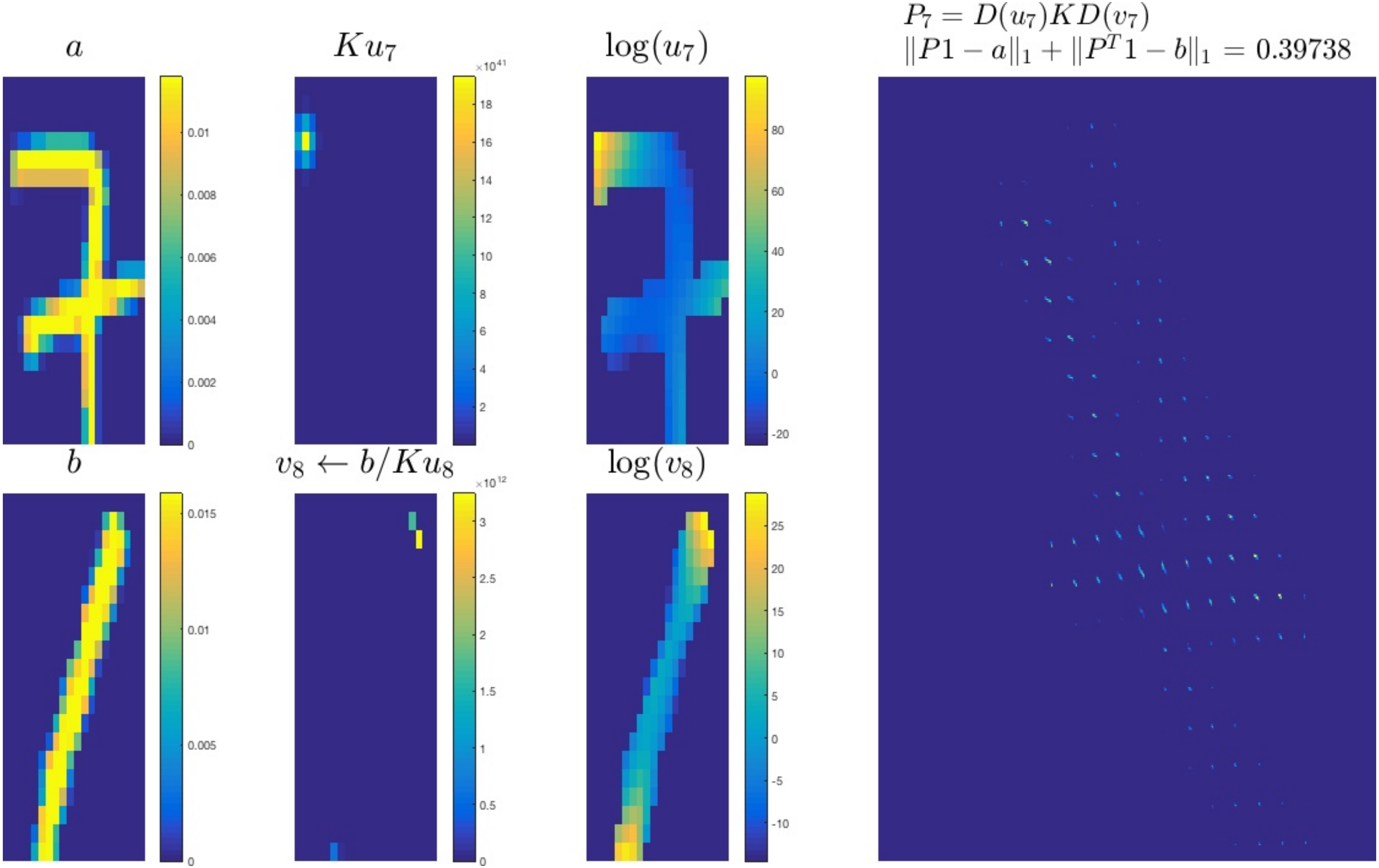
Very Fast EMD Approx. Solver



Very Fast EMD Approx. Solver



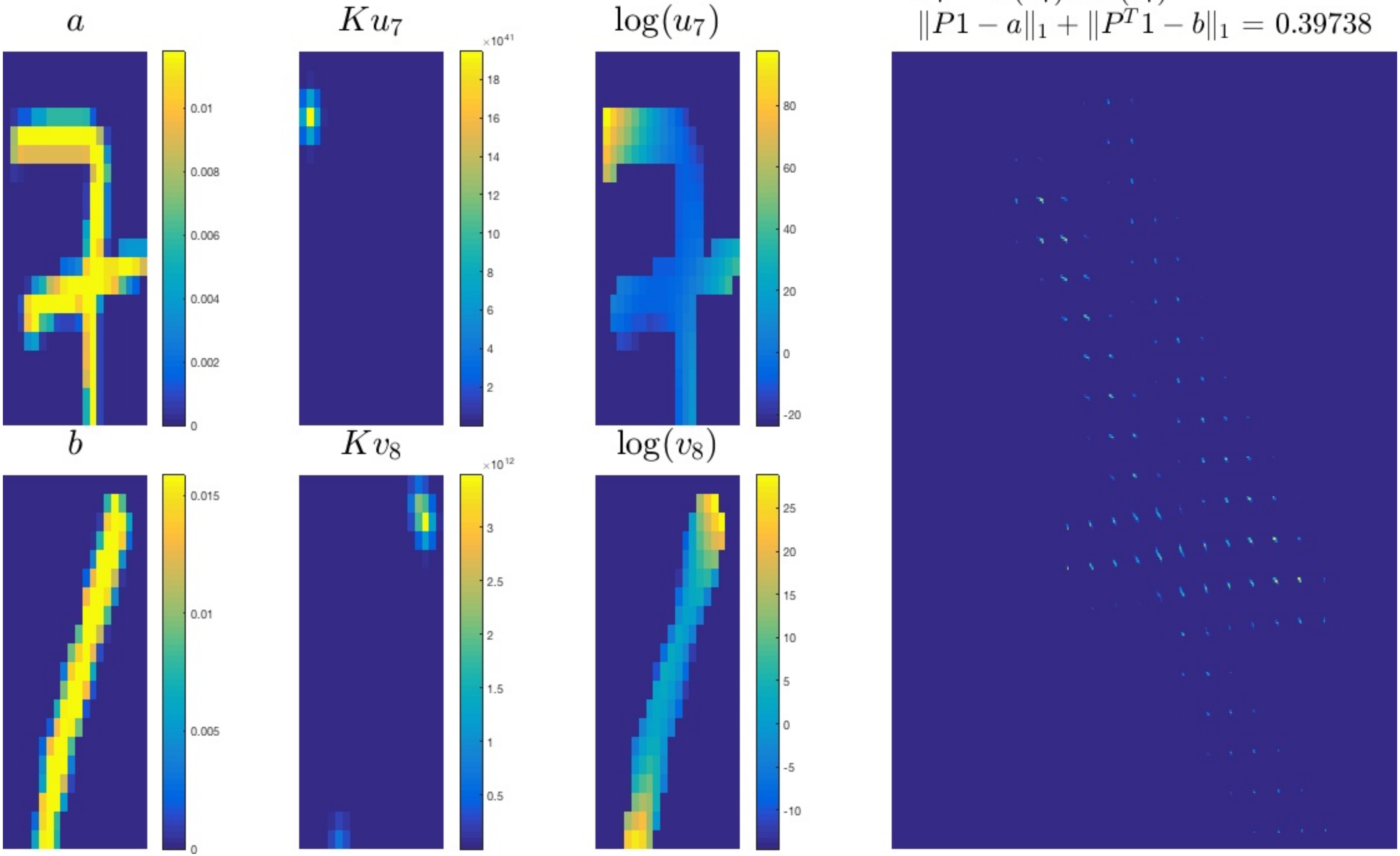
