Tensor trains for sequential state and parameter estimation in state-space models

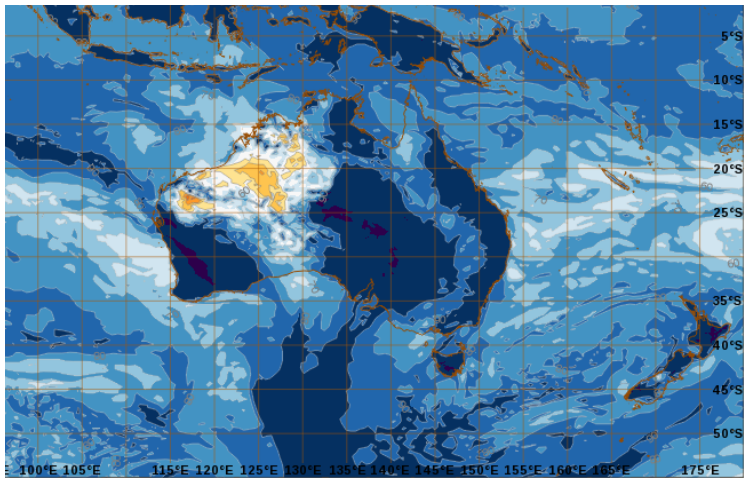
In collaboration with Yiran Zhao

Tiangang Cui
University of Sydney

tiangang.cui@sydney.edu.au
https://www.fastfins.org/

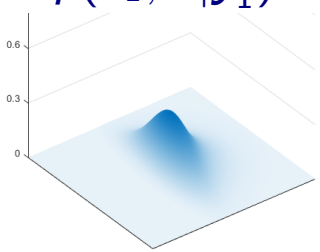
Institute for Mathematical and Statistical Innovation
October 6, 2025

Sequential learning

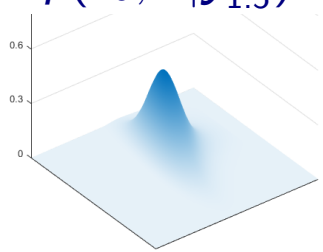


- Aim to **sequentially** estimate hidden states $\{\boldsymbol{x}_t\}_{t=0}^{T}$ and parameters $\boldsymbol{\theta}$ of stochastic dynamical systems from indirect data $\{\boldsymbol{y}_t\}_{t=1}^{T}$
- Need to **recursively** learn posterior random variables

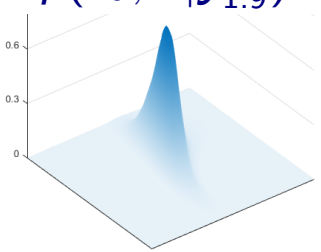
$p(\boldsymbol{x}_1, \boldsymbol{\theta}|\boldsymbol{y}_1)$



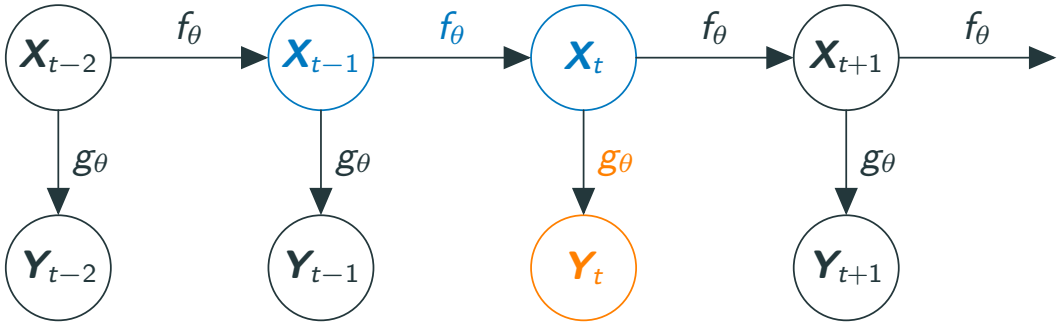
$p(\boldsymbol{x}_5, \boldsymbol{\theta}|\boldsymbol{y}_{1:5})$



$p(\boldsymbol{x}_9, \boldsymbol{\theta}|\boldsymbol{y}_{1:9})$



State-space models



- State evolution and observation follow the **hidden Markov property**

  State transition $\quad \boldsymbol{X}_t | \left(\boldsymbol{X}_{t-1} = \boldsymbol{x}\right) \sim f(\cdot|\boldsymbol{x}, \boldsymbol{\theta})$

  Likelihood function $\quad \boldsymbol{Y}_t | \left(\boldsymbol{X}_t = \boldsymbol{x}\right) \sim g(\cdot|\boldsymbol{x}, \boldsymbol{\theta})$

- **Filtering** estimates the current state using observations up to time t

$$p(\boldsymbol{x}_t, \boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \propto \int p(\boldsymbol{x}_{0:t}, \boldsymbol{\theta}|\boldsymbol{y}_{1:t}) d\boldsymbol{x}_{0:t\text{-}1}$$

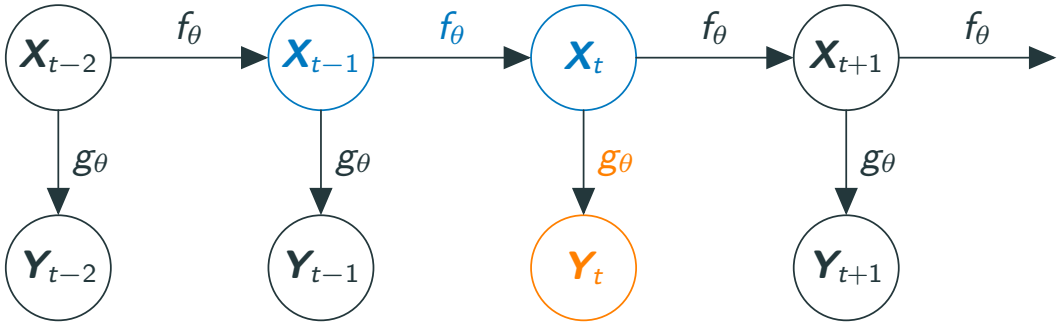
- **Smoothing** estimates the state using future observations

$$p(\boldsymbol{x}_t, \boldsymbol{\theta}|\boldsymbol{y}_{1:T}) \propto \int p(\boldsymbol{x}_{1:T}, \boldsymbol{\theta}|\boldsymbol{y}_{0:T}) d\boldsymbol{x}_{0:t\text{-}1} d\boldsymbol{x}_{t+1:T}$$

- **Parameter estimation:**

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \propto \int p(\boldsymbol{x}_{0:t}, \boldsymbol{\theta}|\boldsymbol{y}_{1:t}) d\boldsymbol{x}_{0:t}$$

State-space models



- State evolution and observation follow the **hidden Markov property**

  State transition $\quad \boldsymbol{X}_t | \left(\boldsymbol{X}_{t-1} = \boldsymbol{x}\right) \sim f(\cdot|\boldsymbol{x}, \boldsymbol{\theta})$

  Likelihood function $\quad \boldsymbol{Y}_t | \left(\boldsymbol{X}_t = \boldsymbol{x}\right) \sim g(\cdot|\boldsymbol{x}, \boldsymbol{\theta})$

- **Filtering** estimates the current state using observations up to time t

$$p(\boldsymbol{x}_t, \boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \propto \int p(\boldsymbol{x}_{0:t}, \boldsymbol{\theta}|\boldsymbol{y}_{1:t}) d\boldsymbol{x}_{0:t\text{-}1}$$

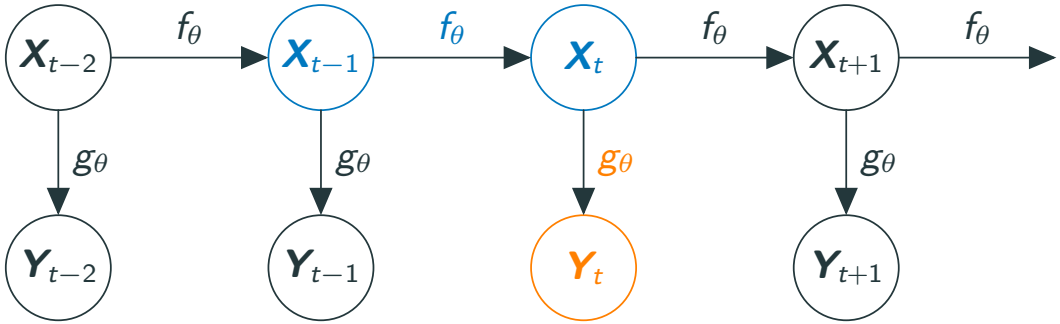
- **Smoothing** estimates the state using future observations

$$p(\boldsymbol{x}_t, \boldsymbol{\theta}|\boldsymbol{y}_{1:T}) \propto \int p(\boldsymbol{x}_{1:T}, \boldsymbol{\theta}|\boldsymbol{y}_{0:T}) d\boldsymbol{x}_{0:t\text{-}1} d\boldsymbol{x}_{t+1:T}$$

- **Parameter estimation:**

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \propto \int p(\boldsymbol{x}_{0:t}, \boldsymbol{\theta}|\boldsymbol{y}_{1:t}) d\boldsymbol{x}_{0:t}$$

State-space models

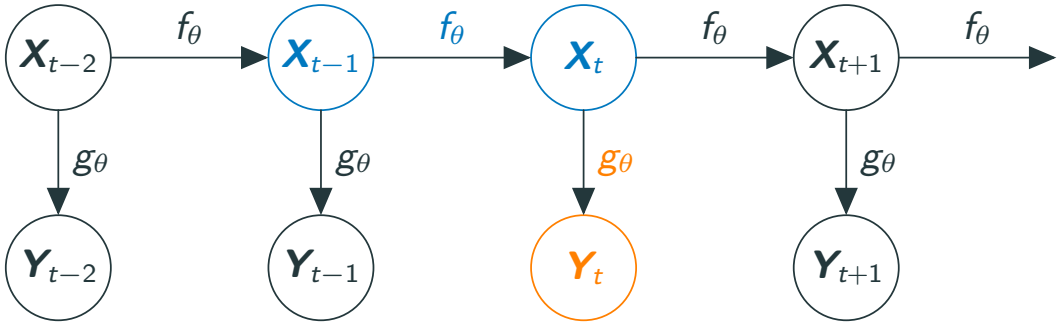


- The joint posterior density is recursively defined:

$$p(x_t, x_{t\text{-}1}, \theta | y_{1:t}) \propto p(x_{t\text{-}1}, \theta | y_{1:t\text{-}1}) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta)$$

- Existing methods:
  - **Without parameter:** particle filters [Del Moral, Doucet, Liu, ...], TT-Zakai [Li *et al.* 2022], Frobenius–Perron [Fox *et al.* 2021], (optimal) transport maps [Gottwald, Reich, Baptista, ...]
  - **With parameter:** SMC^2 and nested filters (filters within filters) [Chopin *et al.* 2013; Crisan *et al.* 2018; etc.], Knothe–Rosenblatt rearrangement [Spantini *et al.* 2018]
- **We propose one method for all sequential estimation problems:**
  1. Recursive posterior approximation and marginalization using tensor trains
  2. Filtering and smoothing via accompanying Knothe–Rosenblatt rearrangement
  3. Examples

State-space models



- The joint posterior density is recursively defined:

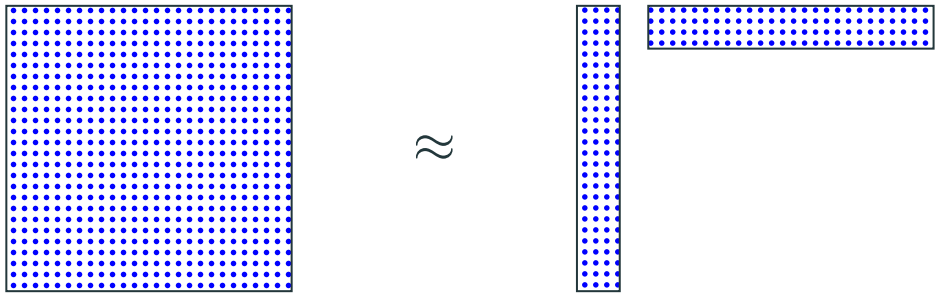
$$p(x_t, x_{t\text{-}1}, \theta | y_{1:t}) \propto p(x_{t\text{-}1}, \theta | y_{1:t\text{-}1}) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta)$$

- Existing methods:
  - **Without parameter:** particle filters [Del Moral, Doucet, Liu, ...], TT-Zakai [Li *et al.* 2022], Frobenius–Perron [Fox *et al.* 2021], (optimal) transport maps [Gottwald, Reich, Baptista, ...]
  - **With parameter:** SMC^2 and nested filters (filters within filters) [Chopin *et al.* 2013; Crisan *et al.* 2018; etc.], Knothe–Rosenblatt rearrangement [Spantini *et al.* 2018]
- **We propose one method for all sequential estimation problems:**
  1. Recursive posterior approximation and marginalization using tensor trains
  2. Filtering and smoothing via accompanying Knothe–Rosenblatt rearrangement
  3. Examples

Recursive approximation

Functional tensor train: what happened when $d = 2$

- 2D: discretisation of $\pi(x_1, x_2)$ gives a matrix $A(i, j)$



- Function/matrix decomposition

$$\pi(x_1, x_2) = \sum_{\alpha_1=1}^{r} \varphi_1(x_1, \alpha_1)\varphi_2(\alpha_1, x_2) + \mathcal{O}(\varepsilon)$$

$$A(i, j) = \sum_{\alpha_1=1}^{r} V(i, \alpha_1)W(\alpha_1, j) + \mathcal{O}(\varepsilon)$$

- *Rank* $r \ll n$ to save storage and computing cost. Separates x_1 and x_2

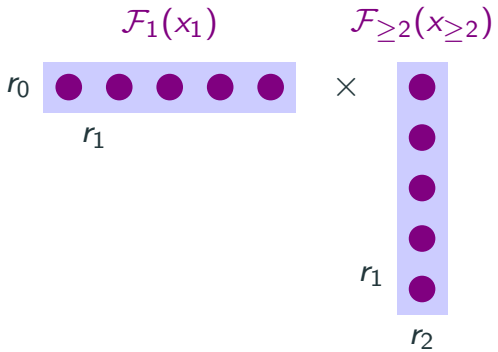
$$\pi(x_2) = \int \pi(x_1, x_2)dx_1 \approx \sum_{\alpha_1=1}^{r} \left(\int \varphi_1(x_1, \alpha_1)dx_1 \right) \varphi_2(\alpha_1, x_2)$$

$$\pi(x_1) = \int \pi(x_1, x_2)dx_2 \approx \sum_{\alpha_1=1}^{r} \varphi_1(x_1, \alpha_1) \left(\int \varphi_2(\alpha_1, x_2)dx_2 \right)$$

How to build a functional tensor train $d > 2$ (on pen and paper)?

- Group variables and apply, e.g., SVD, we obtain

$$\pi(x_1, \underbrace{x_2, \ldots, x_d}_{x_{\geq 2}}) = \pi(x_1, x_{\geq 2}) \approx \sum_{\alpha_1=1}^{r_1} \varphi_1(x_1, \alpha_1)\varphi_{\geq 2}(\alpha_1, x_{\geq 2})$$



$$\mathcal{F}_1(x_1) = [\varphi_1(x_1, 1), \ldots, \varphi_1(x_1, r_1)]$$

$$\mathcal{F}_{\geq 2}(x_{\geq 2}) = [\varphi_{\geq 2}(1, x_{\geq 2}), \ldots, \varphi_{\geq 2}(r_1, x_{\geq 2})]^\top$$

- Decompose $\mathcal{F}_{\geq 2}(x_{\geq 2})$:

$$\varphi_{\geq 2}(\underbrace{\alpha_1, x_2}_{\tilde{x}_{\leq 2}}, x_{\geq 3}) = \varphi_{\geq 2}(\tilde{x}_{\leq 2}, x_{\geq 3}) \approx \sum_{\alpha_2=1}^{r_2} \varphi_2(\alpha_1, x_2, \alpha_2)\varphi_{\geq 3}(\alpha_2, x_{\geq 3})$$

$$\mathcal{F}_2(x_2) = \begin{bmatrix} \varphi_2(1, x_2, 1) & \cdots & \varphi_2(1, x_2, r_2) \\ \vdots & \ddots & \vdots \\ \varphi_2(r_1, x_2, 1) & \cdots & \varphi_2(r_1, x_2, r_2) \end{bmatrix}$$

$$\mathcal{F}_{\geq 3}(x_{\geq 3}) = [\varphi_{\geq 3}(1, x_{\geq 3}), \ldots, \varphi_{\geq 3}(r_2, x_{\geq 3})]^\top$$

- Decompose $\mathcal{F}_{\geq 3}(x_{\geq 3})$ and cont ...

How to build a functional tensor train $d > 2$ (on pen and paper)?

