Transport & Multilevel Approaches for Large-Scale PDE-Constrained Bayesian Inference

Robert Scheichl

Institute of Applied Mathematics &
Interdisciplinary Center for Scientific Computing
Heidelberg University

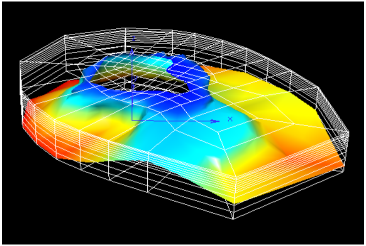
Collaborators:
K Anaya-Izquierdo & S Dolgov (Bath); C Fox (Otago); T Dodwell (Exeter);
AL Teckentrup (Edinburgh); T Cui (Monash); G Detommaso (Amazon)

"Computational Statistics and Data-Driven Models"
ICERM, Brown University, March 23, 2020

Inverse Problems

Data

Parameter



$$\boxed{y} = F(\boxed{x}) + e$$

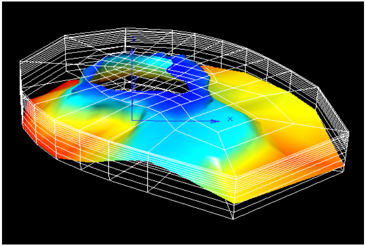
forward model (PDE)

observation/model errors

Inverse Problems

Data

Parameter

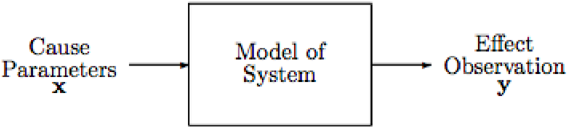


$$y = F(x) + e$$

forward model (PDE) observation/model errors

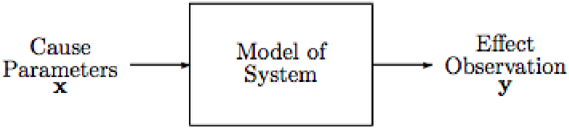
$y \in \mathbb{R}^{N_y}$	Data y are limited in number, noisy, and indirect.
$x \in X$	Parameter x often a function (discretisation needed).
$F : X \to \mathbb{R}^{N_y}$	Continuous, bounded, and sufficiently smooth.

Bayesian interpretation



The (physical) model gives $\pi(y|x)$, the *conditional probability of observing y given x*. However, to predict, control, optimise or quantify uncertainty, the interest is often really in $\pi(x|y)$, the *conditional probability of possible causes x given the observed data y* – the **inverse problem**:

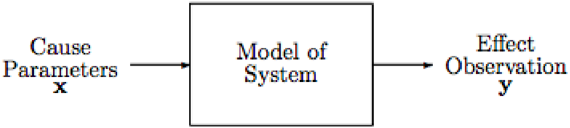
Bayesian interpretation



The (physical) model gives $\pi(y|x)$, the *conditional probability of observing y given x*. However, to predict, control, optimise or quantify uncertainty, the interest is often really in $\pi(x|y)$, the *conditional probability of possible causes x given the observed data y* – the **inverse problem**:

$$\pi_{\text{pos}}(x) := \underbrace{\pi(x|y) \propto \pi(y|x)\,\pi_{\text{pr}}(x)}_{\textbf{Bayes' rule}}$$

Bayesian interpretation



The (physical) model gives $\pi(y|x)$, the *conditional probability of observing y given x*. However, to predict, control, optimise or quantify uncertainty, the interest is often really in $\pi(x|y)$, the *conditional probability of possible causes x given the observed data y* – the **inverse problem**:

$$\pi_{\text{pos}}(x) := \underbrace{\pi(x|y) \propto \pi(y|x)\,\pi_{\text{pr}}(x)}_{\textbf{Bayes' rule}}$$

Extract information from π_{pos} *(means, covariances, event probabilities, predictions)* by evaluating **posterior expectations:**

$$\mathbb{E}_{\pi_{\text{pos}}}[h(x)] = \int h(x)\pi_{\text{pos}}(x)dx$$

Bayes' Rule and Classical Inversion

Classically [Hadamard, 1923]: Inverse map "F^{-1}" ($y \to x$) is typically ill-posed, i.e. lack of (a) **existence**, (b) **uniqueness** or (c) **boundedness**

Bayes' Rule and Classical Inversion

Classically [Hadamard, 1923]: Inverse map "F^{-1}" ($y \to x$) is typically ill-posed, i.e. lack of (a) **existence**, (b) **uniqueness** or (c) **boundedness**

- classical least squares solution $\hat{x}$ is *maximum likelihood estimate*
- prior distribution π_{pr} "acts" as regulariser – **well-posedness !**
- regularised least squares sol. is *maximum a posteriori (MAP) estimate*

Bayes' Rule and Classical Inversion

Classically [Hadamard, 1923]: Inverse map "F^{-1}" ($y \to x$) is typically ill-posed, i.e. lack of (a) **existence**, (b) **uniqueness** or (c) **boundedness**

- classical least squares solution $\hat{x}$ is *maximum likelihood estimate*
- prior distribution π_{pr} "acts" as regulariser – **well-posedness !**
- regularised least squares sol. is *maximum a posteriori (MAP) estimate*

However, in the Bayesian setting, the **full posterior** π_{pos} **contains more information** than the MAP estimator alone, e.g. the posterior covariance matrix reveals components of x that are (relatively) more or less certain.

Bayes' Rule and Classical Inversion

Classically [Hadamard, 1923]: Inverse map "F^{-1}" ($y \to x$) is typically ill-posed, i.e. lack of (a) **existence**, (b) **uniqueness** or (c) **boundedness**

- classical least squares solution $\hat{x}$ is *maximum likelihood estimate*
- prior distribution π_{pr} "acts" as regulariser – **well-posedness !**
- regularised least squares sol. is *maximum a posteriori (MAP) estimate*

However, in the Bayesian setting, the **full posterior** π_{pos} **contains more information** than the MAP estimator alone, e.g. the posterior covariance matrix reveals components of x that are (relatively) more or less certain.

Challenges: high dimension, expensive likelihood & the (inaccessible) normalising constant

$$\pi(y) := \int \pi(y|x)\, \pi_{\text{pr}}(x)\, \mathrm{d}x$$

Require **sample-based** approach to break **"Curse of Dimensionality"**.

Traditional Work Horse: Markov Chain Monte Carlo

ALGORITHM 1 (Metropolis-Hastings Markov Chain Monte Carlo)

- Choose initial state $x^0 \in X$.
- At state x^n generate proposal $x' \in X$ from distribution $q(x' \mid x^n)$ e.g. via a random walk: $x' \sim \mathrm{N}(x^n, \varepsilon^2 \mathrm{I})$
- Accept x' as a sample with probability

$$\alpha(x'|x^n) \;=\; \min\left(1, \frac{\pi(x'|y)\, q(x^n \mid y)}{\pi(x^n|x')\, q(x' \mid x^n)}\right)$$

i.e. $x^{n+1} = x'$ with probability $\alpha(x'|x^n)$; otherwise $x^{n+1} = x^n$.

Traditional Work Horse: Markov Chain Monte Carlo

ALGORITHM 1 (Metropolis-Hastings Markov Chain Monte Carlo)

- Choose initial state $x^0 \in X$.
- At state x^n generate proposal $x' \in X$ from distribution $q(x' \,|\, x^n)$ e.g. via a random walk: $x' \sim \mathrm{N}(x^n, \varepsilon^2 \mathrm{I})$
- Accept x' as a sample with probability

$$\alpha(x'|x^n) \;=\; \min\left(1, \frac{\pi(x'|y)\, q(x^n \,|\, y)}{\pi(x^n|x')\, q(x' \,|\, x^n)}\right)$$

i.e. $x^{n+1} = x'$ with probability $\alpha(x'|x^n)$; otherwise $x^{n+1} = x^n$.

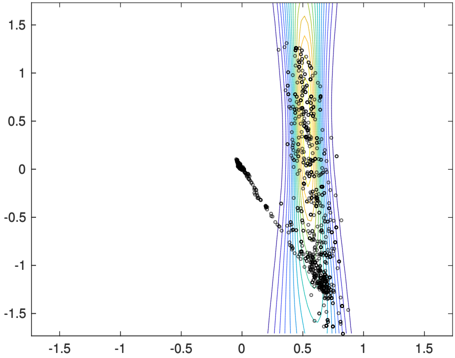
The samples $h(x^n)$ of some output function ("statistic") $h(\cdot)$ can be used for inference as usual – even though not i.i.d.:

$$\mathbb{E}_{\pi(x|y)}\left[h(x)\right] \;\approx\; \frac{1}{N}\sum_{i=1}^{N} h(x^n) := \widehat{h}^{\mathrm{MetH}}$$

Slow Convergence of Random Walk Metropolis-Hastings

But sampling with Metropolis-Hastings can be very inefficient ...