Very Fast EMD Approx. Solver



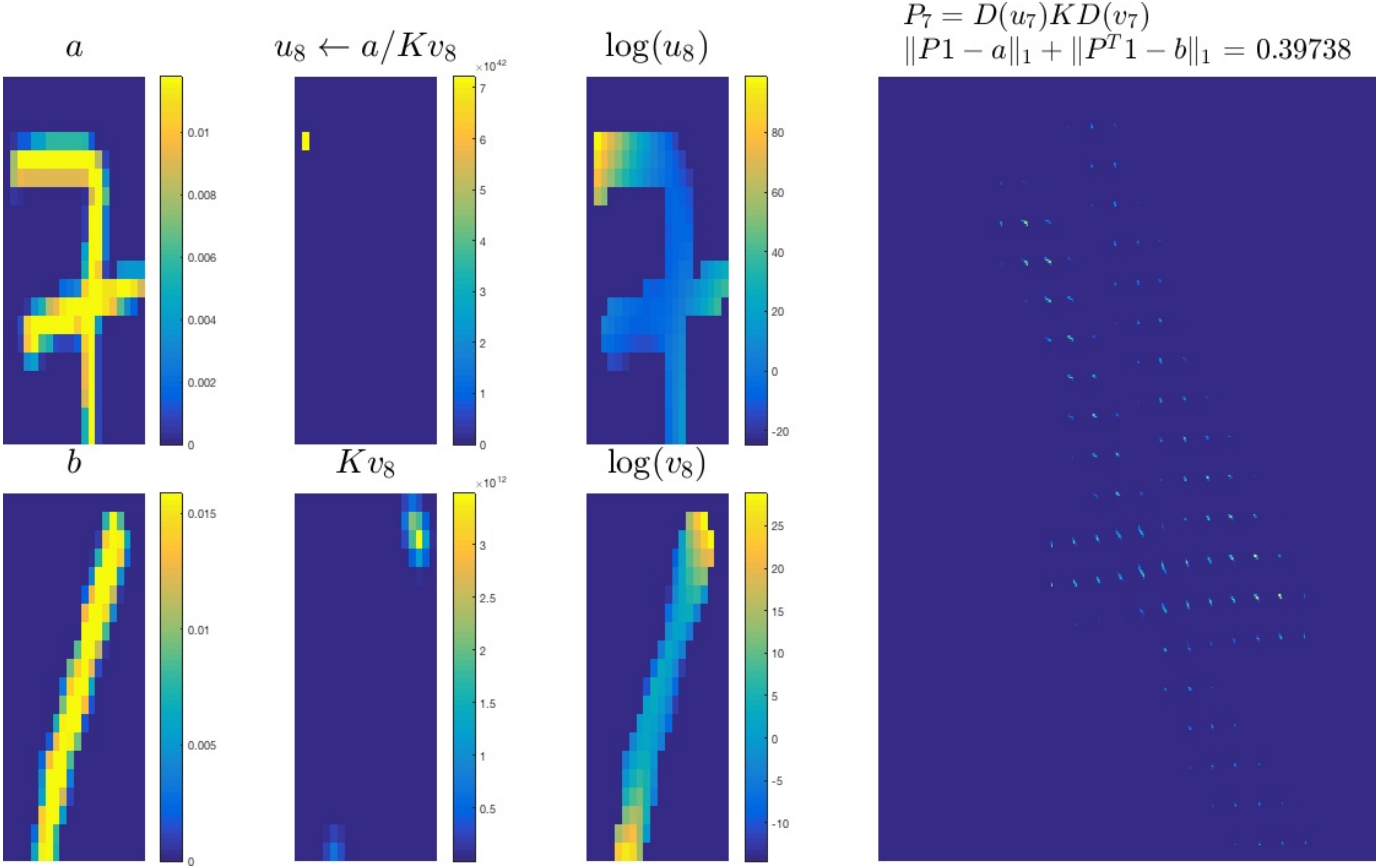
Very Fast EMD Approx. Solver