- Group variables and apply, e.g., SVD, we obtain

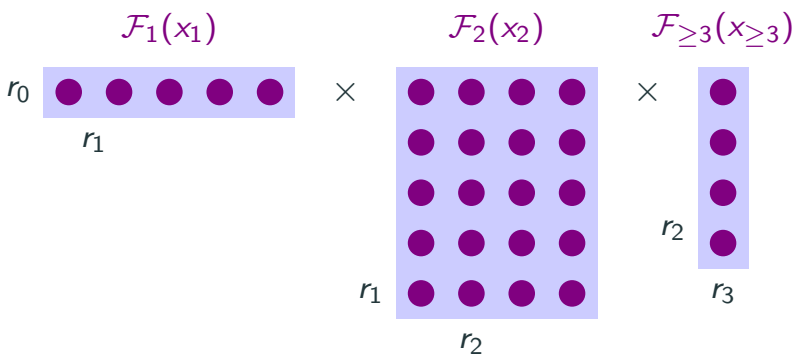
$$\pi(x_1, \underbrace{x_2, \ldots, x_d}_{x_{\geq 2}}) = \pi(x_1, x_{\geq 2}) \approx \sum_{\alpha_1=1}^{r_1} \varphi_1(x_1, \alpha_1)\varphi_{\geq 2}(\alpha_1, x_{\geq 2})$$

$$\mathcal{F}_1(x_1) = [\varphi_1(x_1, 1), \ldots, \varphi_1(x_1, r_1)]$$

$$\mathcal{F}_{\geq 2}(x_{\geq 2}) = [\varphi_{\geq 2}(1, x_{\geq 2}), \ldots, \varphi_{\geq 2}(r_1, x_{\geq 2})]^\top$$

- Decompose $\mathcal{F}_{\geq 2}(x_{\geq 2})$:

$$\varphi_{\geq 2}(\underbrace{\alpha_1, x_2}_{\tilde{x}_{\leq 2}}, x_{\geq 3}) = \varphi_{\geq 2}(\tilde{x}_{\leq 2}, x_{\geq 3}) \approx \sum_{\alpha_2=1}^{r_2} \varphi_2(\alpha_1, x_2, \alpha_2)\varphi_{\geq 3}(\alpha_2, x_{\geq 3})$$



$$\mathcal{F}_2(x_2) = \begin{bmatrix} \varphi_2(1, x_2, 1) & \cdots & \varphi_2(1, x_2, r_2) \\ \vdots & \ddots & \vdots \\ \varphi_2(r_1, x_2, 1) & \cdots & \varphi_2(r_1, x_2, r_2) \end{bmatrix}$$

$$\mathcal{F}_{\geq 3}(x_{\geq 3}) = [\varphi_{\geq 3}(1, x_{\geq 3}), \ldots, \varphi_{\geq 3}(r_2, x_{\geq 3})]^\top$$

- Decompose $\mathcal{F}_{\geq 3}(x_{\geq 3})$ and cont ...

How to build a functional tensor train $d > 2$ (on pen and paper)?

- Group variables and apply, e.g., SVD, we obtain

$$\pi(x_1, \underbrace{x_2, \ldots, x_d}_{x_{\geq 2}}) = \pi(x_1, x_{\geq 2}) \approx \sum_{\alpha_1=1}^{r_1} \varphi_1(x_1, \alpha_1)\varphi_{\geq 2}(\alpha_1, x_{\geq 2})$$

$$\mathcal{F}_1(x_1) = [\varphi_1(x_1, 1), \ldots, \varphi_1(x_1, r_1)]$$

$$\mathcal{F}_{\geq 2}(x_{\geq 2}) = [\varphi_{\geq 2}(1, x_{\geq 2}), \ldots, \varphi_{\geq 2}(r_1, x_{\geq 2})]^\top$$

- Decompose $\mathcal{F}_{\geq 2}(x_{\geq 2})$:

$$\varphi_{\geq 2}(\underbrace{\alpha_1, x_2}_{\tilde{x}_{\leq 2}}, x_{\geq 3}) = \varphi_{\geq 2}(\tilde{x}_{\leq 2}, x_{\geq 3}) \approx \sum_{\alpha_2=1}^{r_2} \varphi_2(\alpha_1, x_2, \alpha_2)\varphi_{\geq 3}(\alpha_2, x_{\geq 3})$$

$$\mathcal{F}_2(x_2) = \begin{bmatrix} \varphi_2(1, x_2, 1) & \cdots & \varphi_2(1, x_2, r_2) \\ \vdots & \ddots & \vdots \\ \varphi_2(r_1, x_2, 1) & \cdots & \varphi_2(r_1, x_2, r_2) \end{bmatrix}$$

$$\mathcal{F}_{\geq 3}(x_{\geq 3}) = [\varphi_{\geq 3}(1, x_{\geq 3}), \ldots, \varphi_{\geq 3}(r_2, x_{\geq 3})]^\top$$

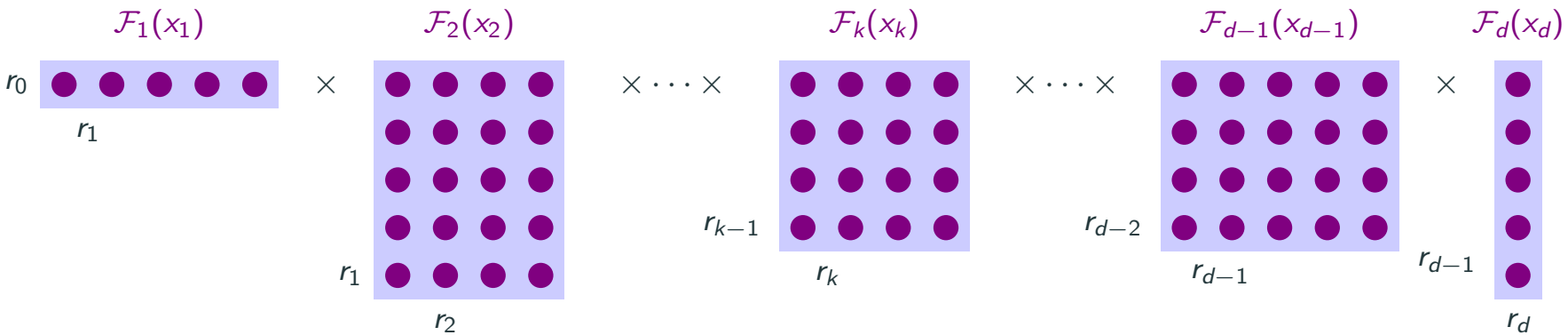
- Decompose $\mathcal{F}_{\geq 3}(x_{\geq 3})$ and cont ...

Functional tensor train: $d > 2$

TT factorization $\implies$ separation of variable

$$\pi(x_1, \ldots, x_d) \approx \mathcal{F}_1(x_1) \cdots \mathcal{F}_k(x_k) \cdots \mathcal{F}_d(x_d)$$

Each $\mathcal{F}_k : \mathbb{R} \to \mathbb{R}^{r_{k-1} \times r_k}$ is a *matrix-valued univariate* function



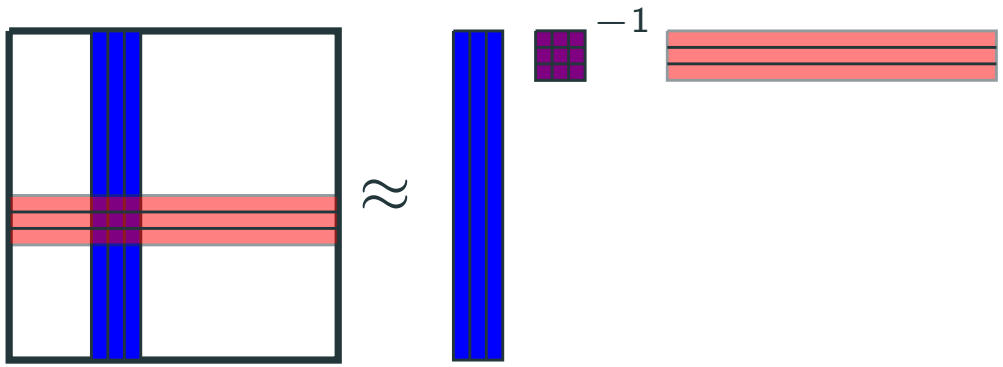
Integration of a matrix-valued function $\mathcal{F}_k(x_k) \implies$ integration over x_k

Cross interpolation: how we build it in practice

- How to build this? SVD/Schmidt decomposition (unpractical for large d)
  needs high-dimensional integration
- A rank-r matrix can be recovered by its r indep. rows and cols:

$$A = \hat{A} \equiv A(:, \mathcal{J}) A^{-1}(\mathcal{I}, \mathcal{J}) A(\mathcal{I}, :),$$

for some *index sets* $\mathcal{I}, \mathcal{J} \subset \{1, \ldots, n\}$ of cardinality r.



Tricks are in details:

- what are the *best* and *practically realizable* indices $\mathcal{I}, \mathcal{J}$?
- how to generalize to $d > 2$ dimensions?

Tensor train: MaxVol and alternating iteration

Maximum Volume (MaxVol): *best* indices in Chebyshev norm:

- If $\left|\det A(\mathcal{I}, \mathcal{J})\right| = \max_{\hat{\mathcal{I}}, \hat{\mathcal{J}}} \left|\det A(\hat{\mathcal{I}}, \hat{\mathcal{J}})\right|$, then $\|A - \hat{A}\|_\infty \leq (r+1)\sigma_{r+1}(A)$. *
- NP-hard to look through all submatrices, so apply alternating iterations

Given an *initial set* $\mathcal{J} \subset \{1 \dots n\}$, repeat the following

1. $V = A(:, \mathcal{J})$
2. $\mathcal{I} = \texttt{pivots}(V)$
3. $W = A(\mathcal{I}, :)$.
4. $\mathcal{J} = \texttt{pivots}(W)$.

`pivots` via LU or QR with $\mathcal{O}(nr^2)$ flops [Goreinov & Tyrtyshnikov 2001]. Easily generalisable to $d > 2$.

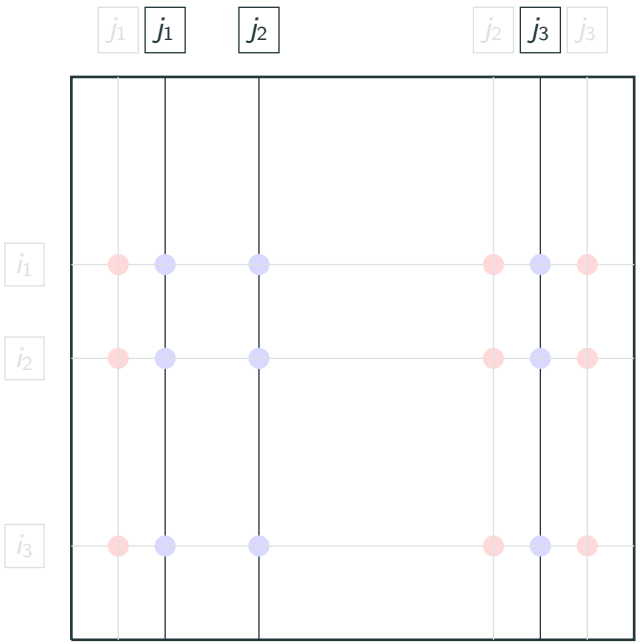
*Frank de Hoog and Markus Hegland further optimised this bound [de Hoog & Hegland 2023]

Tensor train: MaxVol and alternating iteration

Maximum Volume (MaxVol): *best* indices in Chebyshev norm:

- If $\left|\det A(\mathcal{I}, \mathcal{J})\right| = \max_{\hat{\mathcal{I}}, \hat{\mathcal{J}}} \left|\det A(\hat{\mathcal{I}}, \hat{\mathcal{J}})\right|$, then $\|A - \hat{A}\|_\infty \leq (r+1)\sigma_{r+1}(A)$. *
- NP-hard to look through all submatrices, so apply alternating iterations

Given an *initial set* $\mathcal{J} \subset \{1 \dots n\}$, repeat the following



1. $V = A(:, \mathcal{J})$
2. $\mathcal{I} = \texttt{pivots}(V)$
3. $W = A(\mathcal{I}, :)$.
4. $\mathcal{J} = \texttt{pivots}(W)$.

`pivots` via LU or QR with $\mathcal{O}(nr^2)$ flops [Goreinov & Tyrtyshnikov 2001]. Easily generalisable to $d > 2$.

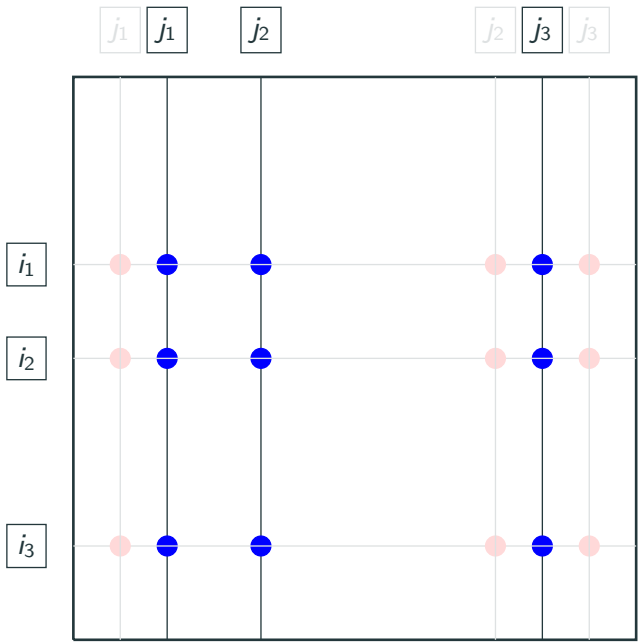
*Frank de Hoog and Markus Hegland further optimised this bound [de Hoog & Hegland 2023]

Tensor train: MaxVol and alternating iteration

Maximum Volume (MaxVol): *best* indices in Chebyshev norm:

- If $\left|\det A(\mathcal{I}, \mathcal{J})\right| = \max_{\hat{\mathcal{I}}, \hat{\mathcal{J}}} \left|\det A(\hat{\mathcal{I}}, \hat{\mathcal{J}})\right|$, then $\|A - \hat{A}\|_\infty \leq (r+1)\sigma_{r+1}(A)$. *
- NP-hard to look through all submatrices, so apply alternating iterations

Given an *initial set* $\mathcal{J} \subset \{1 \dots n\}$, repeat the following



1. $V = A(:, \mathcal{J})$
2. $\mathcal{I} = \texttt{pivots}(V)$
3. $W = A(\mathcal{I}, :)$.
4. $\mathcal{J} = \texttt{pivots}(W)$.

`pivots` via LU or QR with $\mathcal{O}(nr^2)$ flops [Goreinov & Tyrtyshnikov 2001]. Easily generalisable to $d > 2$.

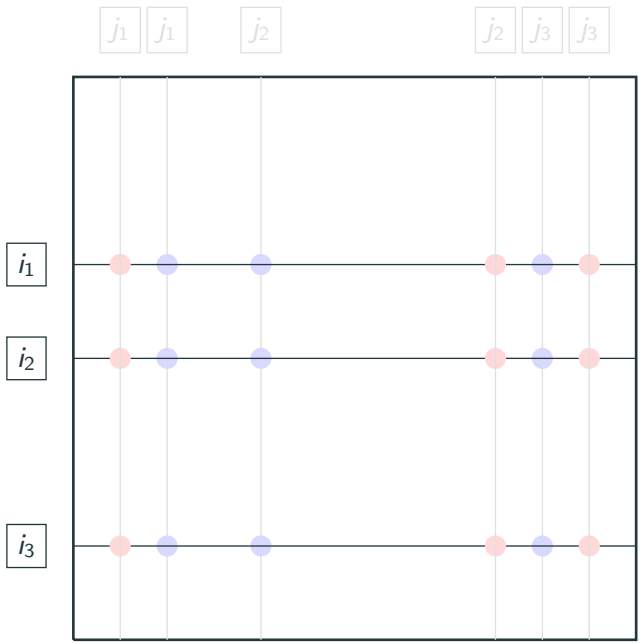
*Frank de Hoog and Markus Hegland further optimised this bound [de Hoog & Hegland 2023]

Tensor train: MaxVol and alternating iteration

Maximum Volume (MaxVol): *best* indices in Chebyshev norm:

- If $\left|\det A(\mathcal{I}, \mathcal{J})\right| = \max_{\hat{\mathcal{I}}, \hat{\mathcal{J}}} \left|\det A(\hat{\mathcal{I}}, \hat{\mathcal{J}})\right|$, then $\|A - \hat{A}\|_\infty \leq (r+1)\sigma_{r+1}(A)$. *
- NP-hard to look through all submatrices, so apply alternating iterations

Given an *initial set* $\mathcal{J} \subset \{1 \dots n\}$, repeat the following



1. $V = A(:, \mathcal{J})$
2. $\mathcal{I} = \texttt{pivots}(V)$
3. $W = A(\mathcal{I}, :)$.
4. $\mathcal{J} = \texttt{pivots}(W)$.

`pivots` via LU or QR with $\mathcal{O}(nr^2)$ flops [Goreinov & Tyrtyshnikov 2001]. Easily generalisable to $d > 2$.