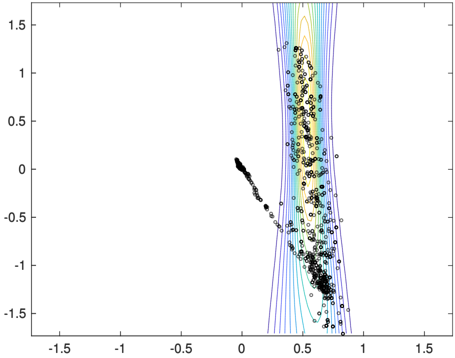
(due to burn-in, small step size and large number of rejections)



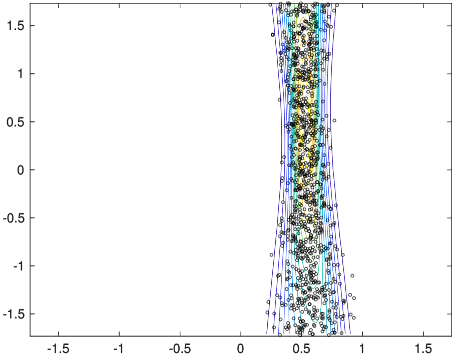
Slow Convergence of Random Walk Metropolis-Hastings

But sampling with Metropolis-Hastings can be very inefficient ...

(due to burn-in, small step size and large number of rejections)



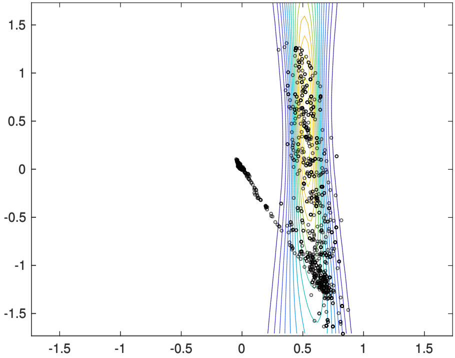
... not like this ...



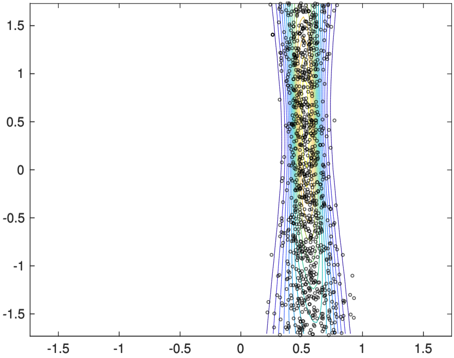
Slow Convergence of Random Walk Metropolis-Hastings

But sampling with Metropolis-Hastings can be very inefficient ...

(due to burn-in, small step size and large number of rejections)



... not like this ...



... **on top of** the slow Monte Carlo convergence rate of $O(N^{-1/2})$ **!**

Variational Bayes (as opposed to Metropolis-Hastings MCMC)

Aim to characterise the posterior distribution (density π_{pos}) **analytically** (at least approximately) for more efficient inference.

Variational Bayes (as opposed to Metropolis-Hastings MCMC)

Aim to characterise the posterior distribution (density π_{pos}) **analytically** (at least approximately) for more efficient inference.

This is a **challenging task** since:

- $x \in \mathbb{R}^d$ is typically **high-dimensional** (e.g., discretised function)
- π_{pos} is in general **non-Gaussian**
  (even if π_{pr} and observational noise are Gaussian)
- evaluations of likelihood may be **expensive** (e.g., solution of a PDE)

Variational Bayes (as opposed to Metropolis-Hastings MCMC)

Aim to characterise the posterior distribution (density π_{pos}) **analytically** (at least approximately) for more efficient inference.

This is a **challenging task** since:

- $x \in \mathbb{R}^d$ is typically **high-dimensional** (e.g., discretised function)
- π_{pos} is in general **non-Gaussian**
  (even if π_{pr} and observational noise are Gaussian)
- evaluations of likelihood may be **expensive** (e.g., solution of a PDE)

Key Tools

Transport Maps, **Optimisation**, Principle Component Analysis, Model Order Reduction, Hierarchies, Sparsity, Low Rank Approximation

Variational Bayes (as opposed to Metropolis-Hastings MCMC)

Aim to characterise the posterior distribution (density π_{pos}) **analytically** (at least approximately) for more efficient inference.

This is a **challenging task** since:

- $x \in \mathbb{R}^d$ is typically **high-dimensional** (e.g., discretised function)
- π_{pos} is in general **non-Gaussian**
  (even if π_{pr} and observational noise are Gaussian)
- evaluations of likelihood may be **expensive** (e.g., solution of a PDE)

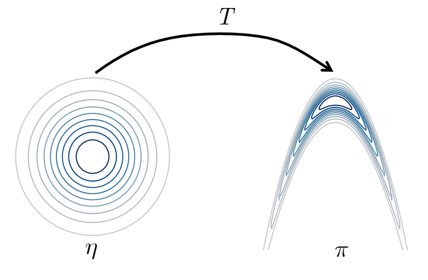
Key Tools – a playground for a numerical analyst!

Transport Maps, **Optimisation**, Principle Component Analysis, Model Order Reduction, Hierarchies, Sparsity, Low Rank Approximation

Deterministic Couplings of Probability Measures

Core idea [Moselhy, Marzouk, 2012]

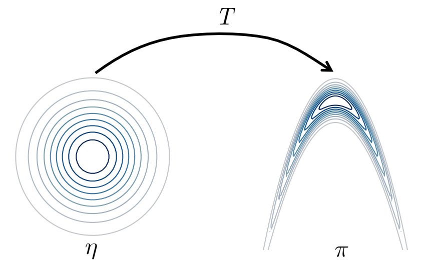
- Choose a *reference distribution* η (e.g., standard Gaussian)
- Seek **transport map** $T : \mathbb{R}^d \to \mathbb{R}^d$ such that $T_\sharp \eta = \pi$ (push-forward) (invertible)



Deterministic Couplings of Probability Measures

Core idea [Moselhy, Marzouk, 2012]

- Choose a *reference distribution* η (e.g., standard Gaussian)
- Seek **transport map** $T : \mathbb{R}^d \to \mathbb{R}^d$ such that $T_\sharp \eta = \pi$ (push-forward) (invertible)

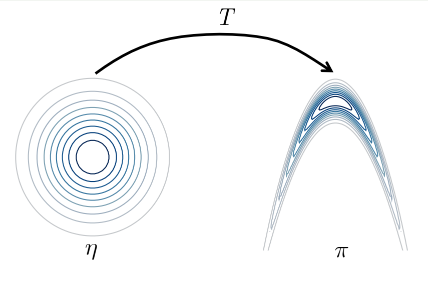


- In principle, enables *exact* (independent, unweighted) sampling!

Deterministic Couplings of Probability Measures

Core idea [Moselhy, Marzouk, 2012]

- Choose a *reference distribution* η (e.g., standard Gaussian)
- Seek **transport map** $T : \mathbb{R}^d \to \mathbb{R}^d$ such that $T_\sharp \eta = \pi$ (push-forward) (invertible)



- In principle, enables *exact* (independent, unweighted) sampling!
- **Approximately** satisfying conditions still useful: **Preconditioning!**

Variational Inference

- **Goal:** Sampling from target density $\pi(x)$

Variational Inference

- **Goal:** Sampling from target density $\pi(x)$
- Given a reference density η, find an invertible map $\hat{T}$ such that

$$\hat{T} := \operatorname*{argmin}_{T} \; \mathscr{D}_{\mathsf{KL}}(T_\sharp \, \eta \,\|\, \pi) = \operatorname*{argmin}_{T} \; \mathscr{D}_{\mathsf{KL}}(\eta \,\|\, T_\sharp^{-1} \pi)$$

where

$$\mathscr{D}_{\mathsf{KL}}(p \,\|\, q) := \int \log\left(\frac{p(x)}{q(x)}\right) p(x)\,\mathrm{d}x \quad \ldots \quad \textbf{Kullback-Leibler divergence}$$

$$T_\sharp \, p(x) := p\left(T^{-1}(x)\right) \left|\det\left(\nabla_x T^{-1}(x)\right)\right| \quad \ldots \quad \textbf{push-forward} \text{ of } p$$