$P_7 = D(u_7)KD(v_7)$

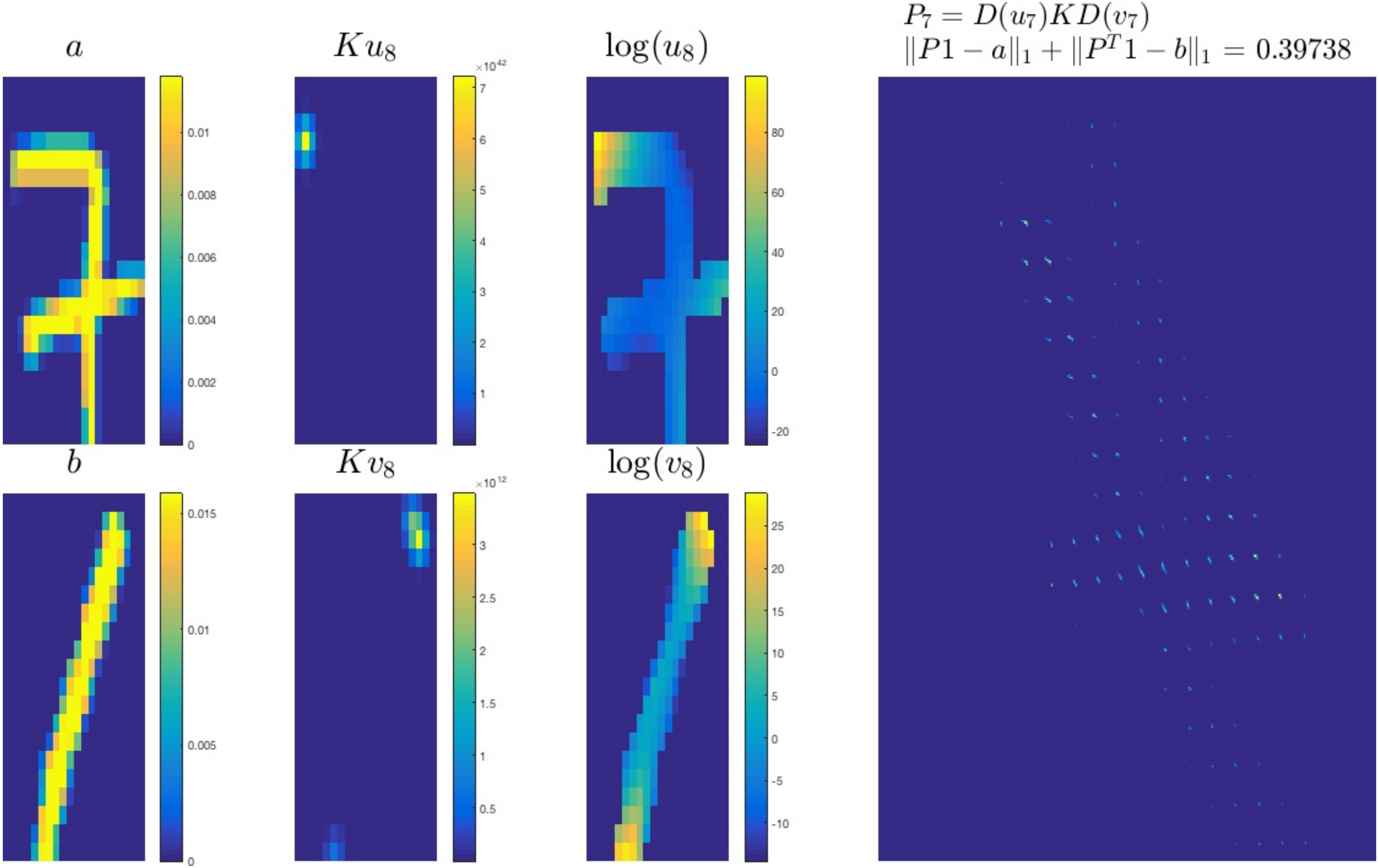
$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.39738$