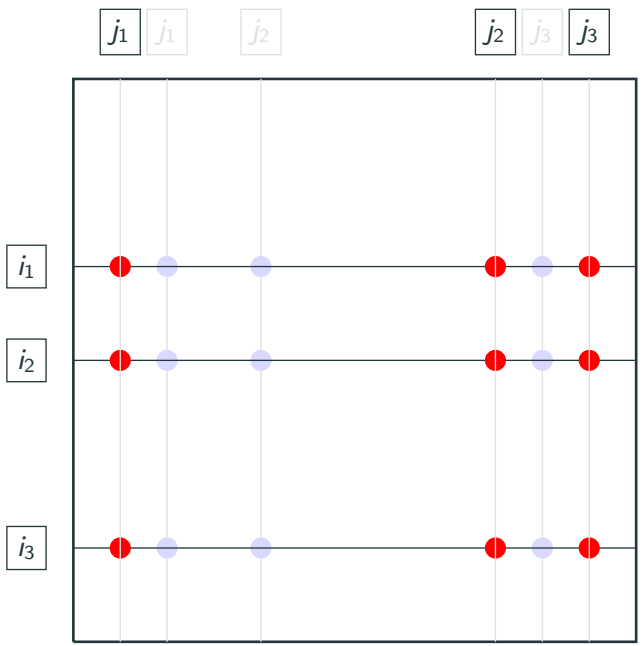
*Frank de Hoog and Markus Hegland further optimised this bound [de Hoog & Hegland 2023]

Tensor train: MaxVol and alternating iteration

Maximum Volume (MaxVol): *best* indices in Chebyshev norm:

- If $\left|\det A(\mathcal{I}, \mathcal{J})\right| = \max_{\hat{\mathcal{I}}, \hat{\mathcal{J}}} \left|\det A(\hat{\mathcal{I}}, \hat{\mathcal{J}})\right|$, then $\|A - \hat{A}\|_\infty \leq (r+1)\sigma_{r+1}(A)$. *
- NP-hard to look through all submatrices, so apply alternating iterations

Given an *initial set* $\mathcal{J} \subset \{1 \dots n\}$, repeat the following



1. $V = A(:, \mathcal{J})$
2. $\mathcal{I} = \texttt{pivots}(V)$
3. $W = A(\mathcal{I}, :)$.
4. $\mathcal{J} = \texttt{pivots}(W)$.

`pivots` via LU or QR with $\mathcal{O}(nr^2)$ flops [Goreinov & Tyrtyshnikov 2001]. Easily generalisable to $d > 2$.

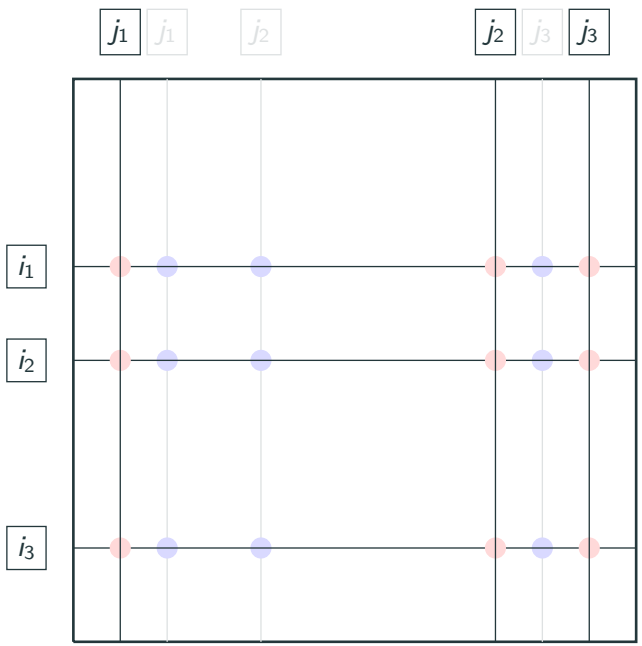
*Frank de Hoog and Markus Hegland further optimised this bound [de Hoog & Hegland 2023]

Tensor train: MaxVol and alternating iteration

Maximum Volume (MaxVol): *best* indices in Chebyshev norm:

- If $\left|\det A(\mathcal{I}, \mathcal{J})\right| = \max_{\hat{\mathcal{I}}, \hat{\mathcal{J}}} \left|\det A(\hat{\mathcal{I}}, \hat{\mathcal{J}})\right|$, then $\|A - \hat{A}\|_\infty \leq (r+1)\sigma_{r+1}(A)$. *
- NP-hard to look through all submatrices, so apply alternating iterations

Given an *initial set* $\mathcal{J} \subset \{1 \dots n\}$, repeat the following



1. $V = A(:, \mathcal{J})$
2. $\mathcal{I} = \texttt{pivots}(V)$
3. $W = A(\mathcal{I}, :)$.
4. $\mathcal{J} = \texttt{pivots}(W)$.

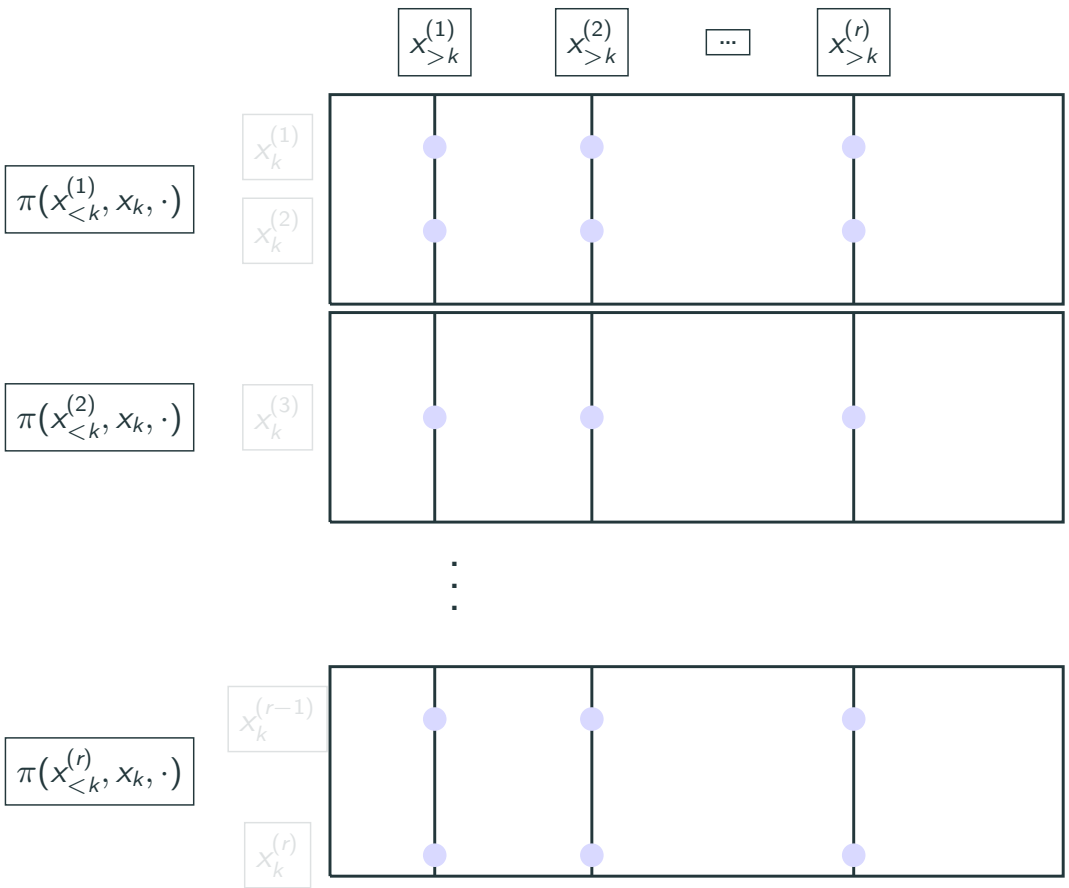
`pivots` via LU or QR with $\mathcal{O}(nr^2)$ flops [Goreinov & Tyrtyshnikov 2001]. Easily generalisable to $d > 2$.

*Frank de Hoog and Markus Hegland further optimised this bound [de Hoog & Hegland 2023]

Functional TT cross

Grouped coordinates: $x_{<k} = (x_1, x_2, \ldots, x_{k-1})$, $x_{>k} = (x_{k+1}, \ldots, x_{d-1}, x_d)$

Left and right interpolation sets: $\mathcal{I}_{<k} = \{x^{(i)}_{<k}\}_{i=1}^{r}$, $\mathcal{J}_{>k} = \{x^{(j)}_{>k}\}_{j=1}^{r}$



Iterate to next sets:

- Discretize x_k: $\pi(\mathcal{I}_{<k}, x_k, \mathcal{J}_{>k})$ for all $x_k \in \mathcal{I}_k$
- $\mathcal{I}_{<k+1} = \texttt{MaxVol} \subset \mathcal{I}_{<k} \otimes \mathcal{I}_k$
- $\mathcal{J}_{>k+1} = \{x^{(j)}_{>k+1}\}_{j=1}^{r}$
- $k \to k+1$, move to the x_{k+1}.
- $k = d$, switch direction and update $\mathcal{J}_{>k}$.
- Stop if converged, repeat otherwise.
- $O(dr^2)$ function evaluations.

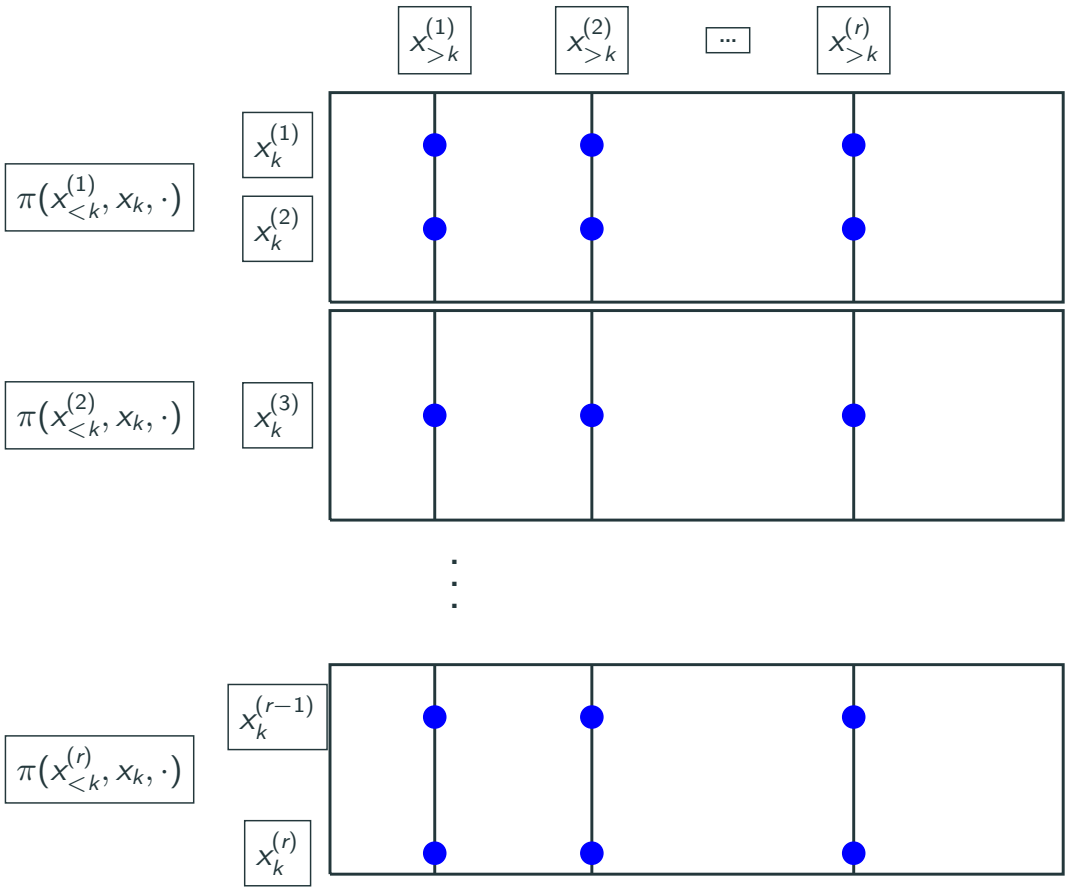
TT cross: [Oseledets & Tyrtyshnikov 2010], [Gorodetsky, Karaman & Marzouk 2018]

Empirical interpolation: [Maday, Chaturantabut, Sorensen ...]

Functional TT cross

Grouped coordinates: $x_{<k} = (x_1, x_2, \ldots, x_{k-1})$, $x_{>k} = (x_{k+1}, \ldots, x_{d-1}, x_d)$

Left and right interpolation sets: $\mathcal{I}_{<k} = \{x^{(i)}_{<k}\}_{i=1}^{r}$, $\mathcal{J}_{>k} = \{x^{(j)}_{>k}\}_{j=1}^{r}$



Iterate to next sets:

- Discretize x_k: $\pi(\mathcal{I}_{<k}, \underline{x_k}, \mathcal{J}_{>k})$ for all $x_k \in \mathcal{I}_k$
- $\mathcal{I}_{<k+1} = \texttt{MaxVol} \subset \mathcal{I}_{<k} \otimes \mathcal{I}_k$
- $\mathcal{J}_{>k+1} = \{x^{(j)}_{>k+1}\}_{j=1}^{r}$
- $k \to k+1$, move to the x_{k+1}.
- $k = d$, switch direction and update $\mathcal{J}_{>k}$.
- Stop if converged, repeat otherwise.
- $O(dr^2)$ function evaluations.

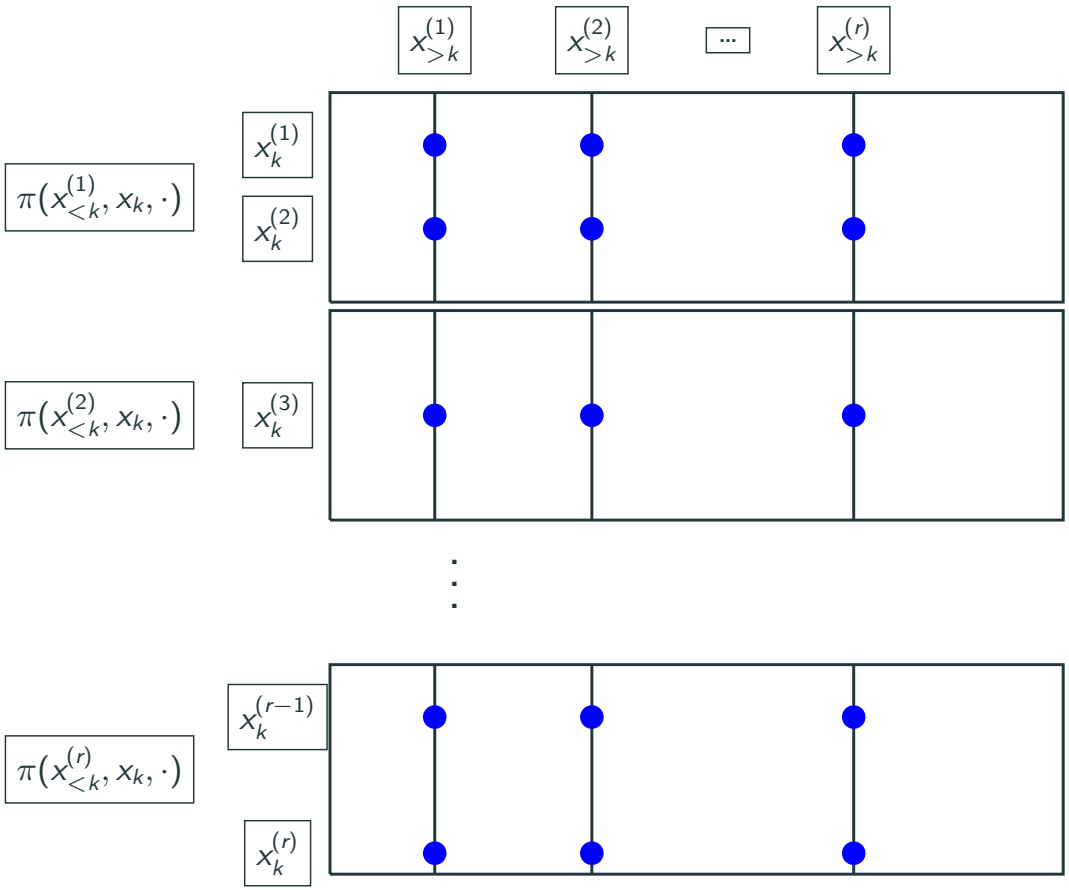
TT cross: [Oseledets & Tyrtyshnikov 2010], [Gorodetsky, Karaman & Marzouk 2018]

Empirical interpolation: [Maday, Chaturantabut, Sorensen ...]

Functional TT cross

Grouped coordinates: $x_{<k} = (x_1, x_2, \ldots, x_{k-1})$, $x_{>k} = (x_{k+1}, \ldots, x_{d-1}, x_d)$

Left and right interpolation sets: $\mathcal{I}_{<k} = \{x^{(i)}_{<k}\}_{i=1}^{r}$, $\mathcal{J}_{>k} = \{x^{(j)}_{>k}\}_{j=1}^{r}$



Iterate to next sets:

- Discretize x_k: $\pi(\mathcal{I}_{<k}, \underline{x_k}, \mathcal{J}_{>k})$ for all $x_k \in \mathcal{I}_k$
- $\mathcal{I}_{<k+1} = \texttt{MaxVol} \subset \mathcal{I}_{<k} \otimes \mathcal{I}_k$
- $\mathcal{J}_{>k+1} = \{x^{(j)}_{>k+1}\}_{j=1}^{r}$
- $k \to k+1$, move to the x_{k+1}.
- $k = d$, switch direction and update $\mathcal{J}_{>k}$.
- Stop if converged, repeat otherwise.
- $O(dr^2)$ function evaluations.