Variational Inference

- **Goal:** Sampling from target density $\pi(x)$
- Given a reference density η, find an invertible map $\hat{T}$ such that

$$\hat{T} := \operatorname*{argmin}_{T} \mathscr{D}_{\mathrm{KL}}(T_\sharp\, \eta \,\|\, \pi) = \operatorname*{argmin}_{T} \mathscr{D}_{\mathrm{KL}}(\eta \,\|\, T_\sharp^{-1}\pi)$$

where

$$\mathscr{D}_{\mathrm{KL}}(p \,\|\, q) := \int \log\left(\frac{p(x)}{q(x)}\right) p(x)\, \mathrm{d}x \quad \dots \quad \textbf{Kullback-Leibler divergence}$$

$$T_\sharp\, p(x) := p\left(T^{-1}(x)\right) \left|\det\left(\nabla_x T^{-1}(x)\right)\right| \quad \dots \quad \textbf{push-forward} \text{ of } p$$

- Advantage of using $\mathscr{D}_{\mathrm{KL}}$: normalising constant for π is **not** needed

Variational Inference

- **Goal:** Sampling from target density $\pi(x)$
- Given a reference density η, find an invertible map $\hat{T}$ such that

$$\hat{T} := \underset{T}{\operatorname{argmin}} \ \mathscr{D}_{\mathrm{KL}}(T_\sharp \eta \,\|\, \pi) = \underset{T}{\operatorname{argmin}} \ \mathscr{D}_{\mathrm{KL}}(\eta \,\|\, T_\sharp^{-1}\pi)$$

  where

$$\mathscr{D}_{\mathrm{KL}}(p \,\|\, q) := \int \log\left(\frac{p(x)}{q(x)}\right) p(x)\,\mathrm{d}x \quad \ldots \quad \textbf{Kullback-Leibler divergence}$$

$$T_\sharp\, p(x) := p\left(T^{-1}(x)\right) \left|\det\left(\nabla_x T^{-1}(x)\right)\right| \quad \ldots \quad \textbf{push-forward} \text{ of } p$$

- Advantage of using $\mathscr{D}_{\mathrm{KL}}$: normalising constant for π is **not** needed
- Minimise over some **suitable class** $\mathscr{T}$ of maps T
  (where ideally Jacobian determinant $\det\left(\nabla_x T^{-1}(x)\right)$ is easy to evaluate)

Variational Inference

- **Goal:** Sampling from target density $\pi(x)$
- Given a reference density η, find an invertible map $\hat{T}$ such that

$$\hat{T} := \operatorname*{argmin}_{T} \mathscr{D}_{\mathrm{KL}}(T_\sharp \eta \,\|\, \pi) = \operatorname*{argmin}_{T} \mathscr{D}_{\mathrm{KL}}(\eta \,\|\, T_\sharp^{-1}\pi)$$

where

$$\mathscr{D}_{\mathrm{KL}}(p \,\|\, q) := \int \log\left(\frac{p(x)}{q(x)}\right) p(x)\,\mathrm{d}x \quad \ldots \quad \textbf{Kullback-Leibler divergence}$$

$$T_\sharp\, p(x) := p\left(T^{-1}(x)\right) \left|\det\left(\nabla_x T^{-1}(x)\right)\right| \quad \ldots \quad \textbf{push-forward} \text{ of } p$$

- Advantage of using $\mathscr{D}_{\mathrm{KL}}$: normalising constant for π is **not** needed
- Minimise over some **suitable class** $\mathscr{T}$ of maps T
  (where ideally Jacobian determinant $\det\left(\nabla_x T^{-1}(x)\right)$ is easy to evaluate)
- **To improve:** **enrich** class $\mathscr{T}$ or use samples of $T_\sharp^{-1}\pi$ as **proposals for MCMC** or in **importance sampling** (see below)

Many Choices ("Architectures") for $\mathscr{T}$ possible

Examples: (list not comprehensive!!)

1. Optimal Transport or Knothe-Rosenblatt Rearrangement
   [Moselhy, Marzouk, 2012], [Marzouk, Moselhy, Parno, Spantini, 2016]
2. Normalizing or Autoregressive Flows [Rezende, Mohamed, 2015]
   (and related methods in the ML literature)

Many Choices ("Architectures") for $\mathscr{T}$ possible

Examples: (list not comprehensive!!)

1. Optimal Transport or Knothe-Rosenblatt Rearrangement
   [Moselhy, Marzouk, 2012], [Marzouk, Moselhy, Parno, Spantini, 2016]
2. Normalizing or Autoregressive Flows [Rezende, Mohamed, 2015]
   (and related methods in the ML literature)
3. Kernel-based variational inference: Stein Variational Methods
   [Liu, Wang, 2016], [Detommaso, Cui, Spantini, Marzouk, **RS**, 2018],
   [Chen, Wu, Chen, O'Leary-Roseberry, Ghattas, 2019] $\boxed{\text{not today!}}$
4. Layers of low-rank maps [Bigoni, Zahm, Spantini, Marzouk, arXiv 2019]
5. Layers of hierarchical invertible neural networks (HINT) $\boxed{\text{not today!}}$
   [Detommaso, Kruse, Ardizzone, Rother, Köthe, **RS**, arXiv:1905.10687]

Many Choices ("Architectures") for $\mathscr{T}$ possible

Examples: (list not comprehensive!!)

1. Optimal Transport or Knothe-Rosenblatt Rearrangement
   [Moselhy, Marzouk, 2012], [Marzouk, Moselhy, Parno, Spantini, 2016]
2. Normalizing or Autoregressive Flows [Rezende, Mohamed, 2015]
   (and related methods in the ML literature)
3. Kernel-based variational inference: Stein Variational Methods
   [Liu, Wang, 2016], [Detommaso, Cui, Spantini, Marzouk, **RS**, 2018], [Chen, Wu, Chen, O'Leary-Roseberry, Ghattas, 2019] not today!
4. Layers of low-rank maps [Bigoni, Zahm, Spantini, Marzouk, arXiv 2019]
5. Layers of hierarchical invertible neural networks (HINT) not today!
   [Detommaso, Kruse, Ardizzone, Rother, Köthe, **RS**, arXiv:1905.10687]
6. **Low-rank tensor approximation** of Knothe-Rosenblatt rearrangement
   [Dolgov, Anaya-Izquierdo, Fox, RS, 2019]

Approximation and Sampling of Multivariate Probability Distributions in the Tensor Train Decomposition

[Dolgov, Anaya-Izquierdo, Fox, RS, 2019]

Variational Inference with Triangular Maps

- In general, in **Variational Inference** aim to find

$$\underset{T}{\operatorname{argmin}}\, \mathscr{D}_{\mathsf{KL}}(\, T_\sharp\, \eta \,||\, \pi)$$

- Note:

$$\mathscr{D}_{\mathsf{KL}}(\, T_\sharp\, \eta \,||\, \pi) = -\mathbb{E}_{\boldsymbol{u}\sim\eta}\Big[\log \pi(\boldsymbol{T}(\boldsymbol{u})) + \log|\det \nabla \boldsymbol{T}(\boldsymbol{u})|\Big] + \text{const}$$

Variational Inference with Triangular Maps

- In general, in **Variational Inference** aim to find

$$\underset{T}{\operatorname{argmin}}\, \mathscr{D}_{\mathrm{KL}}(T_\sharp\, \eta \,||\, \pi)$$

- Note:

$$\mathscr{D}_{\mathrm{KL}}(T_\sharp\, \eta \,||\, \pi) = -\mathbb{E}_{\boldsymbol{u}\sim\eta}\Big[\log \pi(\boldsymbol{T}(\boldsymbol{u})) + \log|\det \nabla \boldsymbol{T}(\boldsymbol{u})|\Big] + \mathrm{const}$$

- Particularly useful family $\mathscr{T}$ are **Knothe-Rosenblatt triangular rearrangements** (see [Marzouk, Moshely, Parno, Spantini, 2016]):

$$T(x) = \begin{bmatrix} T_1(x_1) \\ T_2(x_1, x_2) \\ \vdots \\ T_d(x_1, x_2, \ldots, x_d) \end{bmatrix} \qquad (= \text{autoregressive flow in ML})$$

Then: $\log|\det \nabla \boldsymbol{T}(\boldsymbol{u})| = \sum_k \log \partial_{x_k} T^k$

Knothe-Rosenblatt via Conditional Distribution Sampling

In fact, $\exists!$ **triangular map** satisfying $T_\sharp \eta = \pi$ (for abs. cont. η, π on $\mathbb{R}^d$)

Conditional Distribution Sampling [Rosenblatt '52] (explicitly available!)