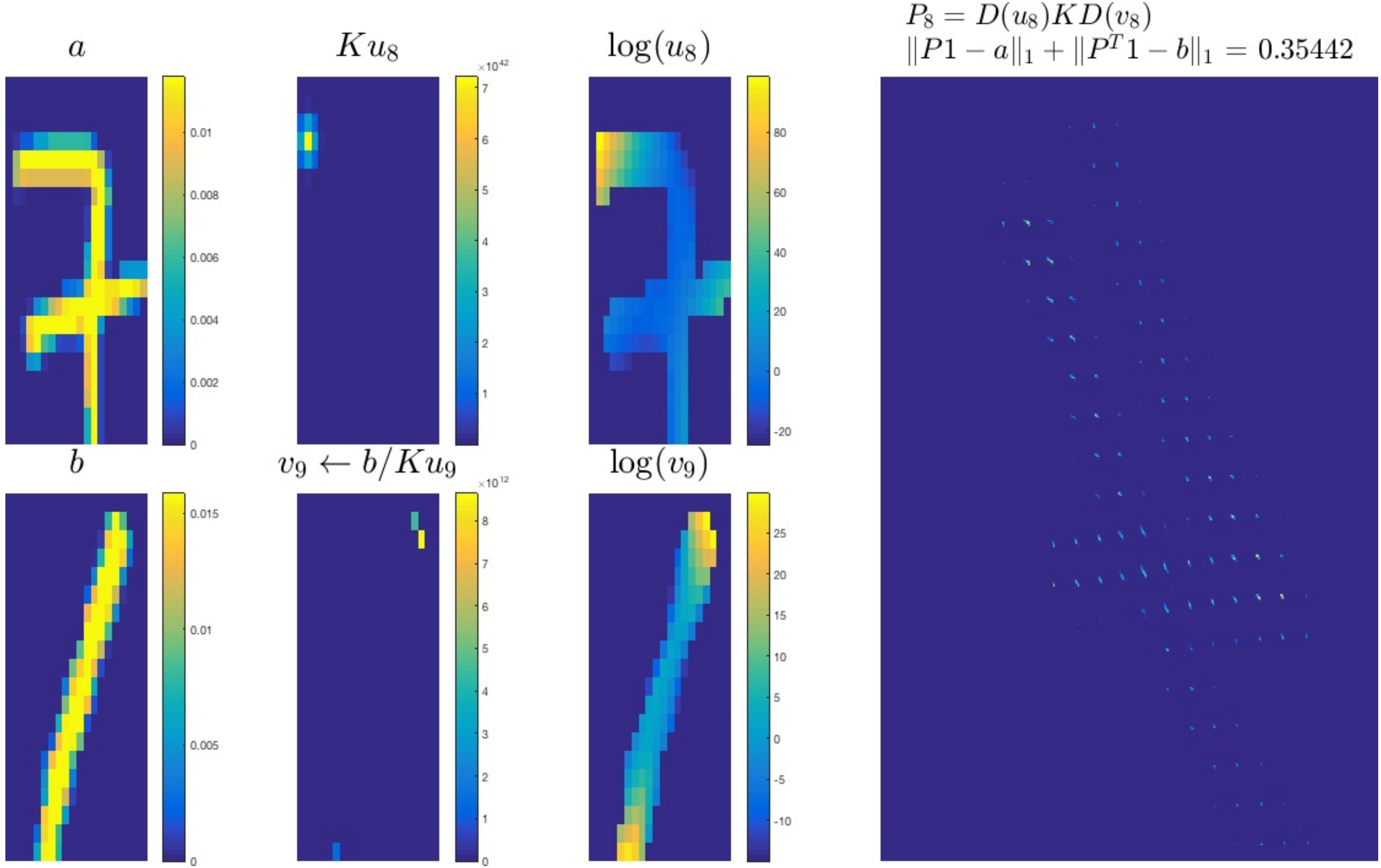
Very Fast EMD Approx. Solver



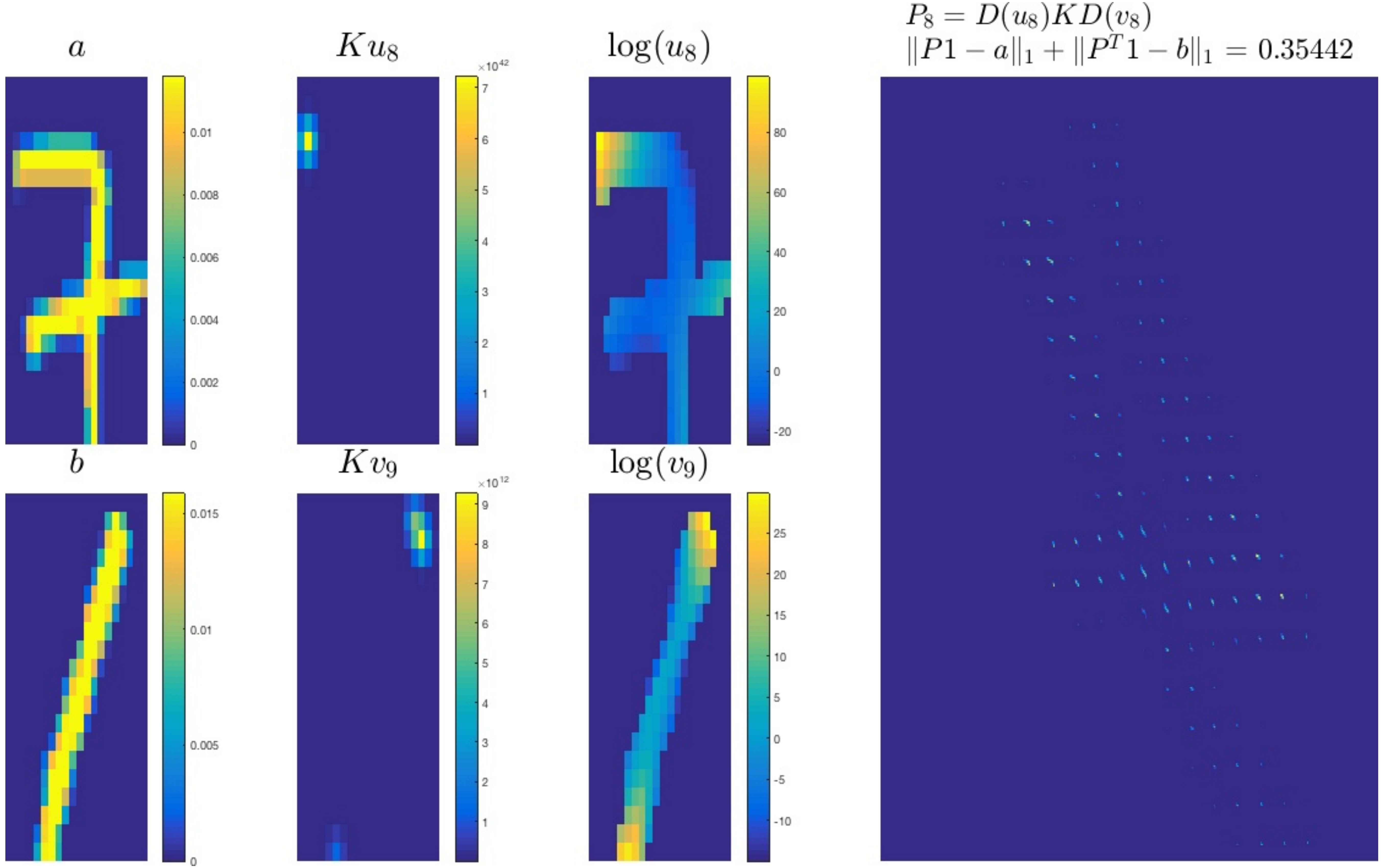
Very Fast EMD Approx. Solver