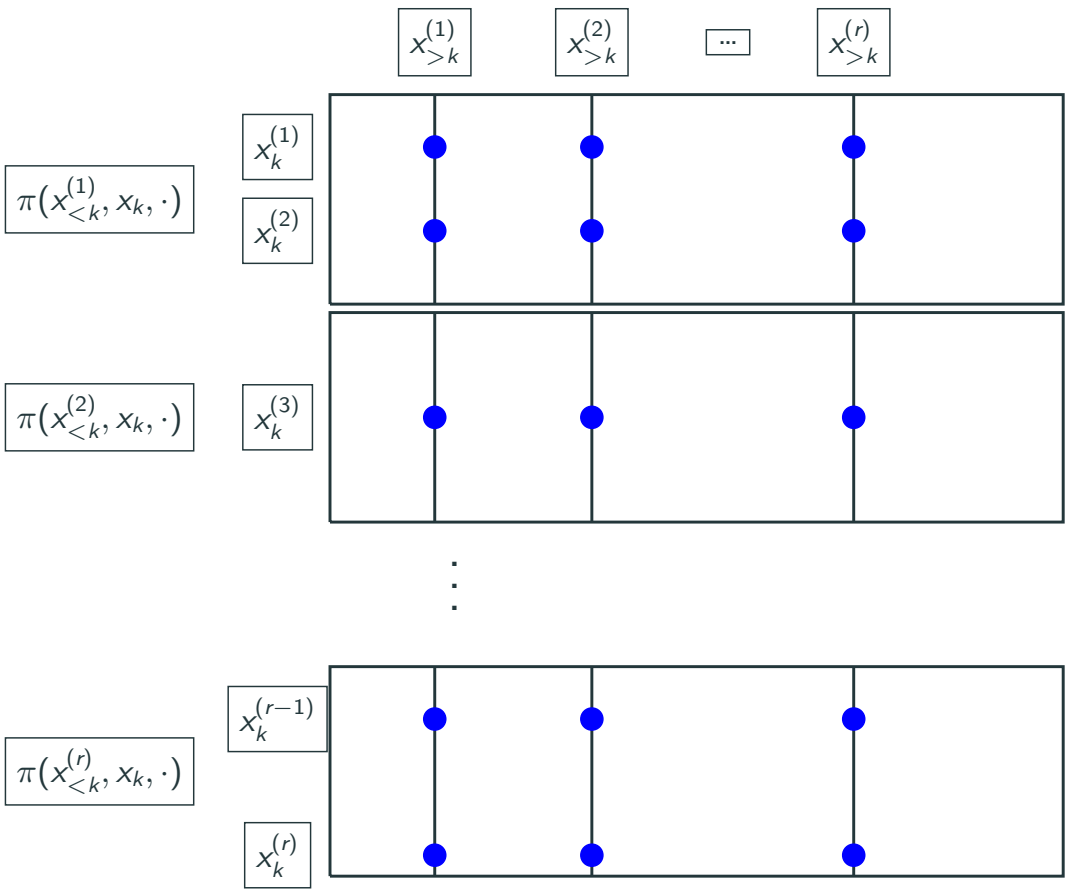
TT cross: [Oseledets & Tyrtyshnikov 2010], [Gorodetsky, Karaman & Marzouk 2018]

Empirical interpolation: [Maday, Chaturantabut, Sorensen ...]

Functional TT cross

Grouped coordinates: $x_{<k} = (x_1, x_2, \ldots, x_{k-1})$, $x_{>k} = (x_{k+1}, \ldots, x_{d-1}, x_d)$

Left and right interpolation sets: $\mathcal{I}_{<k} = \{x^{(i)}_{<k}\}_{i=1}^{r}$, $\mathcal{J}_{>k} = \{x^{(j)}_{>k}\}_{j=1}^{r}$



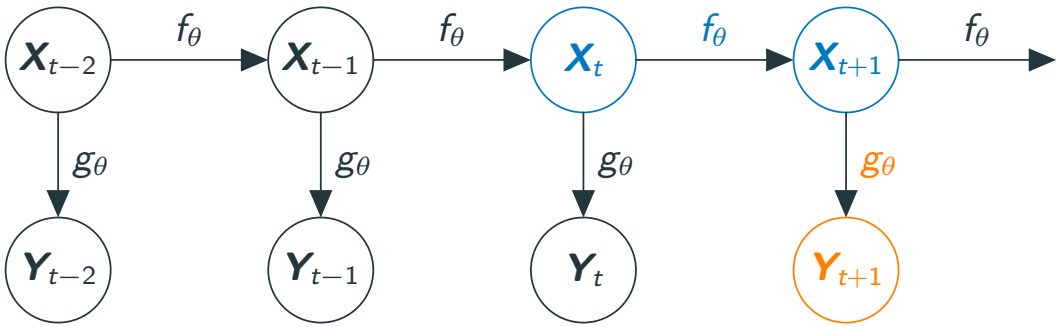
Iterate to next sets:

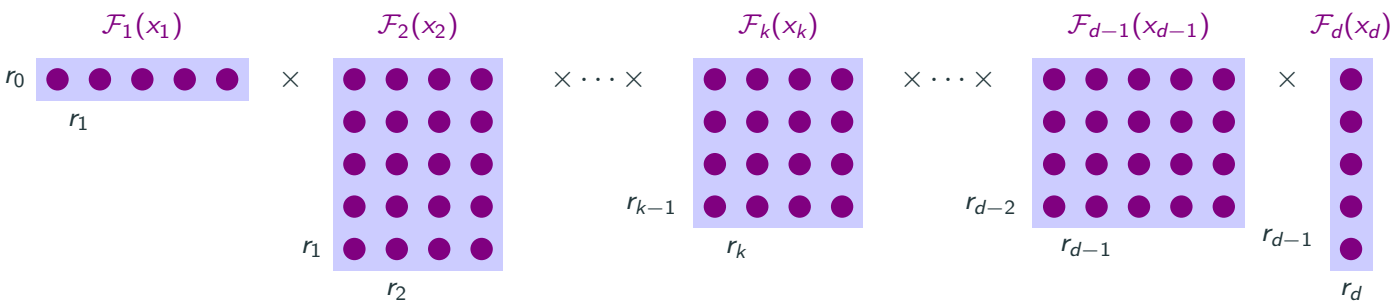
- Discretize x_k: $\pi(\mathcal{I}_{<k}, \underline{x_k}, \mathcal{J}_{>k})$ for all $x_k \in \mathcal{I}_k$
- $\mathcal{I}_{<k+1} = \texttt{MaxVol} \subset \mathcal{I}_{<k} \otimes \mathcal{I}_k$
- $\mathcal{J}_{>k+1} = \{x^{(j)}_{>k+1}\}_{j=1}^{r}$
- $k \to k+1$, move to the x_{k+1}.
- $k = d$, switch direction and update $\mathcal{J}_{>k}$.
- Stop if converged, repeat otherwise.
- $O(dr^2)$ function evaluations.

TT cross: [Oseledets & Tyrtyshnikov 2010], [Gorodetsky, Karaman & Marzouk 2018]

Empirical interpolation: [Maday, Chaturantabut, Sorensen ...]

Recursive approximation and marginalization





Recursive approximation and marginalization

- Previous filtering density:

$$p_{t\text{-}1} := p(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta} | \boldsymbol{y}_{1:t\text{-}1})$$

$$p_{t\text{-}1}(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) \approx \tilde{p}_{t\text{-}1}(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta})$$

- Propagate via the joint:

$$p(\boldsymbol{x}_t, \boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta} | \boldsymbol{y}_{1:t}) \propto p_{t\text{-}1}(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) f(\boldsymbol{x}_t | \boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) g(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta})$$

$$p(\boldsymbol{x}_t, \boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta} | \boldsymbol{y}_{1:t}) \approx q_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) \propto \tilde{p}_{t\text{-}1}(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) f(\boldsymbol{x}_t | \boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) g(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta})$$

- Build ϕ_{TT} such that

$$\| \phi_{\mathsf{TT}}(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) - \sqrt{q_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1})} \|_{L^2} \leq \epsilon_t$$

  and then

$$\tilde{\pi}_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) = \phi_{\mathsf{TT}}(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1})^2 + \tau \lambda(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}), \quad \tau \leq \epsilon_t$$

- Marginalization for new filtering density

$$p(\boldsymbol{x}_t, \boldsymbol{\theta} | \boldsymbol{y}_{1:t}) \propto \int p_t(\boldsymbol{x}_t, \boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) d\boldsymbol{x}_{t\text{-}1}$$

$$p(\boldsymbol{x}_t, \boldsymbol{\theta} | \boldsymbol{y}_{1:t}) \approx \tilde{p}_t(\boldsymbol{x}_t, \boldsymbol{\theta}) \propto \int \tilde{\pi}_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) d\boldsymbol{x}_{t\text{-}1}$$

Approximation targets Computation Existing approximations Defensive term

Recursive approximation and marginalization

- Previous filtering density:

$$p_{t\text{-}1} := p(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta} | \boldsymbol{y}_{1:t\text{-}1})$$

$$p_{t\text{-}1}(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) \approx \tilde{p}_{t\text{-}1}(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta})$$

- Propagate via the joint:

$$p(\boldsymbol{x}_t, \boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta} | \boldsymbol{y}_{1:t})$$
$$\propto \; p_{t\text{-}1}(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta)$$

$$p(\boldsymbol{x}_t, \boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta} | \boldsymbol{y}_{1:t}) \approx q_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1})$$
$$\propto \tilde{p}_{t\text{-}1}(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta)$$

- Build ϕ_{TT} such that

$$\| \phi_{\mathrm{TT}}(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) - \sqrt{q_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1})} \|_{L^2} \leq \epsilon_t$$

and then

$$\tilde{\pi}_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) = \phi_{\mathrm{TT}}(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1})^2 + \tau \lambda(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}), \quad \tau \leq \epsilon_t$$

- Marginalization for new filtering density

$$p(\boldsymbol{x}_t, \boldsymbol{\theta} | \boldsymbol{y}_{1:t}) \propto \int p_t(\boldsymbol{x}_t, \boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) d\boldsymbol{x}_{t\text{-}1}$$

$$p(\boldsymbol{x}_t, \boldsymbol{\theta} | \boldsymbol{y}_{1:t}) \approx \tilde{p}_t(\boldsymbol{x}_t, \boldsymbol{\theta})$$
$$\propto \int \tilde{\pi}_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) d\boldsymbol{x}_{t\text{-}1}$$

Approximation targets Computation Existing approximations Defensive term

Recursive approximation and marginalization

- Previous filtering density:

$$p_{t\text{-}1} := p(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta} | \boldsymbol{y}_{1:t\text{-}1}) \qquad p_{t\text{-}1}(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) \approx \tilde{p}_{t\text{-}1}(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta})$$

- Propagate via the joint:

$$p(\boldsymbol{x}_t, \boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta} | \boldsymbol{y}_{1:t}) \qquad p(\boldsymbol{x}_t, \boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta} | \boldsymbol{y}_{1:t}) \approx q_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1})$$

$$\propto p_{t\text{-}1}(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta) \qquad \propto \tilde{p}_{t\text{-}1}(\boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta)$$

- Build ϕ_{TT} such that

$$\| \phi_{\mathrm{TT}}(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) - \sqrt{q_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1})} \|_{L^2} \leq \epsilon_t$$

  and then

$$\tilde{\pi}_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) = \phi_{\mathrm{TT}}(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1})^2 + \tau \lambda(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}), \quad \tau \leq \epsilon_t$$

- Marginalization for new filtering density

$$p(\boldsymbol{x}_t, \boldsymbol{\theta} | \boldsymbol{y}_{1:t}) \propto \int p_t(\boldsymbol{x}_t, \boldsymbol{x}_{t\text{-}1}, \boldsymbol{\theta}) d\boldsymbol{x}_{t\text{-}1} \qquad p(\boldsymbol{x}_t, \boldsymbol{\theta} | \boldsymbol{y}_{1:t}) \approx \tilde{p}_t(\boldsymbol{x}_t, \boldsymbol{\theta})$$

$$\propto \int \tilde{\pi}_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) d\boldsymbol{x}_{t\text{-}1}$$

Approximation targets Computation Existing approximations Defensive term

Recursive approximation and marginalization

- Previous filtering density:

$$p_{t\text{-}1} := p(x_{t\text{-}1}, \theta | y_{1:t\text{-}1}) \qquad p_{t\text{-}1}(x_{t\text{-}1}, \theta) \approx \tilde{p}_{t\text{-}1}(x_{t\text{-}1}, \theta)$$

- Propagate via the joint:

$$p(x_t, x_{t\text{-}1}, \theta | y_{1:t}) \qquad p(x_t, x_{t\text{-}1}, \theta | y_{1:t}) \approx q_t(x_t, \theta, x_{t\text{-}1})$$

$$\propto p_{t\text{-}1}(x_{t\text{-}1}, \theta) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta) \qquad \propto \tilde{p}_{t\text{-}1}(x_{t\text{-}1}, \theta) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta)$$

- Build ϕ_{TT} such that

$$\| \phi_{\mathrm{TT}}(x_t, \theta, x_{t\text{-}1}) - \sqrt{q_t(x_t, \theta, x_{t\text{-}1})} \|_{L^2} \leq \epsilon_t$$

and then

$$\tilde{\pi}_t(x_t, \theta, x_{t\text{-}1}) = \phi_{\mathrm{TT}}(x_t, \theta, x_{t\text{-}1})^2 + \tau \lambda(x_t, \theta, x_{t\text{-}1}), \quad \tau \leq \epsilon_t$$

- Marginalization for new filtering density

$$p(x_t, \theta | y_{1:t}) \qquad p(x_t, \theta | y_{1:t}) \approx \tilde{p}_t(x_t, \theta)$$

$$p(x_t, \theta | y_{1:t}) \propto \int p_t(x_t, x_{t\text{-}1}, \theta) dx_{t\text{-}1} \qquad \propto \int \tilde{\pi}_t(x_t, \theta, x_{t\text{-}1}) dx_{t\text{-}1}$$

Approximation targets Computation Existing approximations Defensive term

Recursive approximation and marginalization

- Previous filtering density:

$$p_{t\text{-}1} := p(x_{t\text{-}1}, \theta | y_{1:t\text{-}1}) \qquad p_{t\text{-}1}(x_{t\text{-}1}, \theta) \approx \tilde{p}_{t\text{-}1}(x_{t\text{-}1}, \theta)$$

- Propagate via the joint:

$$p(x_t, x_{t\text{-}1}, \theta | y_{1:t}) \qquad p(x_t, x_{t\text{-}1}, \theta | y_{1:t}) \approx q_t(x_t, \theta, x_{t\text{-}1})$$

$$\propto \; p_{t\text{-}1}(x_{t\text{-}1}, \theta) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta) \qquad \propto \tilde{p}_{t\text{-}1}(x_{t\text{-}1}, \theta) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta)$$

- Build ϕ_{TT} such that

$$\| \phi_{\mathrm{TT}}(x_t, \theta, x_{t\text{-}1}) - \sqrt{q_t(x_t, \theta, x_{t\text{-}1})} \|_{L^2} \leq \epsilon_t$$

and then

$$\tilde{\pi}_t(x_t, \theta, x_{t\text{-}1}) = \phi_{\mathrm{TT}}(x_t, \theta, x_{t\text{-}1})^2 + \tau \lambda(x_t, \theta, x_{t\text{-}1}), \quad \tau \leq \epsilon_t$$

- Marginalization for new filtering density

$$p(x_t, \theta | y_{1:t}) \approx \tilde{p}_t(x_t, \theta)$$

$$p(x_t, \theta | y_{1:t}) \propto \int p_t(x_t, x_{t\text{-}1}, \theta) dx_{t\text{-}1} \qquad \propto \int \tilde{\pi}_t(x_t, \theta, x_{t\text{-}1}) dx_{t\text{-}1}$$

Approximation targets Computation Existing approximations Defensive term

Recursive approximation and marginalization

- Previous filtering density:

$$p_{t\text{-}1} := p(x_{t\text{-}1}, \theta | y_{1:t\text{-}1}) \qquad p_{t\text{-}1}(x_{t\text{-}1}, \theta) \approx \tilde{p}_{t\text{-}1}(x_{t\text{-}1}, \theta)$$

- Propagate via the joint:

$$p(x_t, x_{t\text{-}1}, \theta | y_{1:t}) \qquad p(x_t, x_{t\text{-}1}, \theta | y_{1:t}) \approx q_t(x_t, \theta, x_{t\text{-}1})$$

$$\propto p_{t\text{-}1}(x_{t\text{-}1}, \theta) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta) \qquad \propto \tilde{p}_{t\text{-}1}(x_{t\text{-}1}, \theta) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta)$$

- Build ϕ_{TT} such that

$$\| \phi_{\mathrm{TT}}(x_t, \theta, x_{t\text{-}1}) - \sqrt{q_t(x_t, \theta, x_{t\text{-}1})} \|_{L^2} \leq \epsilon_t$$

and then

$$\tilde{\pi}_t(x_t, \theta, x_{t\text{-}1}) = \phi_{\mathrm{TT}}(x_t, \theta, x_{t\text{-}1})^2 + \tau \lambda(x_t, \theta, x_{t\text{-}1}), \quad \tau \leq \epsilon_t$$

- Marginalization for new filtering density

$$p(x_t, \theta | y_{1:t}) \propto \int p_t(x_t, x_{t\text{-}1}, \theta) dx_{t\text{-}1} \qquad p(x_t, \theta | y_{1:t}) \approx \tilde{p}_t(x_t, \theta) \propto \int \tilde{\pi}_t(x_t, \theta, x_{t\text{-}1}) dx_{t\text{-}1}$$

Approximation targets Computation Existing approximations Defensive term

Recursive approximation and marginalization

- Previous filtering density:

$$p_{t\text{-}1} := p(x_{t\text{-}1}, \theta | y_{1:t\text{-}1}) \qquad p_{t\text{-}1}(x_{t\text{-}1}, \theta) \approx \tilde{p}_{t\text{-}1}(x_{t\text{-}1}, \theta)$$