Knothe-Rosenblatt via Conditional Distribution Sampling

In fact, $\exists!$ **triangular map** satisfying $T_\sharp \eta = \pi$ (for abs. cont. η, π on $\mathbb{R}^d$)

Conditional Distribution Sampling [Rosenblatt '52] (explicitly available!)

- Any density factorises into product of conditional densities:

$$\pi(x_1, \dots, x_d) = \pi_1(x_1)\pi_2(x_2|x_1) \cdots \pi_d(x_d|x_1, \dots, x_{d-1})$$

- Can sample (up to normalisation with known scaling factor)

$$x_k \sim \pi_k(x_k|x_1, \dots, x_{k-1}) \sim \int \pi(x_1, \dots, x_d) dx_{k+1} \cdots dx_d$$

Knothe-Rosenblatt via Conditional Distribution Sampling

In fact, $\exists!$ **triangular map** satisfying $T_\sharp \eta = \pi$ (for abs. cont. η, π on $\mathbb{R}^d$)

Conditional Distribution Sampling [Rosenblatt '52] (explicitly available!)

- Any density factorises into product of conditional densities:

$$\pi(x_1, \ldots, x_d) = \pi_1(x_1)\pi_2(x_2|x_1) \cdots \pi_d(x_d|x_1, \ldots, x_{d-1})$$

- 1st step: Produce sample x_1^i via **1D CDF-inversion** from

$$\pi_1(x_1) \sim \int \pi(x_1, x_2, \ldots, x_d) dx_2 \cdots dx_d$$

Knothe-Rosenblatt via Conditional Distribution Sampling

In fact, $\exists!$ **triangular map** satisfying $T_\sharp \eta = \pi$ (for abs. cont. η, π on $\mathbb{R}^d$)

Conditional Distribution Sampling [Rosenblatt '52] (explicitly available!)

- Any density factorises into product of conditional densities:

$$\pi(x_1, \ldots, x_d) = \pi_1(x_1)\pi_2(x_2|x_1)\cdots\pi_d(x_d|x_1, \ldots, x_{d-1})$$

- 1st step: Produce sample x_1^i via **1D CDF-inversion** from

$$\pi_1(x_1) \sim \int \pi(x_1, x_2, \ldots, x_d)dx_2 \cdots dx_d$$

- k-th step: Given $x_1^i, \ldots, x_{k-1}^i$ sample x_k^i via **1D CDF-inversion** from

$$\pi_k(x_k|x_1^i, \ldots, x_{k-1}^i) \sim \int \pi(x_1^i, \ldots, x_{k-1}^i, x_k, x_{k+1}, \ldots, x_d)dx_{k+1} \cdots dx_d$$

Knothe-Rosenblatt via Conditional Distribution Sampling

In fact, $\exists!$ **triangular map** satisfying $T_\sharp \eta = \pi$ (for abs. cont. η, π on $\mathbb{R}^d$)

Conditional Distribution Sampling [Rosenblatt '52] (explicitly available!)

- Any density factorises into product of conditional densities:

$$\pi(x_1, \ldots, x_d) = \pi_1(x_1)\pi_2(x_2|x_1) \cdots \pi_d(x_d|x_1, \ldots, x_{d-1})$$

- 1st step: Produce sample x_1^i via **1D CDF-inversion** from

$$\pi_1(x_1) \sim \int \pi(x_1, x_2, \ldots, x_d) dx_2 \cdots dx_d$$

- k-th step: Given $x_1^i, \ldots, x_{k-1}^i$ sample x_k^i via **1D CDF-inversion** from

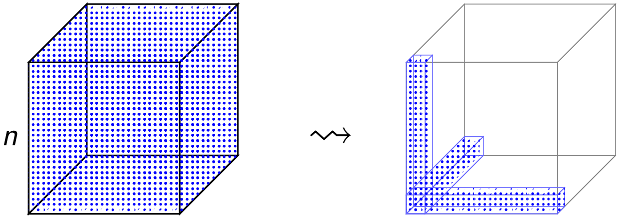
$$\pi_k(x_k|x_1^i, \ldots, x_{k-1}^i) \sim \int \pi(x_1^i, \ldots, x_{k-1}^i, x_k, x_{k+1}, \ldots, x_d) dx_{k+1} \cdots dx_d$$

Problem: $(d-k)$**-dimensional integration** at k-th step!

Remedy: Find approximation $\tilde{\pi} \approx \pi$ where integration is cheap!

Low-rank Tensor Approximation of Distributions

Low-rank tensor decomposition $\Leftrightarrow$ separation of variables:



$\mathcal{O}(n^d)$ $\rightsquigarrow$ $\mathcal{O}(dn)$

- Tensor grid with n points per direction (or n polynomial basis fcts.)
- Approximate: $\underbrace{\pi(x_1, \ldots, x_d)}_{\text{tensor}} \approx \underbrace{\sum_{|\alpha| \leq r} \pi^1_\alpha(x_1)\pi^2_\alpha(x_2) \cdots \pi^d_\alpha(x_d)}_{\text{tensor product decomposition}}$
- Many low-rank tensor formats exist [Kolda, Bader '09], [Hackbusch '12]

Conditional Distribution Sampler (with factorised distribution)

For the low-rank tensor approximation

$$\pi(x) \;\approx\; \tilde{\pi}(x) = \sum_{|\alpha| \le r} \pi_\alpha^1(x_1) \cdot \pi_\alpha^2(x_2) \cdots \pi_\alpha^d(x_d)$$

the k-th step of the CD sampler, given $x_1^i, \ldots, x_{k-1}^i$, simplifies to

$$\tilde{\pi}_k(x_k | x_1^i, \ldots, x_{k-1}^i) \sim \sum_{|\alpha| \le r} \pi_\alpha^1(x_1^i) \cdots \pi_\alpha^{k-1}(x_{k-1}^i) \ldots$$
$$\ldots \pi_\alpha^k(x_k) \ldots$$
$$\ldots \underbrace{\int \pi_\alpha^{k+1}(x_{k+1}) dx_{k+1} \cdots \int \pi_\alpha^d(x_d) dx_d}_{\text{Repeated 1D integrals!}}$$

$\boxed{\textbf{linear in } d}$

Conditional Distribution Sampler (with factorised distribution)

For the low-rank tensor approximation

$$\pi(x) \;\approx\; \tilde{\pi}(x) = \sum_{|\alpha|\le r} \pi_\alpha^1(x_1)\cdot\pi_\alpha^2(x_2)\cdots\pi_\alpha^d(x_d)$$

the k-th step of the CD sampler, given $x_1^i,\dots,x_{k-1}^i$, simplifies to

$$\tilde{\pi}_k(x_k|x_1^i,\dots,x_{k-1}^i) \sim \sum_{|\alpha|\le r} \pi_\alpha^1(x_1^i)\cdots\pi_\alpha^{k-1}(x_{k-1}^i)\dots$$
$$\dots\pi_\alpha^k(x_k)\dots$$
$$\dots\underbrace{\int \pi_\alpha^{k+1}(x_{k+1})dx_{k+1}\cdots\int\pi_\alpha^d(x_d)dx_d}_{\text{Repeated 1D integrals!}}$$

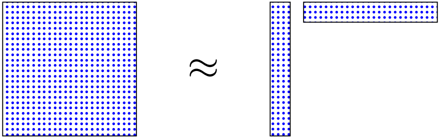
linear in d

To sample (in each step): **Simple 1D CDF-inversions** **linear in** d

Low-rank Decomposition (Two Variables)

Collect discretised values of $\pi(\theta_1, \theta_2)$ on $n \times n$ grid into a matrix:

$$P(i,j) = \sum_{\alpha=1}^{r} V_\alpha(i) W_\alpha(j) + \mathcal{O}(\varepsilon)$$

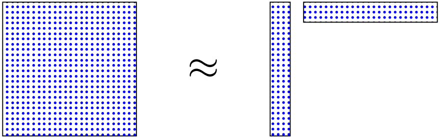


- **Rank** $r \ll n$ (exploiting structure, smoothness, ...)
- $\mathtt{mem}(V) + \mathtt{mem}(W) = 2nr \ll n^2 = \mathtt{mem}(P)$
- **SVD** provides optimal ε for fixed r (s.t. $\min_{V,W} \|P - VW^*\|_F^2$)
- **But** requires **all** n^2 entries of P **!**

Low-rank Decomposition (Two Variables)

Collect discretised values of $\pi(\theta_1, \theta_2)$ on $n \times n$ grid into a matrix:

$$P(i,j) = \sum_{\alpha=1}^{r} V_\alpha(i) W_\alpha(j) + \mathcal{O}(\varepsilon)$$

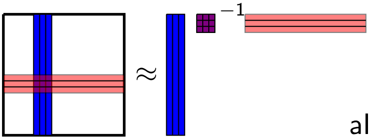


- **Rank** $r \ll n$ (exploiting structure, smoothness, ...)
- $\mathtt{mem}(V) + \mathtt{mem}(W) = 2nr \ll n^2 = \mathtt{mem}(P)$
- **SVD** provides optimal ε for fixed r (s.t. $\min_{V,W} \|P - VW^*\|_F^2$)
- **But** requires **all** n^2 entries of P !