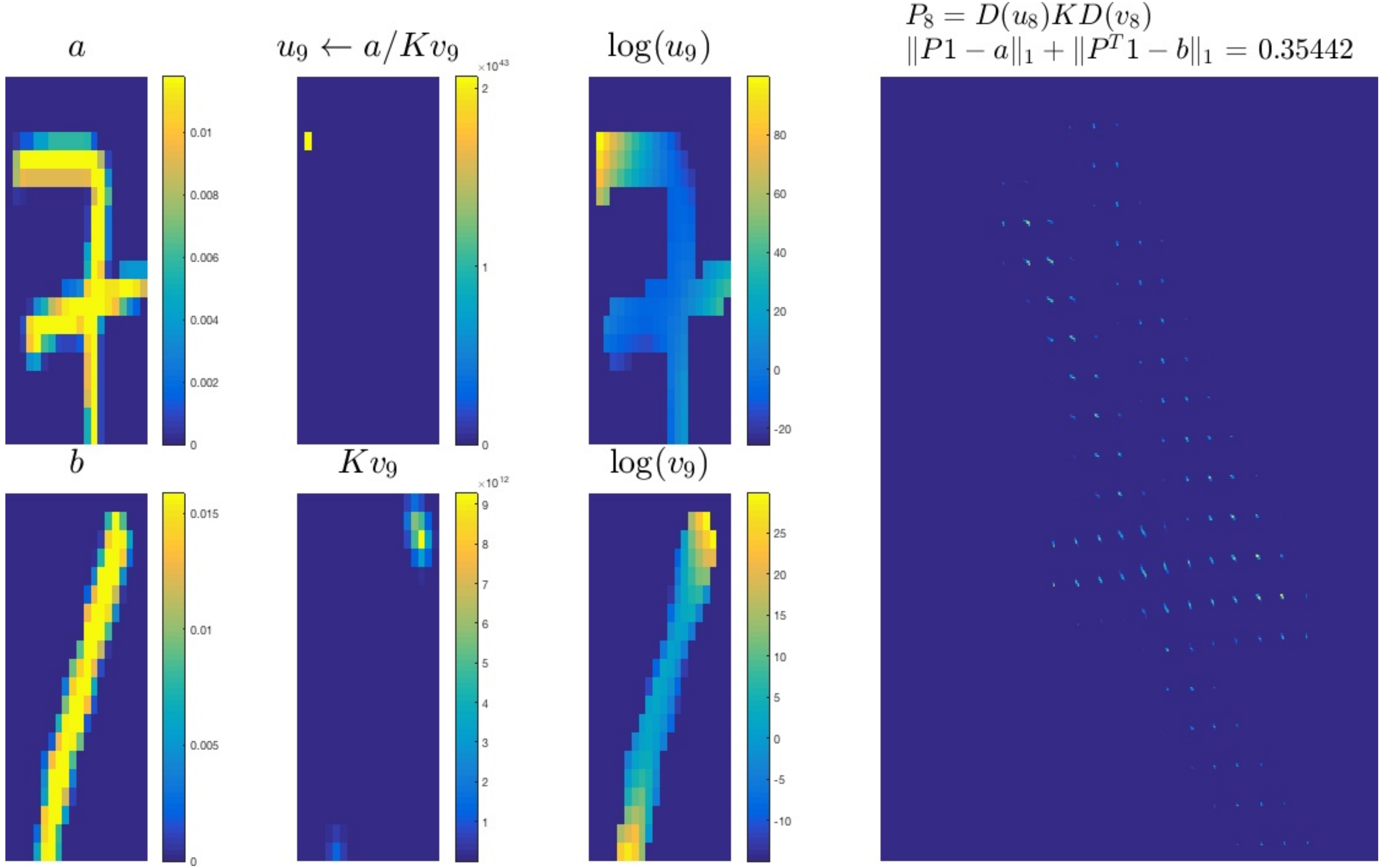
Very Fast EMD Approx. Solver



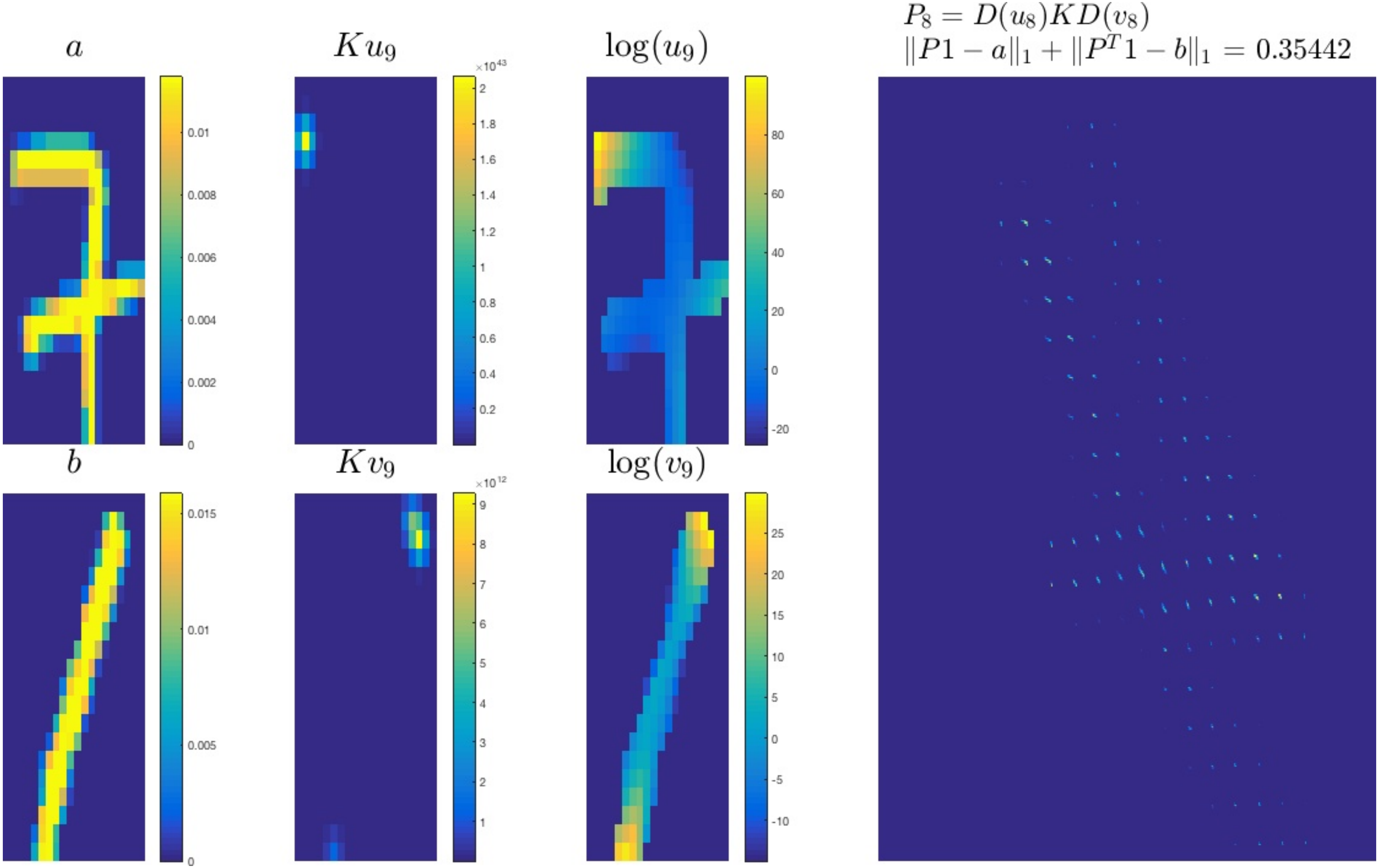