- Propagate via the joint:

$$p(x_t, x_{t\text{-}1}, \theta | y_{1:t}) \qquad p(x_t, x_{t\text{-}1}, \theta | y_{1:t}) \approx q_t(x_t, \theta, x_{t\text{-}1})$$

$$\propto p_{t\text{-}1}(x_{t\text{-}1}, \theta) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta) \qquad \propto \tilde{p}_{t\text{-}1}(x_{t\text{-}1}, \theta) f(x_t | x_{t\text{-}1}, \theta) g(y_t | x_t, \theta)$$

- Build ϕ_{TT} such that

$$\| \phi_{\mathrm{TT}}(x_t, \theta, x_{t\text{-}1}) - \sqrt{q_t(x_t, \theta, x_{t\text{-}1})} \|_{L^2} \leq \epsilon_t$$

and then

$$\tilde{\pi}_t(x_t, \theta, x_{t\text{-}1}) = \phi_{\mathrm{TT}}(x_t, \theta, x_{t\text{-}1})^2 + \tau \lambda(x_t, \theta, x_{t\text{-}1}), \quad \tau \leq \epsilon_t$$

- Marginalization for new filtering density

$$p(x_t, \theta | y_{1:t}) \approx \tilde{p}_t(x_t, \theta)$$

$$p(x_t, \theta | y_{1:t}) \propto \int p_t(x_t, x_{t\text{-}1}, \theta) dx_{t\text{-}1} \qquad \propto \int \tilde{\pi}_t(x_t, \theta, x_{t\text{-}1}) dx_{t\text{-}1}$$

Approximation targets Computation Existing approximations Defensive term

Computational cost and error accumulation [Zhao & Cui 2024]

- Cost: $\mathcal{O}((2d_x + d_\theta)r^2)$ densities + $\mathcal{O}(d_x r^3)$ flops (for marginalization)
- Assume either of the following bounds holds

$$C_t^{(g)} = \sup_{x_t \in \mathcal{X}, \theta \in \Theta} g(y_t|x_t, \theta) < \infty, \tag{1}$$

$$C_t^{(f)} = \sup_{x_{t-1} \in \mathcal{X}, \theta \in \Theta} \int f(x_t|x_{t-1}, \theta) g(y_t|x_t, \theta) dx_t < \infty. \tag{2}$$

- At time t, approximation satisfies $\|\sqrt{q_t} - \phi_{\mathrm{TT}}\|_{L^2} \leq \epsilon_t$. See [Cui & Dolgov 2023] and [Griebel & Harbrecht 2023].
- Then the Hellinger distance between the joint density p_t and its (normalized) recursive tensor-train approximation $\tilde{p}_t \propto \tilde{\pi}_t$ is bounded by

$$D_{\mathrm{H}}(p_t, \tilde{p}_t) \leq \frac{2}{\sqrt{p(y_{1:t})}} \sum_{k=1}^{t} C^{t-k} \epsilon_k,$$

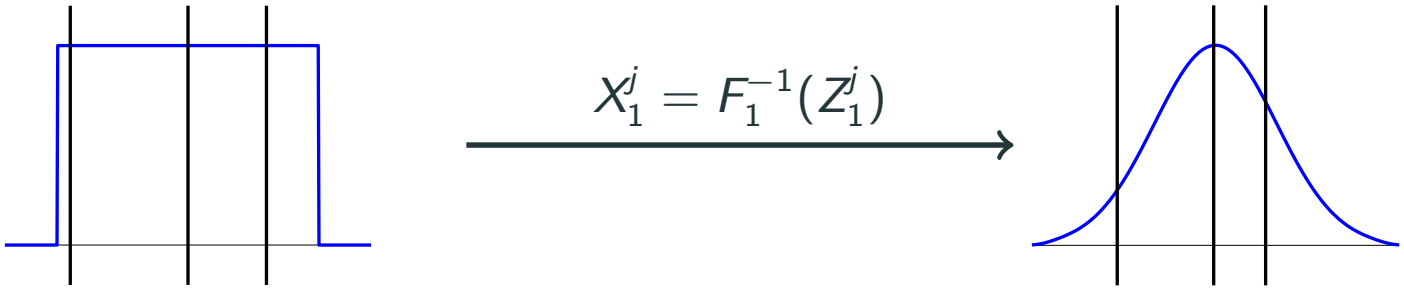
where $p(y_{1:t})$ is the evidence and $C = \sup_t C_t^{(h)} < \infty$ for $h \in \{f, g\}$.

Filtering and smoothing

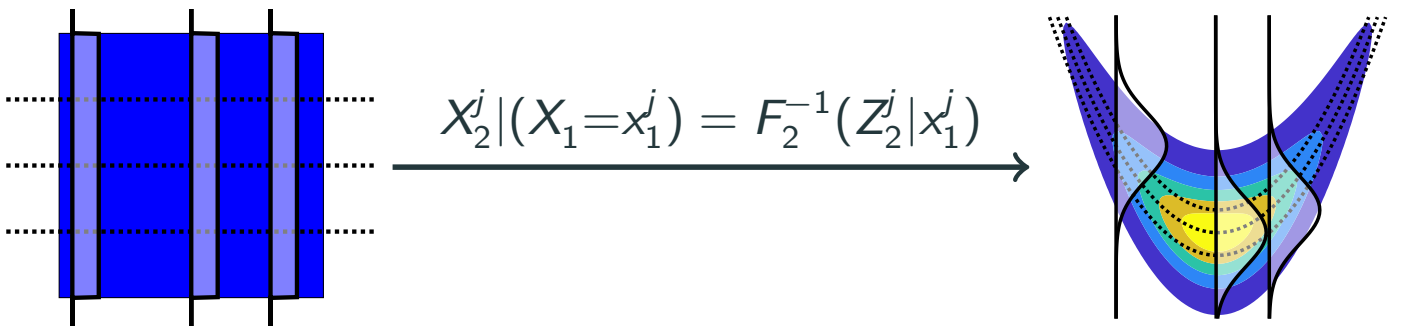
Knothe–Rosenblatt (KR) rearrangement

A prob. density has factorization: $\pi(x_1, \ldots, x_d) = \pi_1(x_1) \cdots \pi_k(x_k|x_{<k}) \cdots \pi_d(x_d|x_{<d})$

Couple $X \sim \pi$ with $Z \sim \text{uniform}(0,1]^d$



(a) $u_1(z_1)$ (b) $\pi_{\leq 1}(x_1)$



(c) $u(z_1, z_2) = u_1(z_1)u_2(z_2)$ (d) $\pi(x_1, x_2) = \pi_{\leq 1}(x_1)\pi_2(x_2|x_1)$

TT-based KR rearrangement

- In each iteration, unnormalized (joint) filtering density π_t has an approximation

$$\pi_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) \approx \tilde{\pi}_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) := \phi^2_{\mathrm{TT}}(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) + \tau\lambda(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1})$$

- Obtain marginal densities and conditional CDFs using TT, e.g.,

$$\tilde{\pi}_{t,\leq k}(\boldsymbol{x}_{t,\leq k}) = \int \phi^2_{\mathrm{TT}}(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) d\boldsymbol{x}_{t,>k} d\boldsymbol{\theta} d\boldsymbol{x}_{t\text{-}1} + \tau\lambda(\boldsymbol{x}_{t,\leq k})$$

$$F_{t,k}(x_{t,k} | \boldsymbol{x}_{t,<k}, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) = \int_{-\infty}^{x_{t,k}} \frac{\tilde{\pi}_{t,\leq k}(\boldsymbol{x}_{t,<k}, x'_{t,k})}{\tilde{\pi}_{t,<k}(\boldsymbol{x}_{t,<k})} dx'_{t,k}$$

- Integrate $\hat{\pi}_t$ from right to left to obtain lower triangular KR rearrangement

$$\mathcal{F}^l_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) = \begin{bmatrix} \mathcal{F}^l_{t,t} & (\boldsymbol{x}_t) \\ \mathcal{F}^l_{t,\theta} & (\boldsymbol{\theta} \quad | \boldsymbol{x}_t) \\ \mathcal{F}^l_{t,t\text{-}1} & (\boldsymbol{x}_{t\text{-}1} | \boldsymbol{\theta}, \boldsymbol{x}_t) \end{bmatrix}$$

- (Conditional) sampling from $\tilde{\pi}_t$ is easy—a sequence of 1D inverse CDF

TT-based KR rearrangement

- Use the last block of the lower triangular KR rearrangement

$$\mathcal{F}_t^l(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) = \begin{bmatrix} \mathcal{F}_{t,t}^l & (\boldsymbol{x}_t) \\ \mathcal{F}_{t,\theta}^l & (\boldsymbol{\theta} \quad |\boldsymbol{x}_t) \\ \mathcal{F}_{t,t\text{-}1}^l & (\boldsymbol{x}_{t\text{-}1}|\boldsymbol{\theta}, \boldsymbol{x}_t) \end{bmatrix}$$

  for generating (backward) conditional random variable $\widehat{\boldsymbol{X}}_{t\text{-}1}|\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{X}}_t$

- Integrate $\tilde{\pi}_t$ from left to right to obtain upper triangular KR rearrangement

$$\mathcal{F}_t^u(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) = \begin{bmatrix} \mathcal{F}_{t,t}^u & (\boldsymbol{x}_t \quad |\boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) \\ \mathcal{F}_{t,\theta}^u & (\boldsymbol{\theta} \quad |\boldsymbol{x}_{t\text{-}1}) \\ \mathcal{F}_{t,t\text{-}1}^u & (\boldsymbol{x}_{t\text{-}1}) \end{bmatrix}$$

  for generating (forward) conditional random variable $\widehat{\boldsymbol{X}}_t|\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{X}}_{t\text{-}1}$

Accompanying particle filter (forward sampling) $\widehat{x}_t^{(i)} | \widehat{\theta}^{(i)}, \widehat{x}_{t-1}^{(i)}$

Use the upper-triangular $\mathcal{F}_{t,t}^u$

$$\mathcal{F}_t^u(x_t, \theta, x_{t\text{-}1}) = \begin{bmatrix} \mathcal{F}_{t,t}^u & (x_t & | \theta, x_{t\text{-}1}) \\ \mathcal{F}_{t,\theta}^u & (\theta & | x_{t\text{-}1}) \\ \mathcal{F}_{t,t\text{-}1}^u & (x_{t\text{-}1}) & \end{bmatrix}$$

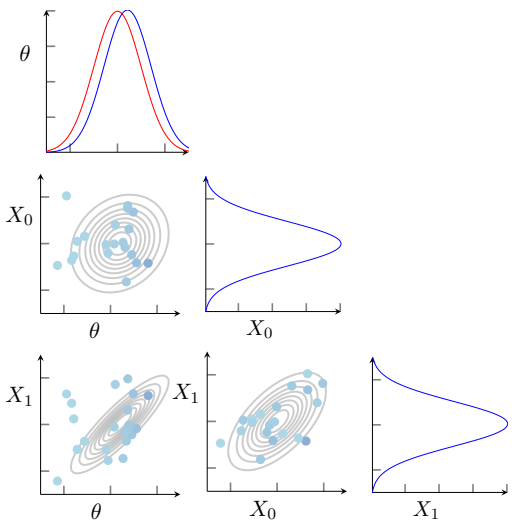
$\{\widehat{X}_{0:t\text{-}1}^{(i)}, \widehat{\Theta}^{(i)}, w_{t\text{-}1}^{(i)}\}$

$(\mathcal{F}_{t,t}^u)^{-1}$

$\{\widehat{X}_{0:t}^{(i)}, \widehat{\Theta}^{(i)}, w_{t\text{-}1}^{(i)}\}$

$w_t^{(i)} = w_{t\text{-}1}^{(i)} \times \omega_F^{(i)}$

$\{\widehat{X}_{0:t}^{(i)}, \widehat{\Theta}^{(i)}, \ldots$

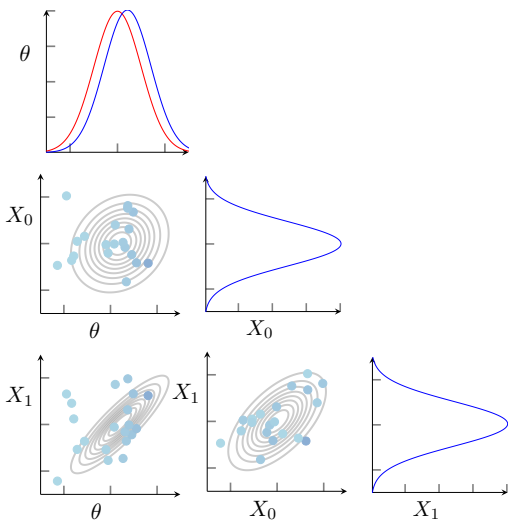


Accompanying particle filter (forward sampling) $\widehat{x}_t^{(i)} | \widehat{\theta}^{(i)}, \widehat{x}_{t-1}^{(i)}$

Use the upper-triangular $\mathcal{F}_{t,t}^u$

$$\mathcal{F}_t^u(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) = \begin{bmatrix} \mathcal{F}_{t,t}^u & (\boldsymbol{x}_t & | \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) \\ \mathcal{F}_{t,\theta}^u & (\boldsymbol{\theta} & | \boldsymbol{x}_{t\text{-}1}) \\ \mathcal{F}_{t,t\text{-}1}^u & (\boldsymbol{x}_{t\text{-}1}) & \end{bmatrix}$$

$\{\widehat{\boldsymbol{X}}_{0:t\text{-}1}^{(i)}, \widehat{\boldsymbol{\Theta}}^{(i)}, w_{t\text{-}1}^{(i)}\}$ $\rightarrow$ $(\mathcal{F}_{t,t}^u)^{-1}$ $\rightarrow$ $\{\widehat{\boldsymbol{X}}_{0:t}^{(i)}, \widehat{\boldsymbol{\Theta}}^{(i)}, w_{t\text{-}1}^{(i)}\}$ $\rightarrow$ $w_t^{(i)} = w_{t\text{-}1}^{(i)} \times \omega_{\mathsf{F}}^{(i)}$ $\rightarrow$ $\{\widehat{\boldsymbol{X}}_{0:t}^{(i)}, \widehat{\boldsymbol{\Theta}}^{(i)}, \dots$

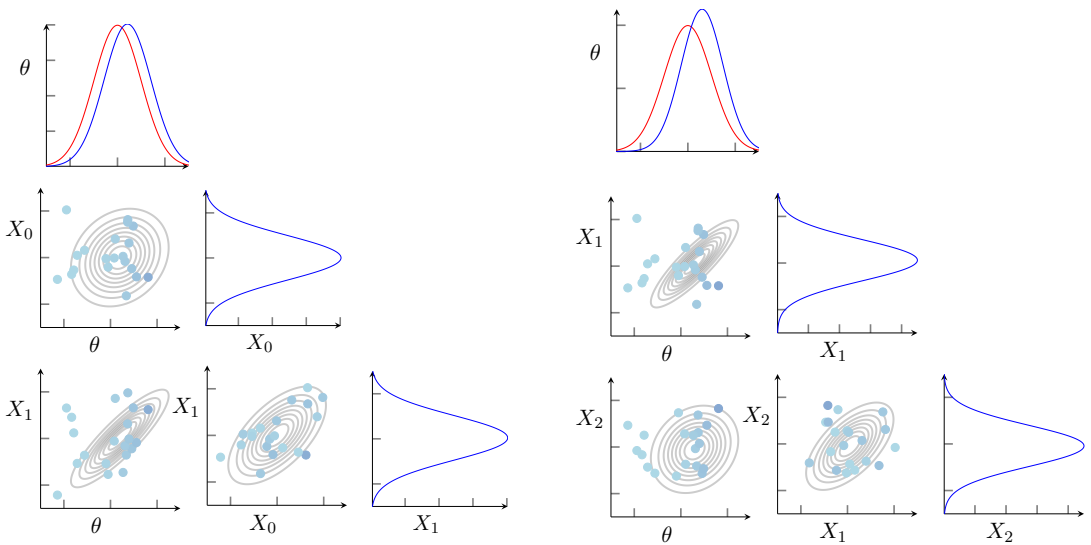


Accompanying particle filter (forward sampling) $\widehat{x}_t^{(i)} | \widehat{\theta}^{(i)}, \widehat{x}_{t-1}^{(i)}$

Use the upper-triangular $\mathcal{F}_{t,t}^u$

$$\mathcal{F}_t^u(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) = \begin{bmatrix} \mathcal{F}_{t,t}^u & (\boldsymbol{x}_t & | \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) \\ \mathcal{F}_{t,\theta}^u & (\boldsymbol{\theta} & | \boldsymbol{x}_{t\text{-}1}) \\ \mathcal{F}_{t,t\text{-}1}^u & (\boldsymbol{x}_{t\text{-}1}) & \end{bmatrix}$$

$\{\widehat{\boldsymbol{X}}_{0:t\text{-}1}^{(i)}, \widehat{\boldsymbol{\Theta}}^{(i)}, w_{t\text{-}1}^{(i)}\}$ → $(\mathcal{F}_{t,t}^u)^{-1}$ → $\{\widehat{\boldsymbol{X}}_{0:t}^{(i)}, \widehat{\boldsymbol{\Theta}}^{(i)}, w_{t\text{-}1}^{(i)}\}$ → $w_t^{(i)} = w_{t\text{-}1}^{(i)} \times \omega_{\mathsf{F}}^{(i)}$ → $\{\widehat{\boldsymbol{X}}_{0:t}^{(i)}, \widehat{\boldsymbol{\Theta}}^{(i)}, \dots$