$\boxed{n^d \textbf{ in } d \textbf{ dimensions!}}$

Cross Algorithm (construct low-rank factorisation from few entries)

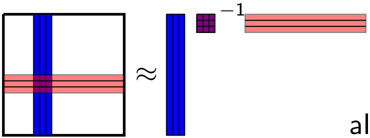
- Interpolation arguments show: for some suitable **index sets** $\mathscr{I}, \mathscr{J} \subset \{1, \ldots, n\}$ with $|\mathscr{I}| = |\mathscr{J}| = r$, the **cross** decomposition



also $P(:, \mathscr{J})P^{-1}(\mathscr{I}, \mathscr{J})P(\mathscr{I}, :) \approx P$

Cross Algorithm (construct low-rank factorisation from few entries)

- Interpolation arguments show: for some suitable **index sets** $\mathscr{I}, \mathscr{J} \subset \{1, \ldots, n\}$ with $|\mathscr{I}| = |\mathscr{J}| = r$, the **cross** decomposition



also $P(:, \mathscr{J})P^{-1}(\mathscr{I}, \mathscr{J})P(\mathscr{I}, :) \approx P$

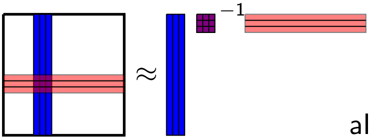
- **Maxvol** principle gives **'best'** indices $\mathscr{I}, \mathscr{J}$ [Goreinov, Tyrtyshnikov '01]

$$|\det P(\mathscr{I}, \mathscr{J})| = \max_{\hat{\mathscr{I}}, \hat{\mathscr{J}}} \left|\det P(\hat{\mathscr{I}}, \hat{\mathscr{J}})\right| \;\Rightarrow\; \|P - \tilde{P}\|_C \le (r+1) \min_{\mathrm{rank}\hat{P} = r} \|P - \hat{P}\|_2$$

(NP-hard)

Cross Algorithm (construct low-rank factorisation from few entries)

- Interpolation arguments show: for some suitable **index sets** $\mathscr{I}, \mathscr{J} \subset \{1, \ldots, n\}$ with $|\mathscr{I}| = |\mathscr{J}| = r$, the **cross** decomposition



also $P(:, \mathscr{J}) P^{-1}(\mathscr{I}, \mathscr{J}) P(\mathscr{I}, :) \approx P$

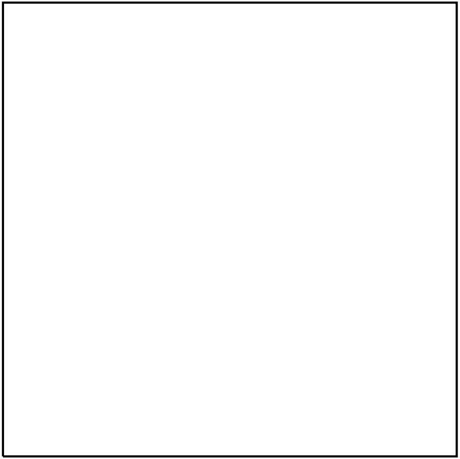
- **Maxvol** principle gives **'best'** indices $\mathscr{I}, \mathscr{J}$ [Goreinov, Tyrtyshnikov '01]

$$|\det P(\mathscr{I}, \mathscr{J})| = \max_{\hat{\mathscr{I}}, \hat{\mathscr{J}}} \left|\det P(\hat{\mathscr{I}}, \hat{\mathscr{J}})\right| \Rightarrow \|P - \tilde{P}\|_C \leq (r+1) \min_{\mathrm{rank}\hat{P}=r} \|P - \hat{P}\|_2$$

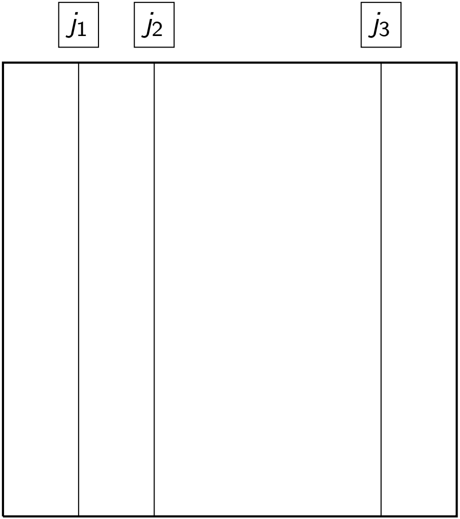
(NP-hard)

- But how can we find good sets $\mathscr{I}, \mathscr{J}$ **in practice**?
- And how can we generalise this to $d > 2$?

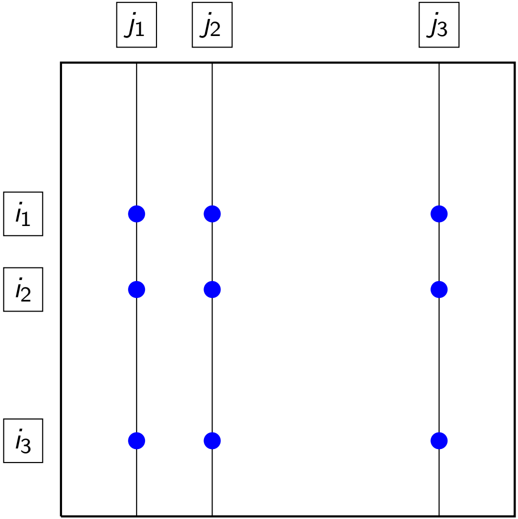
Alternating Iteration (for cross approximation)



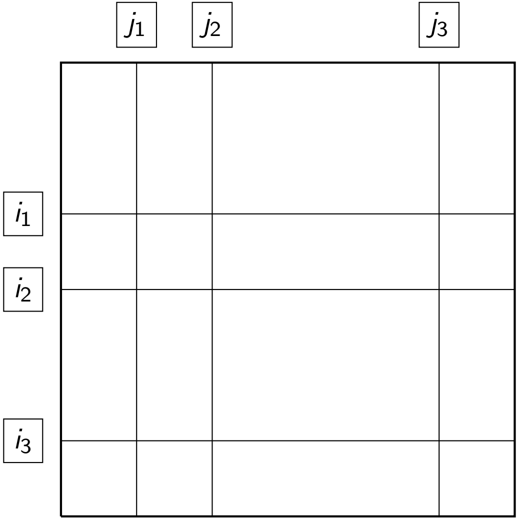
Alternating Iteration (for cross approximation)



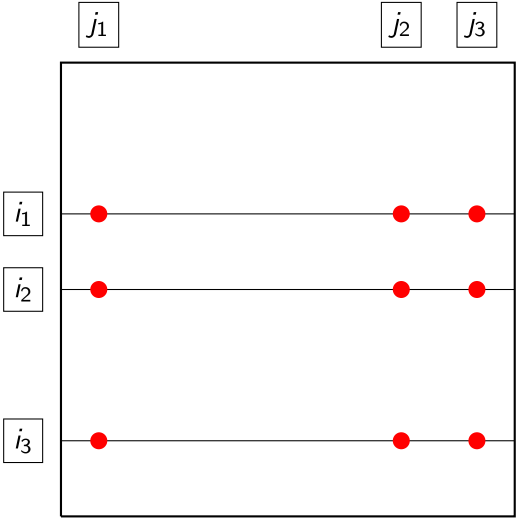
Alternating Iteration (for cross approximation)