Very Fast EMD Approx. Solver



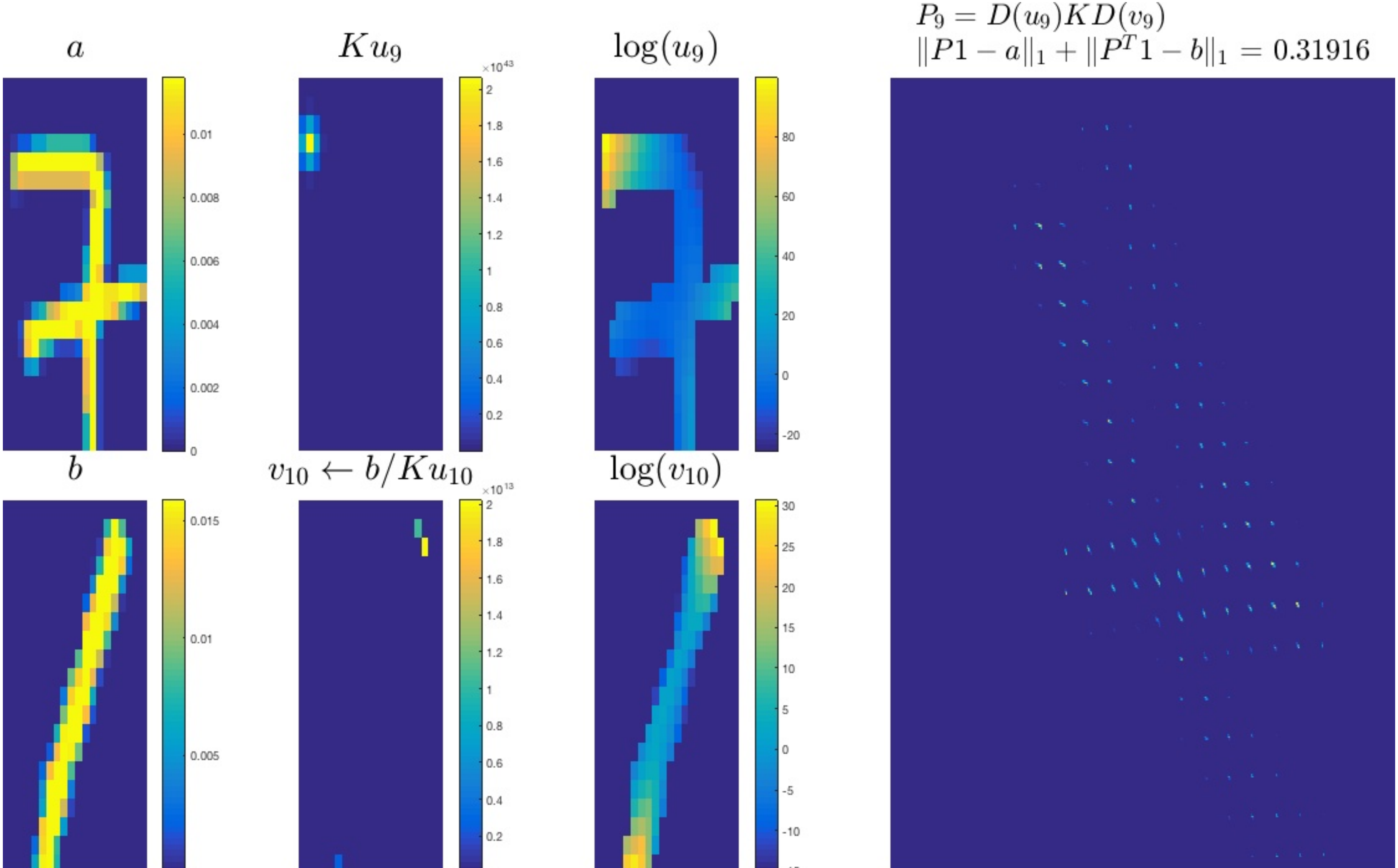
Very Fast EMD Approx. Solver



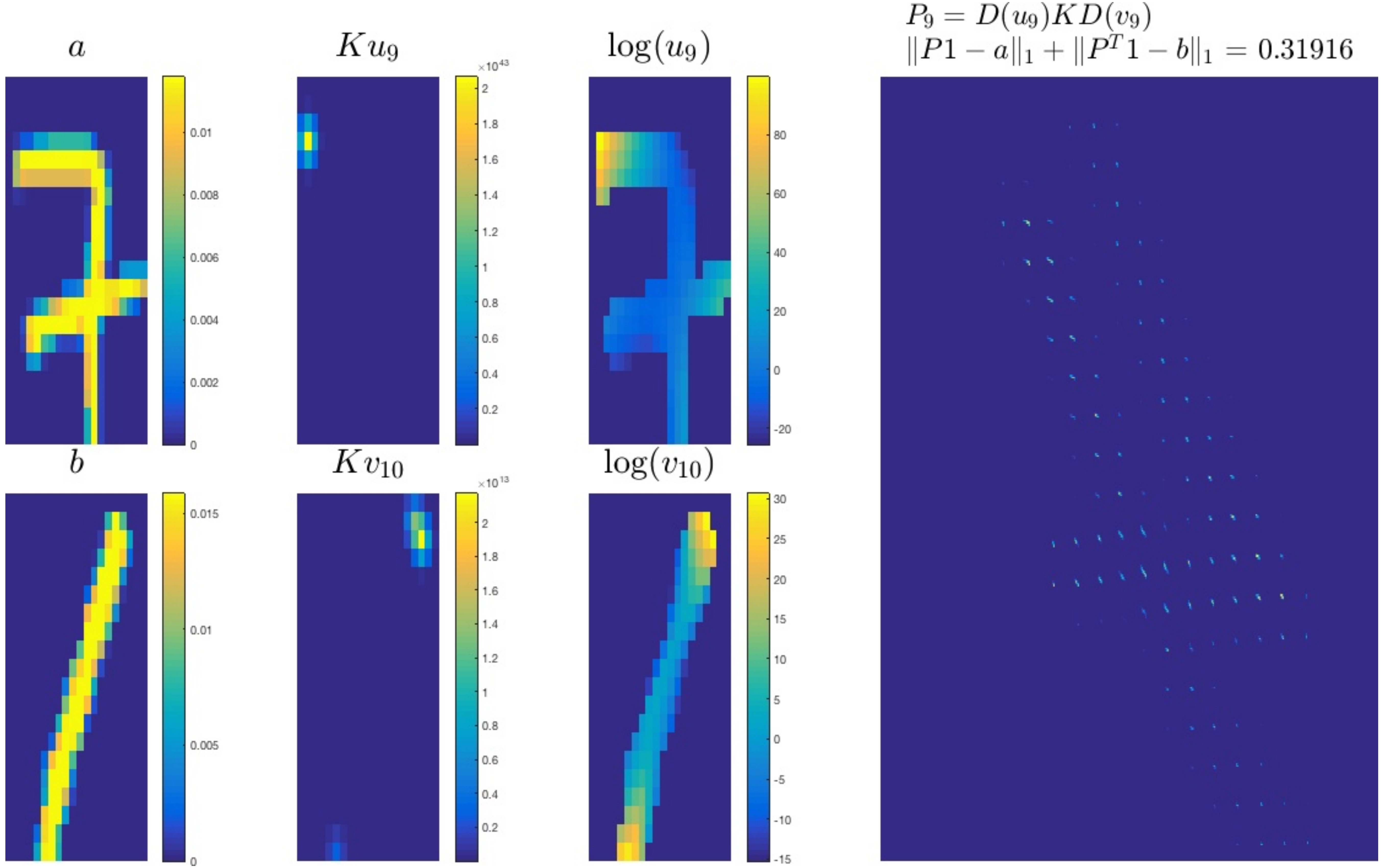