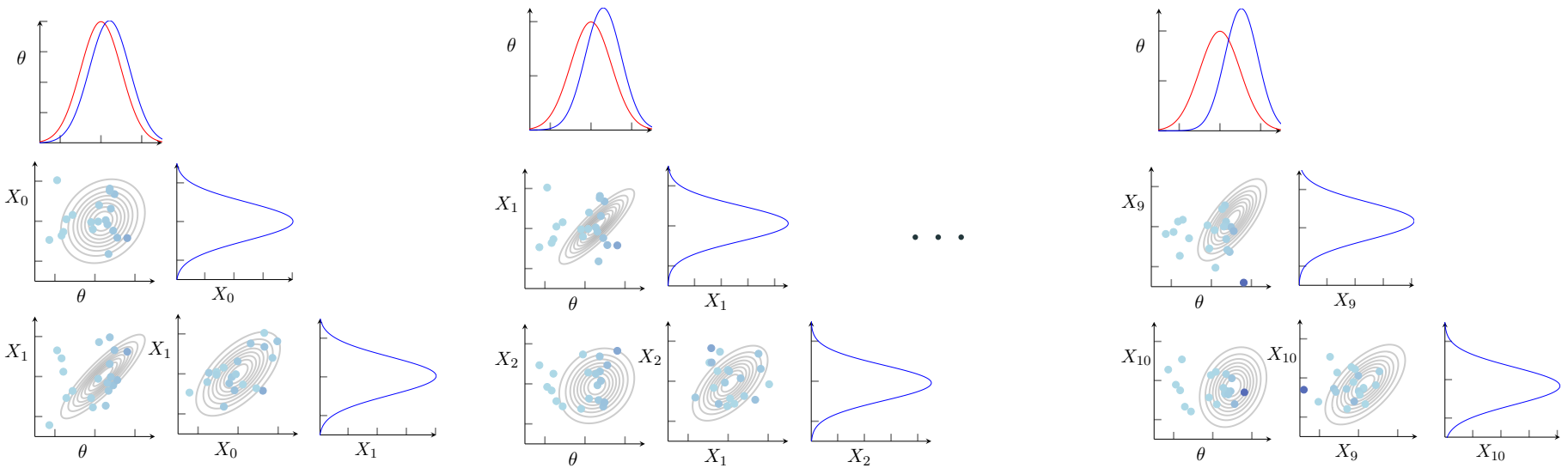


Accompanying particle filter (forward sampling) $\widehat{x}_t^{(i)} | \widehat{\theta}^{(i)}, \widehat{x}_{t-1}^{(i)}$

Use the upper-triangular $\mathcal{F}_{t,t}^u$

$$\mathcal{F}_t^u(x_t, \theta, x_{t-1}) = \begin{bmatrix} \mathcal{F}_{t,t}^u & (x_t & | \theta, x_{t-1}) \\ \mathcal{F}_{t,\theta}^u & (\theta & | x_{t-1}) \\ \mathcal{F}_{t,t-1}^u & (x_{t-1}) & \end{bmatrix}$$

$\{\widehat{x}_{0:t-1}^{(i)}, \widehat{\Theta}^{(i)}, w_{t-1}^{(i)}\}$ $\rightarrow$ $(\mathcal{F}_{t,t}^u)^{-1}$ $\rightarrow$ $\{\widehat{x}_{0:t}^{(i)}, \widehat{\Theta}^{(i)}, w_{t-1}^{(i)}\}$ $\rightarrow$ $w_t^{(i)} = w_{t-1}^{(i)} \times \omega_F^{(i)}$ $\rightarrow$ $\{\widehat{x}_{0:t}^{(i)}, \widehat{\Theta}^{(i)}, \ldots$



Accompanying smoother (backward sampling) $\widehat{x}_{t\text{-}1}|\widehat{\Theta}^{(i)}, \widehat{x}_t^{(i)}$

Use the lower-triangular $\mathcal{F}^I_{t,t\text{-}1}$

$$\mathcal{F}^I_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) = \begin{bmatrix} \mathcal{F}^I_{t,t} & (\boldsymbol{x}_t) \\ \mathcal{F}^I_{t,\theta} & (\boldsymbol{\theta} \quad |\boldsymbol{x}_t) \\ \mathcal{F}^I_{t,t\text{-}1} & (\boldsymbol{x}_{t\text{-}1}|\boldsymbol{\theta}, \boldsymbol{x}_t) \end{bmatrix}$$

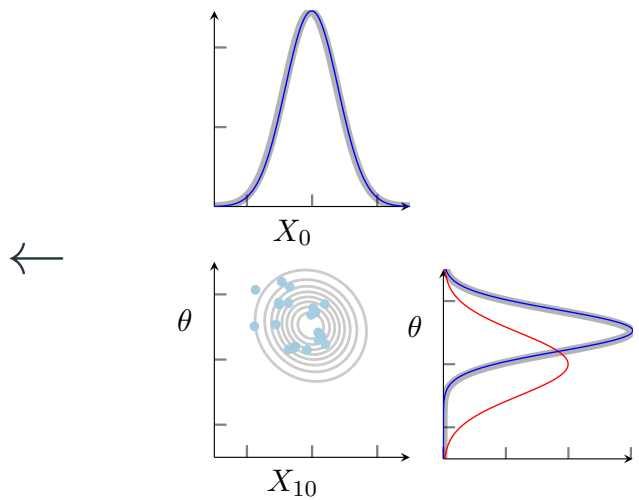
$(\widehat{\Theta}^{(i)}, \widehat{x}^{(i)}_{t-1:T}, w^{(i)}_{t-1})$

$w^{(i)}_{t-1} = w^{(i)}_t \times \omega^{(i)}_B$

$(\widehat{\Theta}^{(i)}, \widehat{x}^{(i)}_{t:T},$

$(\mathcal{F}^I_{t,t\text{-}1})^{-1}$

$(\widehat{\Theta}^{(i)}, \widehat{x}^{(i)}_{t-1:T}, w^{(i)}_t)$



Accompanying smoother (backward sampling) $\widehat{x}_{t\text{-}1}|\widehat{\Theta}^{(i)}, \widehat{x}_t^{(i)}$

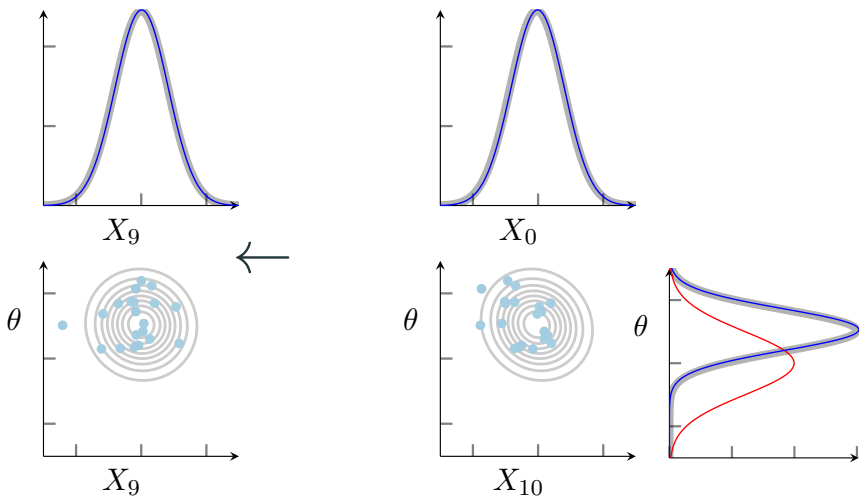
Use the lower-triangular $\mathcal{F}^I_{t,t\text{-}1}$

$$\mathcal{F}^I_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) = \begin{bmatrix} \mathcal{F}^I_{t,t} & (\boldsymbol{x}_t) \\ \mathcal{F}^I_{t,\theta} & (\boldsymbol{\theta} \quad |\boldsymbol{x}_t) \\ \mathcal{F}^I_{t,t\text{-}1} & (\boldsymbol{x}_{t\text{-}1}|\boldsymbol{\theta}, \boldsymbol{x}_t) \end{bmatrix}$$

$(\widehat{\Theta}^{(i)}, \widehat{\boldsymbol{X}}^{(i)}_{t\text{-}1:T}, w^{(i)}_{t\text{-}1})$ $\qquad$ $(\widehat{\Theta}^{(i)}, \widehat{\boldsymbol{X}}^{(i)}_{t:T},$

$w^{(i)}_{t\text{-}1} = w^{(i)}_t \times \omega^{(i)}_B$ $\qquad$ $(\mathcal{F}^I_{t,t\text{-}1})^{-1}$

$(\widehat{\Theta}^{(i)}, \widehat{\boldsymbol{X}}^{(i)}_{t\text{-}1:T}, w^{(i)}_t)$



Accompanying smoother (backward sampling) $\widehat{x}_{t\text{-}1}|\widehat{\Theta}^{(i)}, \widehat{x}_t^{(i)}$

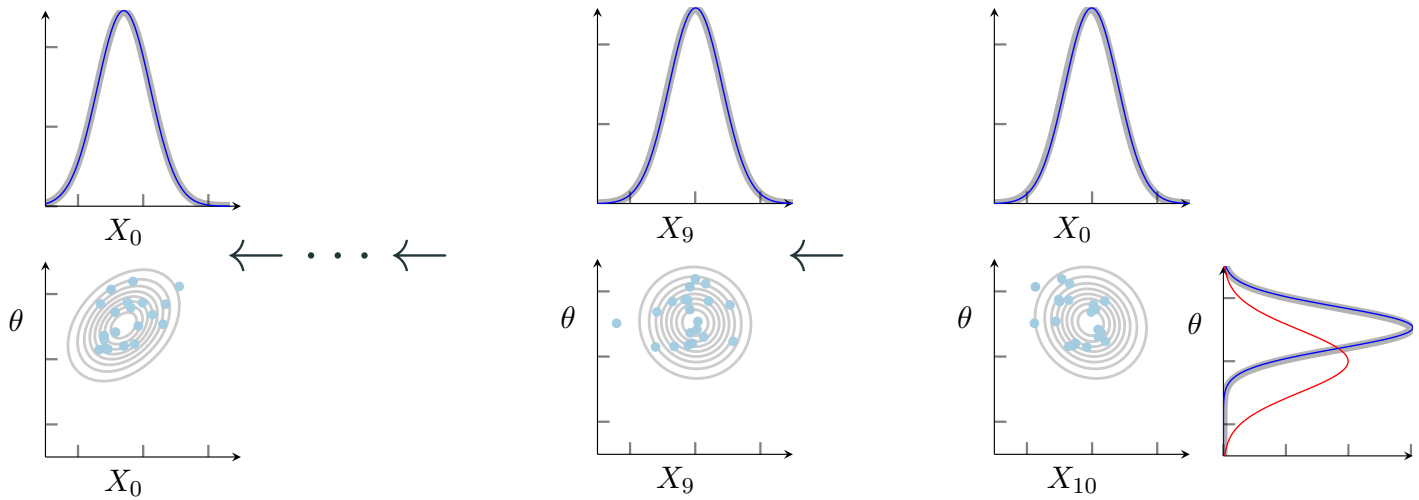
Use the lower-triangular $\mathcal{F}^I_{t,t\text{-}1}$

$$\mathcal{F}^I_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) = \begin{bmatrix} \mathcal{F}^I_{t,t} & (\boldsymbol{x}_t) \\ \mathcal{F}^I_{t,\theta} & (\boldsymbol{\theta} \quad |\boldsymbol{x}_t) \\ \mathcal{F}^I_{t,t\text{-}1} & (\boldsymbol{x}_{t\text{-}1}|\boldsymbol{\theta}, \boldsymbol{x}_t) \end{bmatrix}$$

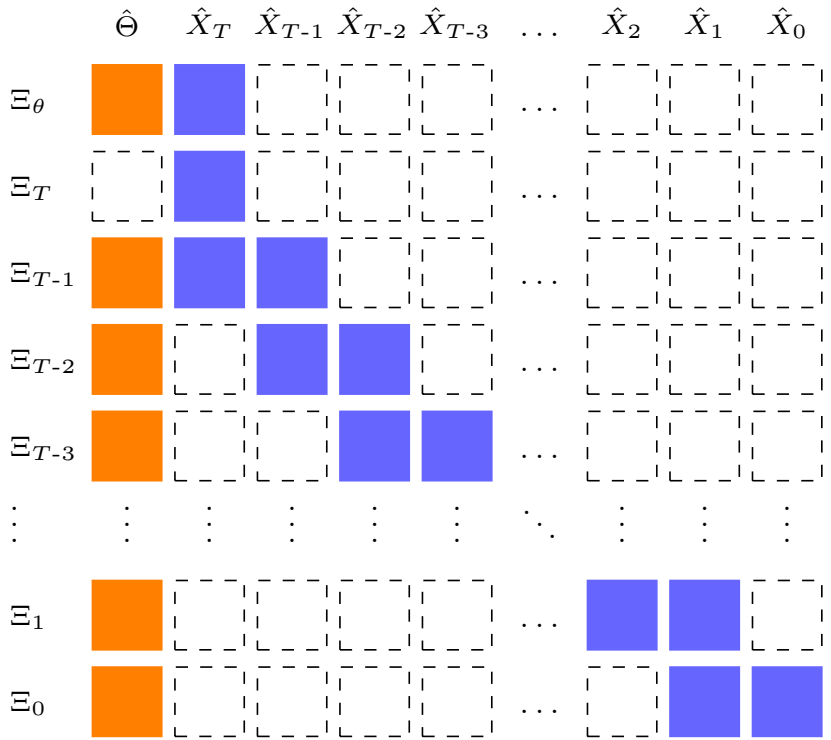
$(\widehat{\Theta}^{(i)}, \widehat{\boldsymbol{x}}^{(i)}_{t\text{-}1:T}, w^{(i)}_{t\text{-}1})$ $\qquad$ $(\widehat{\Theta}^{(i)}, \widehat{\boldsymbol{x}}^{(i)}_{t:T},$

$w^{(i)}_{t\text{-}1} = w^{(i)}_t \times \omega^{(i)}_B$ $\qquad$ $(\mathcal{F}^I_{t,t\text{-}1})^{-1}$

$(\widehat{\Theta}^{(i)}, \widehat{\boldsymbol{x}}^{(i)}_{t\text{-}1:T}, w^{(i)}_t)$



Block sparse structure of the joint KR rearrangement



$$\Xi_\theta = \mathcal{F}^l_{T,\theta} \quad (\hat{\Theta}|\hat{X}_T)$$

$$\Xi_T = \mathcal{F}^l_{T,T} \quad (\hat{X}_T)$$

$$\Xi_{T\text{-}1} = \mathcal{F}^l_{T,T\text{-}1} \quad (\hat{X}_{T\text{-}1}|\hat{X}_T, \hat{\Theta})$$

$$\Xi_{T\text{-}2} = \mathcal{F}^l_{T\text{-}1,T\text{-}2}(\hat{X}_{T\text{-}2}|\hat{X}_{T\text{-}1}, \hat{\Theta})$$

$$\Xi_{T\text{-}3} = \mathcal{F}^l_{T\text{-}2,T\text{-}3}(\hat{X}_{T\text{-}3}|\hat{X}_{T\text{-}2}, \hat{\Theta})$$

$$\Xi_1 = \mathcal{F}^l_{2,1} \quad (\hat{X}_1|\hat{X}_2, \hat{\Theta})$$

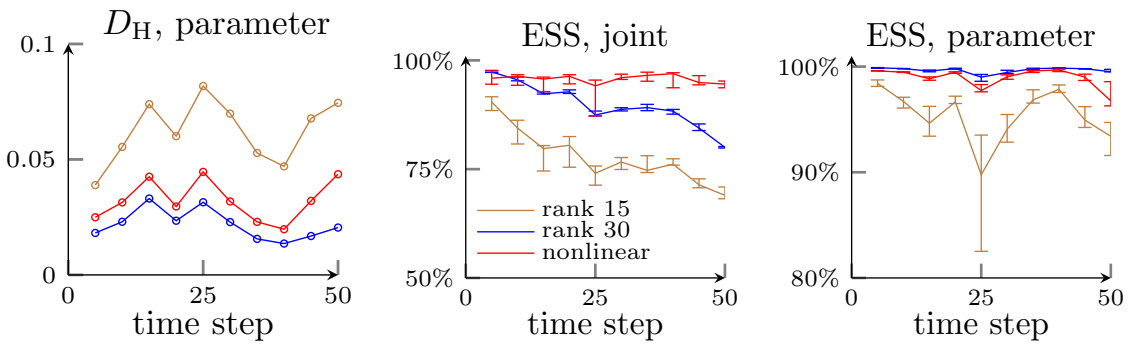
$$\Xi_0 = \mathcal{F}^l_{1,0} \quad (\hat{X}_0|\hat{X}_1, \hat{\Theta})$$

Examples

Example 1: Kalman filter (with analytical solutions)

$$\begin{cases} \boldsymbol{X}_t - \mu &= \sqrt{1-a^2}\,(\boldsymbol{X}_{t-1} - \mu) + a\,\varepsilon_t^{(x)} \\ \boldsymbol{Y}_t &= C\boldsymbol{X}_t + d\varepsilon_t^{(y)} \end{cases}$$

- $\boldsymbol{X}_t, \boldsymbol{Y}_t \in \mathbb{R}^3$ with fixed $\mu = 0, C \in \mathbb{R}^{3\times 3}$
- Unknown parameter $\boldsymbol{\theta} = (a, d)$
- Synthetic data are generated using $\boldsymbol{\theta} = (0.8, 0.5)$, to $T = 50$