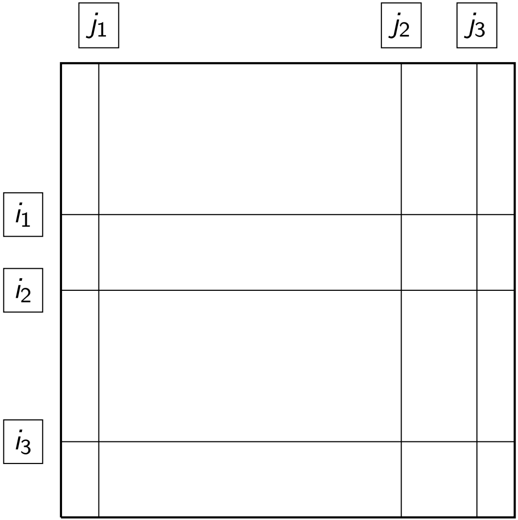
Alternating Iteration (for cross approximation)



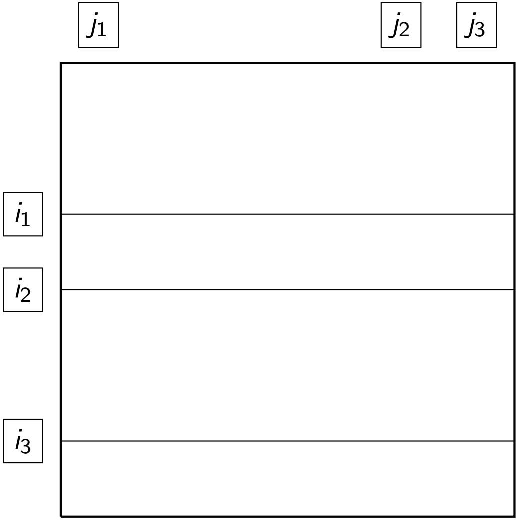
Alternating Iteration (for cross approximation)



Alternating Iteration (for cross approximation)



Alternating Iteration (for cross approximation)

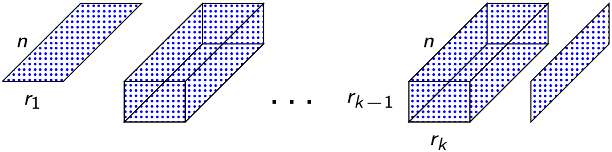


- **Practically realizable** strategy (with $\mathcal{O}(2nr)$ samples & $\mathcal{O}(nr^2)$ flops).
- For numerical stability use **rank-revealing QR** in practice.
- To **adapt rank** expand $V \to [V \;\; Z]$) (with residual Z)
- Several similar algorithms exist: e.g. ACA [Bebendorf '00] or EIM [Barrault et al '04]

Tensor Train (TT) Decomposition (Many Variables)

- Many variables: **Matrix Product States/Tensor Train**

$$\pi(i_1 \dots i_d) = \sum_{\substack{\alpha_k=1 \\ 0<k<d}}^{r_k} \pi^1_{\alpha_1}(i_1) \cdot \pi^2_{\alpha_1,\alpha_2}(i_2) \cdot \pi^3_{\alpha_2,\alpha_3}(i_3) \cdots \pi^d_{\alpha_{d-1}}(i_d)$$

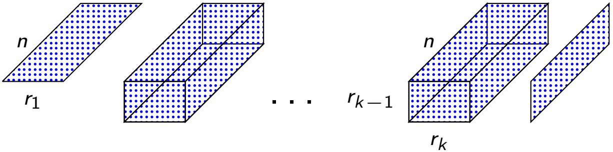


[Wilson '75] (comput. physics), [White '93], [Verstraete '04]; [Oseledets '09]

Tensor Train (TT) Decomposition (Many Variables)

- Many variables: **Matrix Product States/Tensor Train**

$$\pi(i_1 \dots i_d) = \sum_{\substack{\alpha_k=1 \\ 0<k<d}}^{r_k} \pi^1_{\alpha_1}(i_1) \cdot \pi^2_{\alpha_1,\alpha_2}(i_2) \cdot \pi^3_{\alpha_2,\alpha_3}(i_3) \cdots \pi^d_{\alpha_{d-1}}(i_d)$$



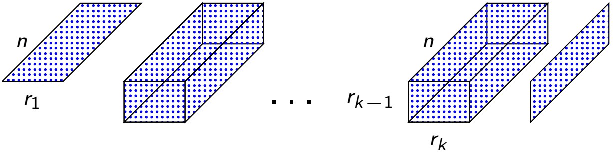
[Wilson '75] (comput. physics), [White '93], [Verstraete '04]; [Oseledets '09]

- TT blocks π^k are **three-dimensional** $r_{k-1} \times n \times r_k$ tensors
- with **TT ranks** $r_1, \dots, r_{d-1} \le r$

Tensor Train (TT) Decomposition (Many Variables)

- Many variables: **Matrix Product States/Tensor Train**

$$\pi(i_1 \dots i_d) = \sum_{\substack{\alpha_k=1 \\ 0<k<d}}^{r_k} \pi^1_{\alpha_1}(i_1) \cdot \pi^2_{\alpha_1,\alpha_2}(i_2) \cdot \pi^3_{\alpha_2,\alpha_3}(i_3) \cdots \pi^d_{\alpha_{d-1}}(i_d)$$



[Wilson '75] (comput. physics), [White '93], [Verstraete '04]; [Oseledets '09]

- TT blocks π^k are **three-dimensional** $r_{k-1} \times n \times r_k$ tensors
- with **TT ranks** $r_1, \dots, r_{d-1} \leq r$
- Storage: $\mathcal{O}(dnr^2)$

linear in d

TT Cross – An Efficient Computation of a TT Decomposition

Given random initial sets $\mathscr{J}_0, \ldots, \mathscr{J}_d$ iterate: [Oseledets, Tyrtyshnikov '10]

1. Update kth TT block: $\pi^k(i_k) = \pi(\mathscr{I}_{k-1}, i_k, \mathscr{J}_k)$
2. Update kth index set: $\mathscr{I}_k = \texttt{pivots}_{\textbf{row}}(\pi^k)$

   (using **maxvol** principle on different **matrizations** of tensor in each step)

TT Cross – An Efficient Computation of a TT Decomposition

Given random initial sets $\mathscr{J}_0, \ldots, \mathscr{J}_d$ iterate: [Oseledets, Tyrtyshnikov '10]

1. Update kth TT block: $\pi^k(\underline{\mathbf{i_k}}) = \pi(\mathscr{I}_{k-1}, \underline{\mathbf{i_k}}, \mathscr{J}_k)$
2. Update kth index set: $\mathscr{I}_k = \texttt{pivots}_{\textbf{row}}(\pi^k)$

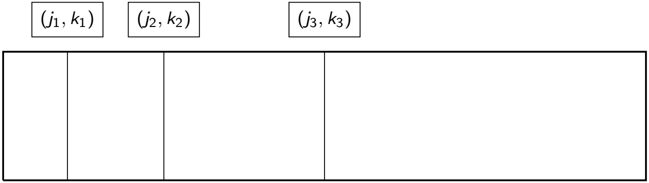
   (using **maxvol** principle on different **matrizations** of tensor in each step)

TT Cross – An Efficient Computation of a TT Decomposition

Given random initial sets $\mathscr{J}_0, \dots, \mathscr{J}_d$ iterate: [Oseledets, Tyrtyshnikov '10]

1. Update kth TT block: $\pi^k() = \pi(\mathscr{I}_{k-1}, , \mathscr{J}_k)$
2. Update kth index set: $\mathscr{I}_k = \texttt{pivots}_{\textbf{row}}(\pi^k)$

   (using **maxvol** principle on different **matrizations** of tensor in each step)

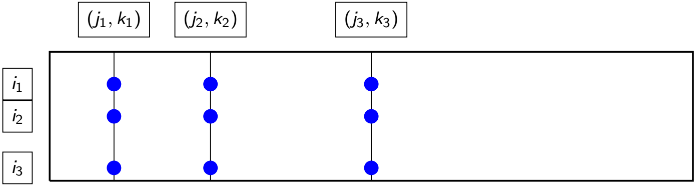


TT Cross – An Efficient Computation of a TT Decomposition

Given random initial sets $\mathscr{J}_0, \ldots, \mathscr{J}_d$ iterate: [Oseledets, Tyrtyshnikov '10]