Very Fast EMD Approx. Solver



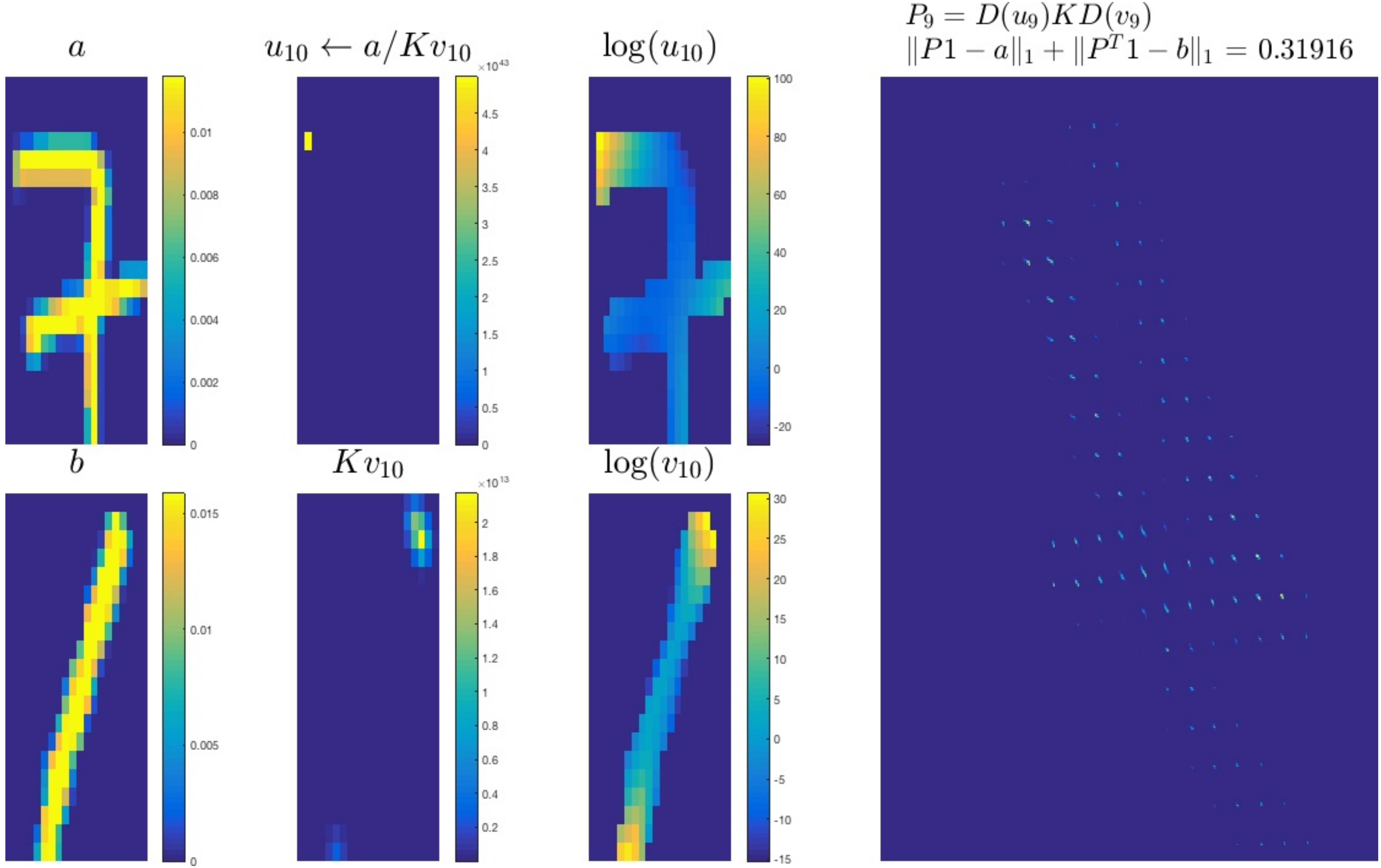
Very Fast EMD Approx. Solver



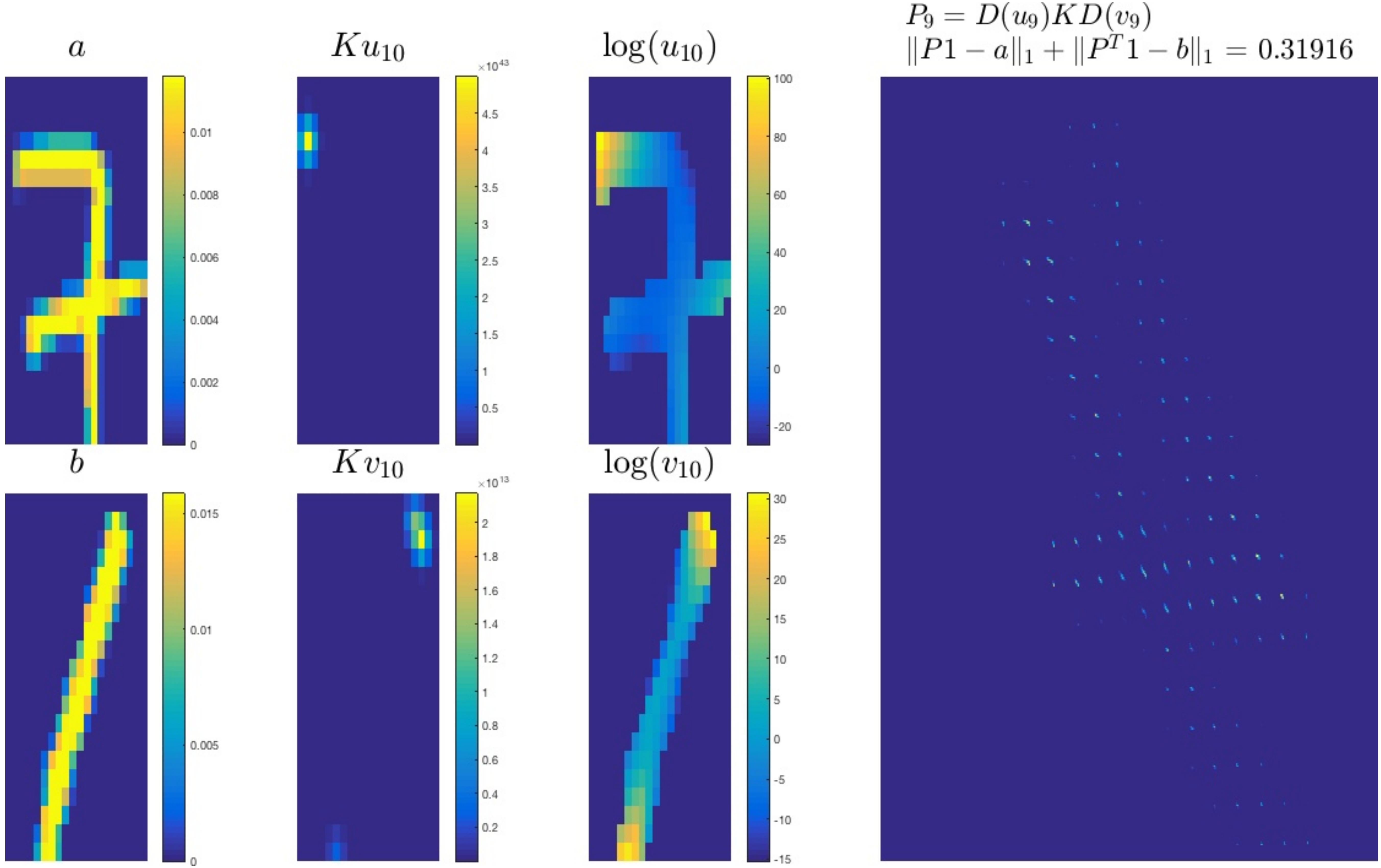
Very Fast EMD Approx. Solver



Very Fast EMD Approx. Solver



Very Fast EMD Approx. Solver



Very Fast EMD Approx. Solver