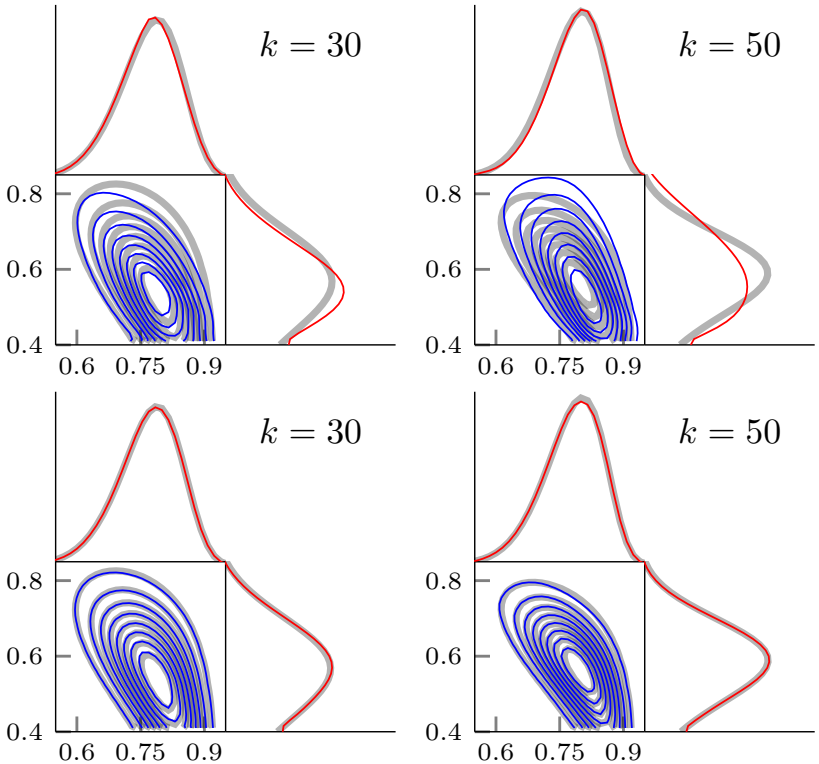


- Non-linear precondition: composition of two rank-15 tensor trains†

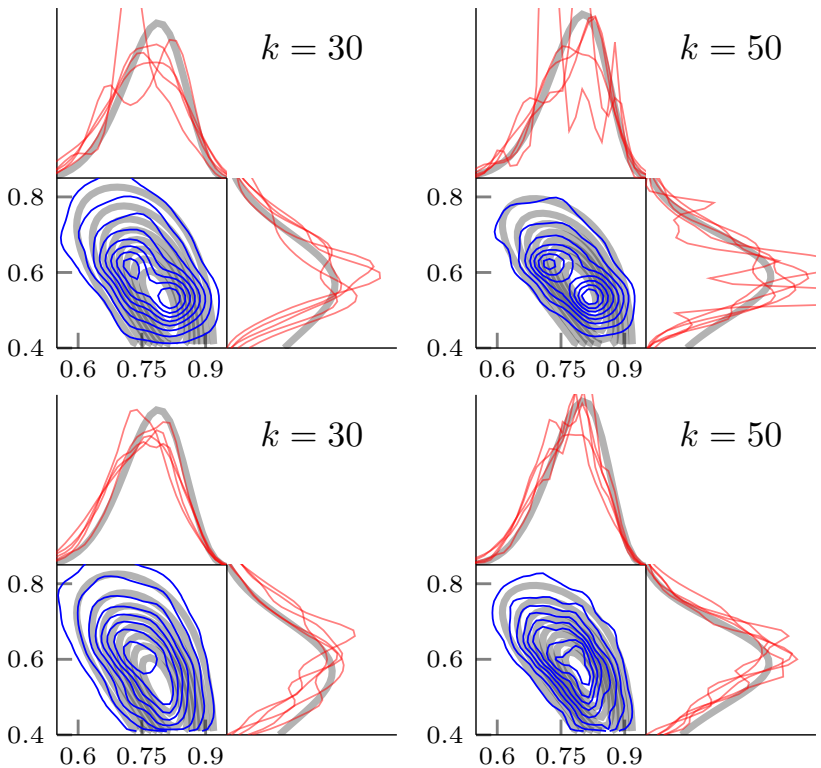
†5. More examples

Example 1: parameter posterior densities

TT (top: rank 15; bottom: rank 30)



SMC^2 (top: 500; bottom: 5000 particles)



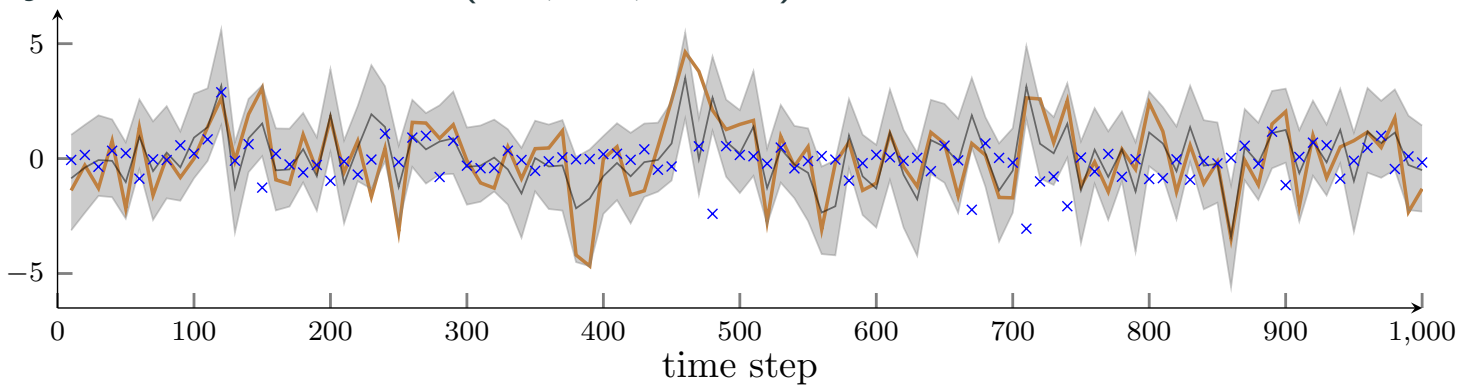
- The cost of running SMC^2 with 5000 particles is comparable to TT with rank 30.

Example 2: stochastic volatility

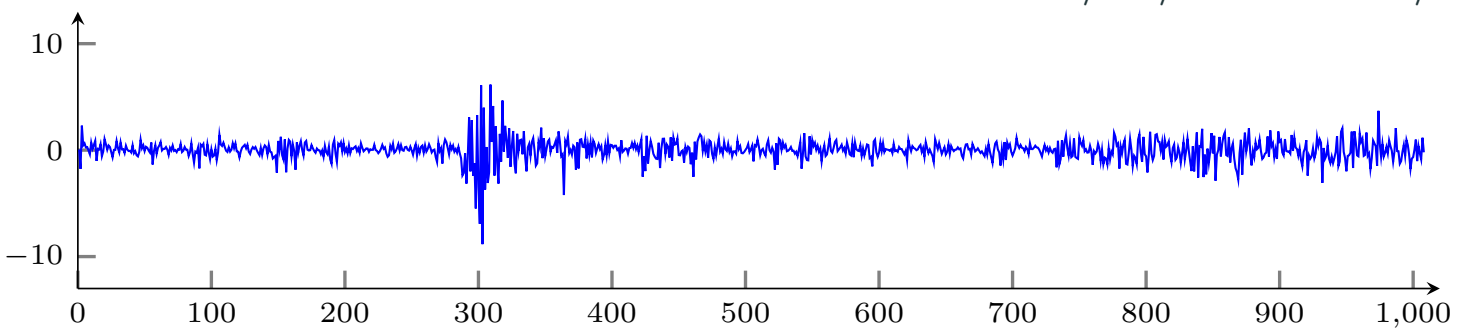
- The unknown parameter $\boldsymbol{\theta} = (\gamma, \beta, \sigma)$

$$\begin{cases} \boldsymbol{X}_t &= \gamma \boldsymbol{X}_{t-1} + \sigma \varepsilon_t^{(x)} \\ \boldsymbol{Y}_t &= \varepsilon_t^{(y)} \beta \exp(\frac{1}{2} \boldsymbol{X}_t) \end{cases}$$

- **Synthetic data:** $\theta = (0.6, 0.4, \sigma = 1)$ to $T = 1000$

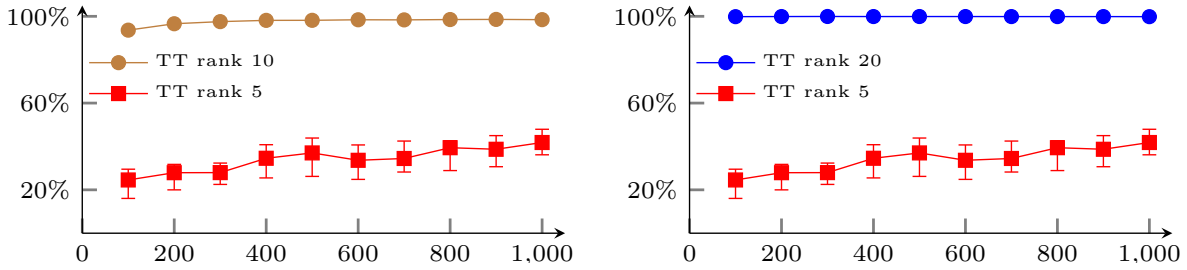


- **Real data:** returns of the S&P 500 index from 31/12/2019 to 29/12/2023

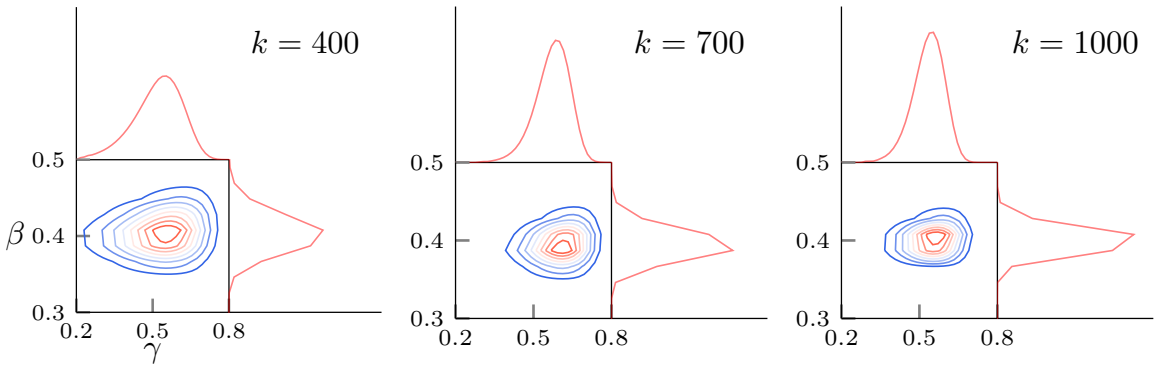


Example 2: synthetic data

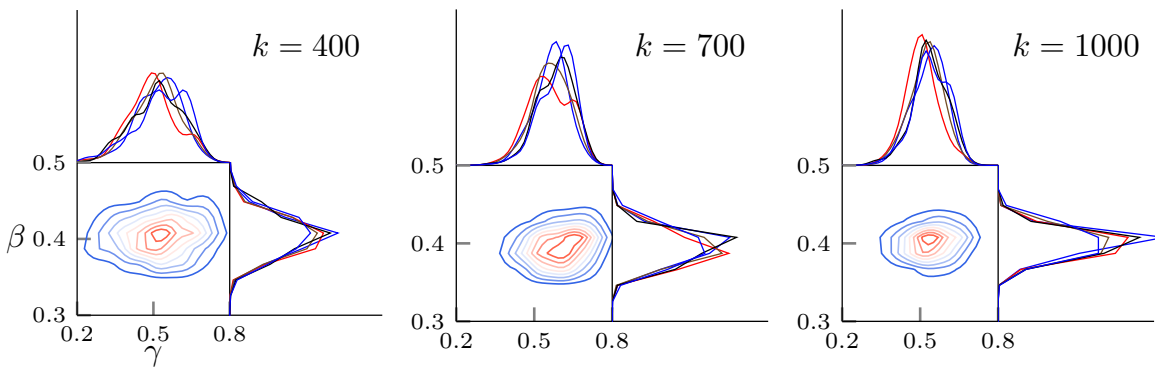
TT ESS



TT density



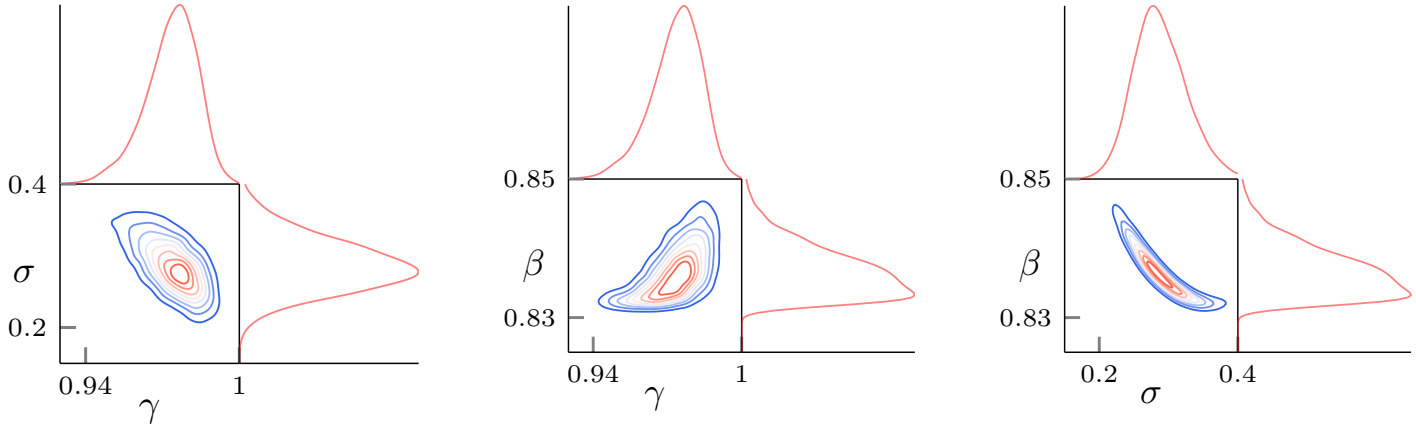
SMC^2



- Computation time of TT is 10% of that of SMC^2 (1000 parameter particles)

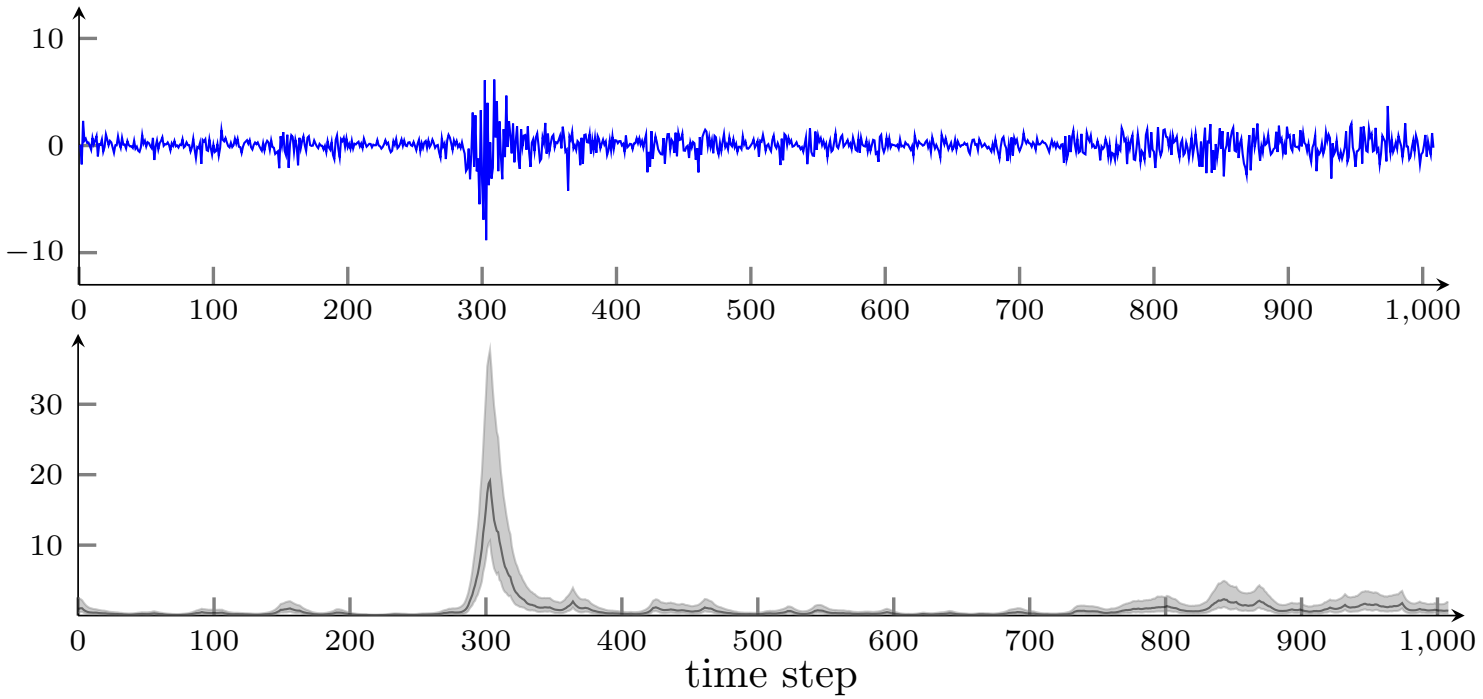
Example 2: S&P 500 returns

Estimated parameters



S&P 500 returns

Estimated volatilities



- The peak of estimated volatilities matches the volatile returns at $t = 300$

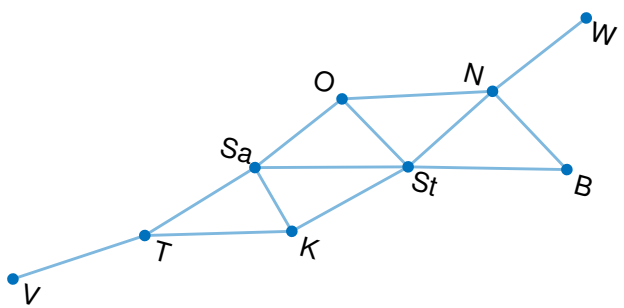
Example 3: susceptible-infectious-removed (SIR) model

$$\begin{cases} dS_j &= \big(-\kappa_j S_j I_j + \frac{1}{2}\sum_{i\in\mathcal{I}_j}(S_i - S_j)\big)dt + \sigma d\mathcal{W} \\ dI_j &= \big(\kappa_j S_j I_j - \nu_j I_j + \frac{1}{2}\sum_{i\in\mathcal{I}_j}(I_i - I_j)\big)dt + \sigma d\mathcal{W} \\ dR_j &= \big(\nu_j I_j + \frac{1}{2}\sum_{i\in\mathcal{I}_j}(R_i - R_j)\big)dt + \sigma d\mathcal{W} \end{cases}, \quad j = 1, \ldots, J$$