1. Update kth TT block: $\pi^k() = \pi(\mathscr{I}_{k-1}, , \mathscr{J}_k)$
2. Update kth index set: $\mathscr{I}_k = \texttt{pivots}_{\textbf{row}}(\pi^k)$

   (using **maxvol** principle on different **matrizations** of tensor in each step)

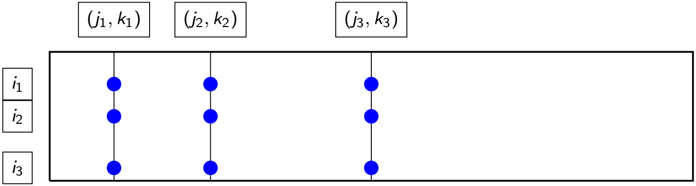


TT Cross – An Efficient Computation of a TT Decomposition

Given random initial sets $\mathscr{J}_0, \ldots, \mathscr{J}_d$ iterate: [Oseledets, Tyrtyshnikov '10]

1. Update kth TT block: $\pi^k() = \pi(\mathscr{I}_{k-1},, \mathscr{J}_k)$
2. Update kth index set: $\mathscr{I}_k = \texttt{pivots}_{\textbf{row}}(\pi^k)$

   (using **maxvol** principle on different **matrizations** of tensor in each step)



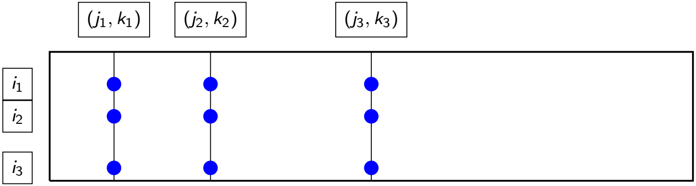
3. Set $k \to k+1$ and move to the next block.

TT Cross – An Efficient Computation of a TT Decomposition

Given random initial sets $\mathscr{J}_0, \ldots, \mathscr{J}_d$ iterate: [Oseledets, Tyrtyshnikov '10]

1. Update kth TT block: $\pi^k() = \pi(\mathscr{I}_{k-1}, , \mathscr{J}_k)$
2. Update kth index set: $\mathscr{I}_k = \texttt{pivots}_{\textbf{row}}(\pi^k)$

   (using **maxvol** principle on different **matrizations** of tensor in each step)



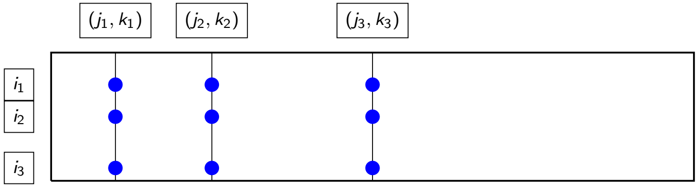
3. Set $k \to k + 1$ and move to the next block.
4. When $k = d$, switch direction and update index set $\mathscr{J}_{k-1}$.

TT Cross – An Efficient Computation of a TT Decomposition

Given random initial sets $\mathcal{J}_0, \ldots, \mathcal{J}_d$ iterate: [Oseledets, Tyrtyshnikov '10]

1. Update kth TT block: $\pi^k() = \pi(\mathcal{I}_{k-1},, \mathcal{J}_k)$
2. Update kth index set: $\mathcal{I}_k = \texttt{pivots}_{\textbf{row}}(\pi^k)$

   (using **maxvol** principle on different **matrizations** of tensor in each step)



3. Set $k \to k+1$ and move to the next block.
4. When $k = d$, switch direction and update index set $\mathcal{J}_{k-1}$.

Cost: $\mathcal{O}(dnr^2)$ samples & $\mathcal{O}(dnr^3)$ flops per iteration **linear** in d

Tensor Train (TT) Transport Maps (Summary & Comments)

[Dolgov, Anaya-Izquierdo, Fox, RS, 2019]

- Generic – not problem specific ('black box')
- Cross approximation: **'sequential'** design along **1D lines**
- Separable product form: $\tilde{\pi}(x_1, \ldots, x_d) = \sum_{|\alpha| \le r} \pi_\alpha^1(x_1) \ldots \pi_\alpha^d(x_d)$

  Cheap construction/storage & low # model evals $\boxed{\text{linear in } d}$

  Cheap integration w.r.t. x $\boxed{\text{linear in } d}$

  Cheap samples via **conditional distribution method** $\boxed{\text{linear in } d}$

Tensor Train (TT) Transport Maps (Summary & Comments)

[Dolgov, Anaya-Izquierdo, Fox, RS, 2019]

- Generic – not problem specific ('black box')
- Cross approximation: **'sequential'** design along **1D lines**
- Separable product form: $\tilde{\pi}(x_1, \ldots, x_d) = \sum_{|\alpha| \le r} \pi_\alpha^1(x_1) \ldots \pi_\alpha^d(x_d)$

  Cheap construction/storage & low # model evals | linear in d |

  Cheap integration w.r.t. x | linear in d |

  Cheap samples via **conditional distribution method** | linear in d |

- Tuneable approximation error ε (by adapting ranks r):

  $\Longrightarrow$ cost & storage **(poly)logarithmic in** ε | next slide |

Tensor Train (TT) Transport Maps (Summary & Comments)

[Dolgov, Anaya-Izquierdo, Fox, RS, 2019]

- Generic – not problem specific ('black box')
- Cross approximation: **'sequential'** design along **1D lines**
- Separable product form: $\tilde{\pi}(x_1, \ldots, x_d) = \sum_{|\alpha| \leq r} \pi^1_\alpha(x_1) \ldots \pi^d_\alpha(x_d)$

  Cheap construction/storage & low # model evals $\boxed{\text{linear in } d}$

  Cheap integration w.r.t. x $\boxed{\text{linear in } d}$

  Cheap samples via **conditional distribution method** $\boxed{\text{linear in } d}$

- Tuneable approximation error ε (by adapting ranks r):

  $\Longrightarrow$ cost & storage **(poly)logarithmic in** ε $\boxed{\text{next slide}}$

- Many known ways to use these samples for fast inference!

  (as proposals for MCMC, as control variates, importance weighting, ...)

Theoretical Result [Rohrbach, Dolgov, Grasedyck, RS, 2020]

For **Gaussian distributions** $\pi(x)$ we have the following result: Let

$$\pi : \mathbb{R}^d \to \mathbb{R}, \quad x \mapsto \exp\left(-\tfrac{1}{2} x^T \Sigma x\right)$$

and define

$$\Sigma := \begin{bmatrix} \Sigma_{11}^{(k)} & \Gamma_k^T \\ \Gamma_k & \Sigma_{22}^{(k)} \end{bmatrix} \quad \text{where} \quad \Gamma_k \in \mathbb{R}^{(d-k)\times k}.$$

Theorem. Let Σ be SPD with $\lambda_{\min} > 0$. Suppose $\rho := \max_k \operatorname{rank}(\Gamma_k)$ and $\sigma := \max_{k,i} \sigma_i^{(k)}$, where $\sigma_i^{(k)}$ are the singular values of Γ_k.

Then, for all $\varepsilon > 0$, there exists a TT-approximation $\tilde{\pi}_\varepsilon$ s.t.

$$\|\pi - \tilde{\pi}_\varepsilon\|_{L^2(\mathbb{R}^d)} \leq \varepsilon \|\pi\|_{L^2(\mathbb{R}^d)}$$

and the TT-ranks of $\tilde{\pi}_\varepsilon$ are bounded by

$$r \;\leq\; \left(\left(1 + 7\frac{\sigma}{\lambda_{\min}}\right) \log\left(7\rho\,\frac{d}{\varepsilon}\right)\right)^{\rho}. \qquad \text{(polylogarithmic growth)}$$

How to use the TT-CD sampler to estimate $\mathbb{E}_\pi Q$?

Problem: We are sampling from approximate $\tilde{\pi} = \pi + \mathscr{O}(\varepsilon)$.

How to use the TT-CD sampler to estimate $\mathbb{E}_\pi Q$?

Problem: We are sampling from approximate $\tilde{\pi} = \pi + \mathcal{O}(\varepsilon)$.