- $J \in \mathbb{N}$: number of spatially dependent demographic compartments
- $\mathcal{I}_j$: geographical connections
- Collect observed date from $\{I_j\}^J$

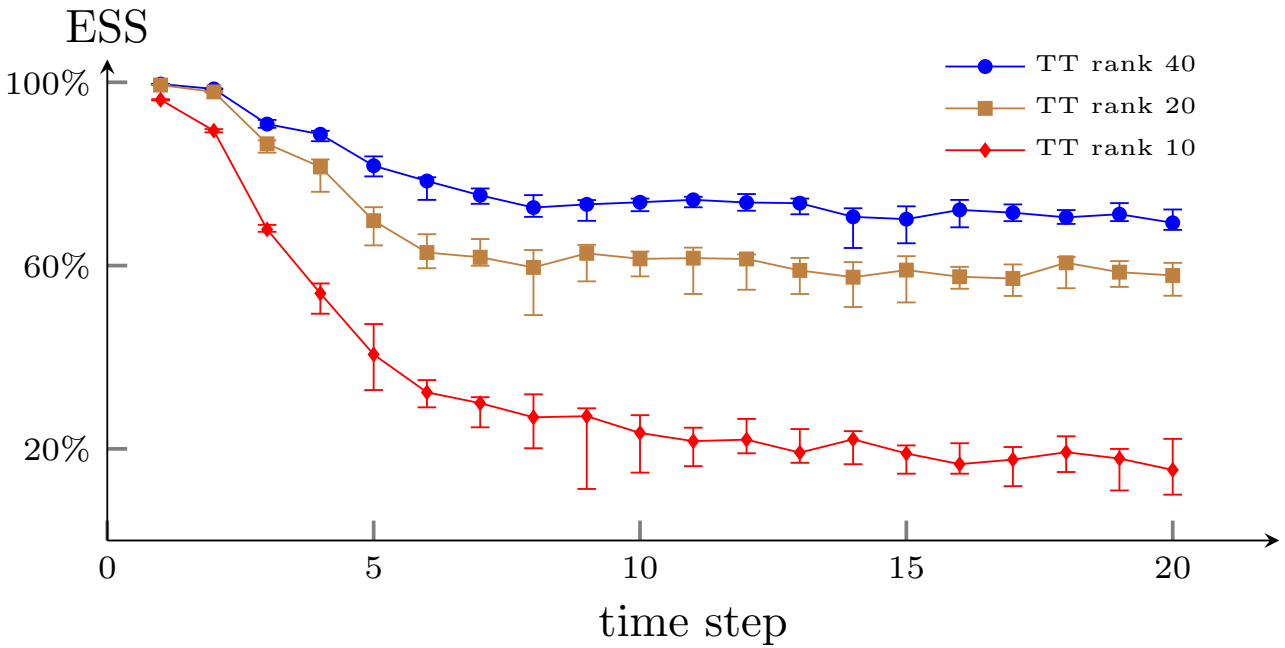
$$\boldsymbol{Y}_{t,j} = \mathbf{I}_{t,j} + \varepsilon_{t,j}^{(y)}, \quad k = 1, 2, \ldots, T, \quad j = 1, 2, \ldots, J.$$

- Parameters: $\boldsymbol{\theta} = (\kappa_j, \mu_j, \sigma)$ with $\kappa_j = 0.1$, $\nu_j = 18$, $\sigma^2 = 50$ for simulating synthetic data

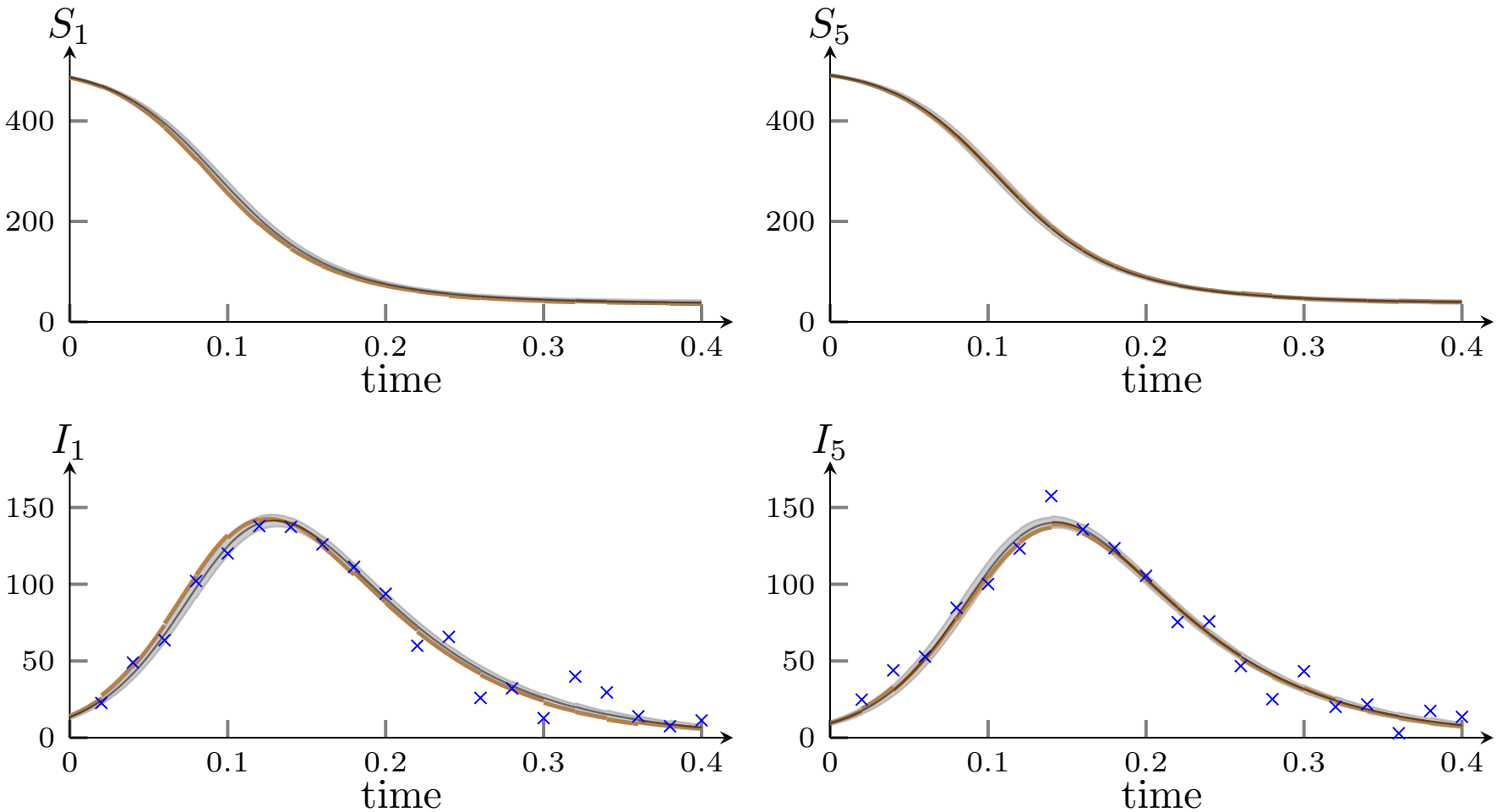


V - Vorarlberg
T - Tyrol
Sa - Salzburg
K - Carinthia
St - Styria
O - Upper Austria
N - Lower Austria
W - Vienna
B - Burgenland

Example 3: effective sample size



Example 3: state trajectories



- The estimated paths of Vorarlberg and Styria

References:

- Zhao, Y., & Cui, T. (2024). Tensor-train methods for sequential state and parameter learning in state-space models. *Journal of Machine Learning Research*, 25 (244), 1–51.
- Matlab code: `https://github.com/DeepTransport/tensor-ssm-paper-demo`
- Dolgov, S., Anaya-Izquierdo, K., Fox, C., & Scheichl, R. (2020). Approximation and sampling of multivariate probability distributions in the tensor train decomposition. *Statistics and Computing*, 30, 603–625.
- Cui, T., & Dolgov, S. (2022). Deep composition of Tensor-Trains using squared inverse Rosenblatt transports. *Foundations of Computational Mathematics*, 22(6), 1863–1922.
- DIRT package (Matlab): `https://github.com/DeepTransport/deep-tensor`
- Spantini, A., Bigoni, D., & Marzouk, Y. (2018). Inference via low-dimensional couplings. *Journal of Machine Learning Research*, 19(66), 1–71.

Preconditioning

Preconditioning

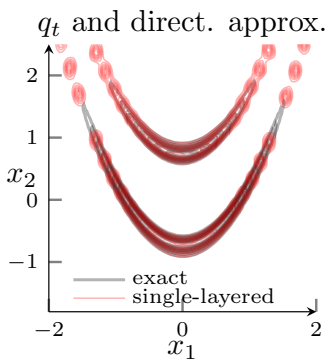
Consider a function $q(x_1, x_2)$ with high tensor ranks and a transport map M. The pushforward function $q_\sharp(u_1, u_2) \propto M_\sharp q(u_1, u_2)$ may be easier to approximate.

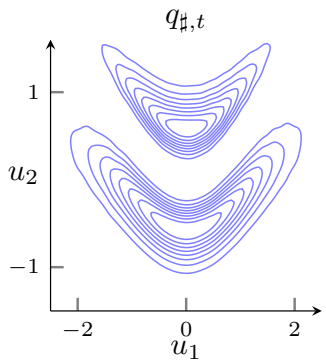
Example: linear preconditioning

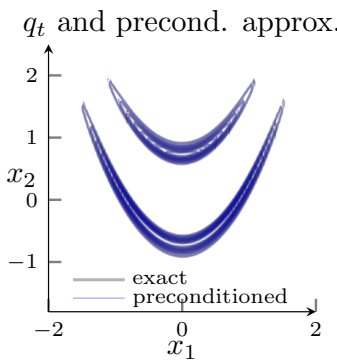
For $\boldsymbol{X} \sim N(\mu, LL^\top)$ with density $q(\boldsymbol{x})$, define the linear transport map $M(x) = L^{-1}(\boldsymbol{x}-\mu)$, then the pushforward function $q_\sharp(\boldsymbol{u}) \propto M_\sharp q(\boldsymbol{u})$ is rank-one.

Example: Non-linear preconditioning via tempering

- Tempered density: $q_t^\alpha(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1}) = q_t(\boldsymbol{x}_t, \boldsymbol{\theta}, \boldsymbol{x}_{t\text{-}1})^\alpha$
- Build TT approximation $\phi_{\rho,t} \approx \sqrt{q^\alpha}$, and hence the KR map $M_\sharp q_t^\alpha \approx \rho$
- The pushforward density is given by $q_{\sharp,t} = M_\sharp q_t \approx q_t^{1-\alpha}$







More examples

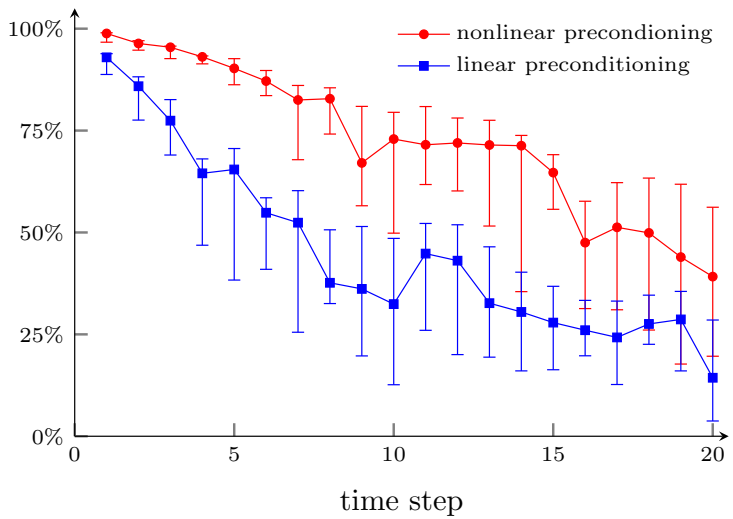
Example 4: predator prey

- State transition: $\boldsymbol{x}_t = (P_t, Q_t)$ following

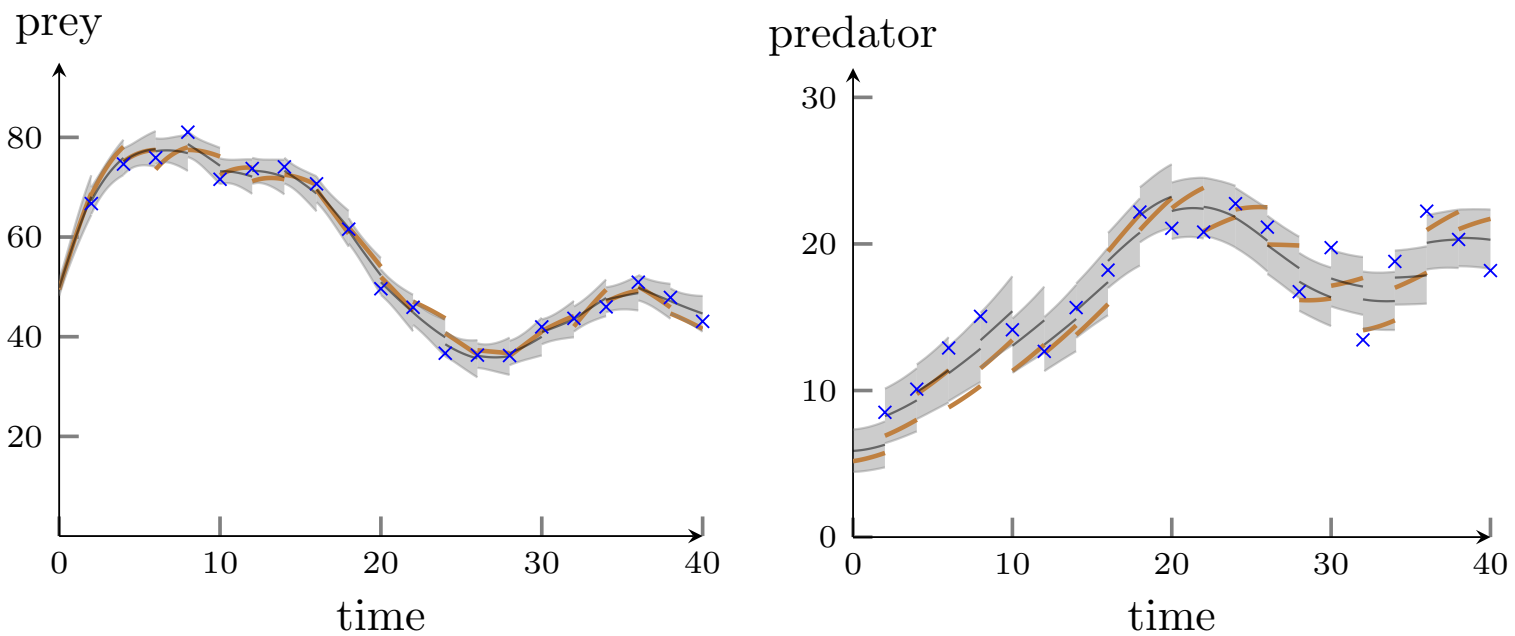
$$\begin{cases} dP &= \left(rP\left(1 - \frac{P}{K}\right) - s\left(\frac{sPQ}{a+P}\right)\right)dt + d\mathcal{W} \\ dQ &= \left(u\left(\frac{PQ}{a+P} - vQ\right)\right)dt + d\mathcal{W} \end{cases}$$

- Observation: $\boldsymbol{Y}_t = \boldsymbol{X}_t + \varepsilon_t^{(y)}$
- Parameters: $\boldsymbol{\theta} = (r, K, a, s, u, v)$

TT ESS



Example 4: state trajectories



- Trajectories of the states estimated from tensor-train smoother
- Orange curves: discretized true trajectory