Option 0: **Classical variational inference**

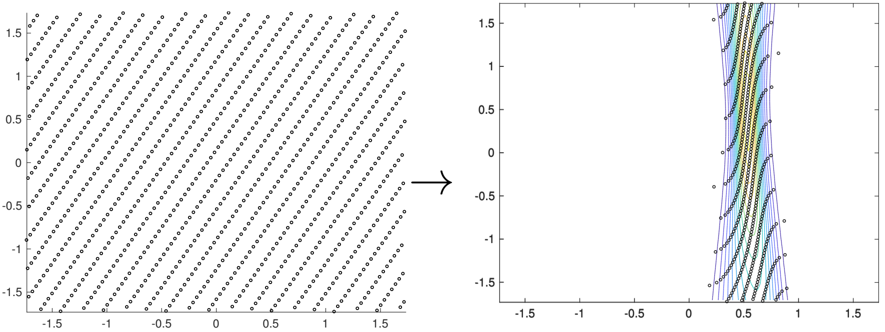
- **Explicit integration** (linear in d): get biased estimator $\mathbb{E}_{\tilde{\pi}} Q \approx \mathbb{E}_\pi Q$

How to use the TT-CD sampler to estimate $\mathbb{E}_\pi Q$?

Problem: We are sampling from approximate $\tilde{\pi} = \pi + \mathscr{O}(\varepsilon)$.

Option 0: **Classical variational inference**

- **Explicit integration** (linear in d): get biased estimator $\mathbb{E}_{\tilde{\pi}} Q \approx \mathbb{E}_\pi Q$
- Non-smooth Q: use Monte Carlo sampling, or **better**, *QMC 'seeds'*



2D projection of 11D map (problem specification below!)

Sampling from exact π: Unbiased estimates of $\mathbb{E}_\pi Q$

Option 1: Use $\{x^i_{\tilde{\pi}}\}$ as (i.i.d.) **proposals** in Metropolis-Hastings

- Accept proposal $x^i_{\tilde{\pi}}$ with probability $\alpha = \min\left(1, \frac{\pi(x^i_{\tilde{\pi}})\tilde{\pi}(x^{i-1}_{\pi})}{\pi(x^{i-1}_{\pi})\tilde{\pi}(x^i_{\tilde{\pi}})}\right)$
- Can prove that **rejection rate** $\sim \varepsilon$ and **IACT** $\tau \sim 1 + \varepsilon$

Sampling from exact π: Unbiased estimates of $\mathbb{E}_\pi Q$

Option 1: Use $\{x^i_{\tilde\pi}\}$ as (i.i.d.) **proposals** in Metropolis-Hastings

- Accept proposal $x^i_{\tilde\pi}$ with probability $\alpha = \min\left(1, \frac{\pi(x^i_{\tilde\pi})\tilde\pi(x^{i-1}_\pi)}{\pi(x^{i-1}_\pi)\tilde\pi(x^i_{\tilde\pi})}\right)$
- Can prove that **rejection rate** $\sim \varepsilon$ and **IACT** $\tau \sim 1 + \varepsilon$

Option 2: Use $\tilde\pi$ **importance weighting** **with QMC quadrature**

$$\mathbb{E}_\pi Q \approx \frac{1}{Z}\frac{1}{N}\sum_{i=1}^N Q(x^i_{\tilde\pi})\frac{\pi(x^i_{\tilde\pi})}{\tilde\pi(x^i_{\tilde\pi})} \quad \text{with} \quad Z = \frac{1}{N}\sum_{i=1}^N \frac{\pi(x^i_{\tilde\pi})}{\tilde\pi(x^i_{\tilde\pi})}$$

- We can use an unbiased (randomised) QMC rule for both integrals.

Sampling from exact π: Unbiased estimates of $\mathbb{E}_\pi Q$

using TT approximation as **preconditioner**, **importance weight** or **control variate**

Option 1: Use $\{x^i_{\tilde\pi}\}$ as (i.i.d.) **proposals** in Metropolis-Hastings

- Accept proposal $x^i_{\tilde\pi}$ with probability $\alpha = \min\left(1, \frac{\pi(x^i_{\tilde\pi})\tilde\pi(x^{i-1}_{\pi})}{\pi(x^{i-1}_{\pi})\tilde\pi(x^i_{\tilde\pi})}\right)$
- Can prove that **rejection rate** $\sim \varepsilon$ and **IACT** $\tau \sim 1 + \varepsilon$

Option 2: Use $\tilde\pi$ **importance weighting** **with QMC quadrature**

$$\mathbb{E}_\pi Q \approx \frac{1}{Z}\frac{1}{N}\sum_{i=1}^{N} Q(x^i_{\tilde\pi})\frac{\pi(x^i_{\tilde\pi})}{\tilde\pi(x^i_{\tilde\pi})} \quad \text{with} \quad Z = \frac{1}{N}\sum_{i=1}^{N}\frac{\pi(x^i_{\tilde\pi})}{\tilde\pi(x^i_{\tilde\pi})}$$

- We can use an unbiased (randomised) QMC rule for both integrals.

Option 3: Use estimate w.r.t. $\tilde\pi$ as **control variate** (**multilevel MCMC**)

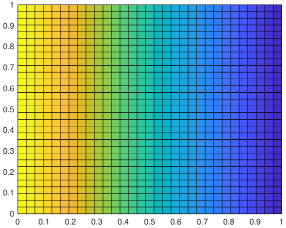
Numerical Example (Inverse Stationary Diffusion Problem)

Model Problem (representative for subsurface flow or structural mechanics)

$$-\nabla \kappa(\boldsymbol{\xi}, x) \nabla u(\boldsymbol{\xi}, x) = 0 \qquad \boldsymbol{\xi} \in (0,1)^2$$

$$u|_{\xi_1=0} = 1, \qquad u|_{\xi_1=1} = 0,$$

$$\left.\frac{\partial u}{\partial n}\right|_{\xi_2=0} = 0, \quad \left.\frac{\partial u}{\partial n}\right|_{\xi_2=1} = 0.$$



- **Karhunen-Loève expansion** of $\log \kappa(\boldsymbol{\xi}, x) = \sum_{k=1}^{d} \phi_k(\boldsymbol{\xi}) x_k$ with prior $d = 11$, $x_k \sim U[-1,1]$, $\|\phi_k\|_\infty = \mathcal{O}(k^{-\frac{3}{2}})$ [Eigel, Pfeffer, Schneider '16]

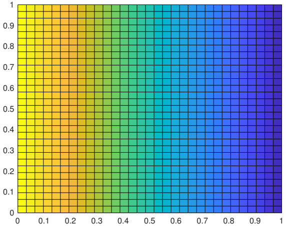
Numerical Example (Inverse Stationary Diffusion Problem)

Model Problem (representative for subsurface flow or structural mechanics)

$$-\nabla \kappa(\boldsymbol{\xi}, x) \nabla u(\boldsymbol{\xi}, x) = 0 \qquad \boldsymbol{\xi} \in (0,1)^2$$

$$u\big|_{\xi_1=0} = 1, \qquad u\big|_{\xi_1=1} = 0,$$

$$\frac{\partial u}{\partial n}\bigg|_{\xi_2=0} = 0, \quad \frac{\partial u}{\partial n}\bigg|_{\xi_2=1} = 0.$$



- **Karhunen-Loève expansion** of $\log \kappa(\boldsymbol{\xi}, x) = \sum_{k=1}^{d} \phi_k(\boldsymbol{\xi}) x_k$ with prior $d = 11$, $x_k \sim U[-1,1]$, $\|\phi_k\|_\infty = \mathcal{O}(k^{-\frac{3}{2}})$ [Eigel, Pfeffer, Schneider '16]
- Discretisation with bilinear FEs on uniform mesh with $h = 1/64$.

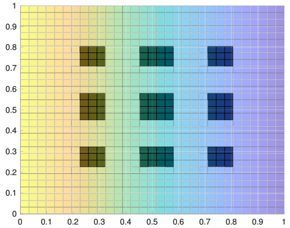
Numerical Example (Inverse Stationary Diffusion Problem)

Model Problem (representative for subsurface flow or structural mechanics)

$$-\nabla \kappa(\boldsymbol{\xi}, x)\nabla u(\boldsymbol{\xi}, x) = 0 \qquad \boldsymbol{\xi} \in (0,1)^2$$

$$u|_{\xi_1=0} = 1, \qquad u|_{\xi_1=1} = 0,$$

$$\left.\frac{\partial u}{\partial n}\right|_{\xi_2=0} = 0, \quad \left.\frac{\partial u}{\partial n}\right|_{\xi_2=1} = 0.$$



- **Karhunen-Loève expansion** of $\log \kappa(\boldsymbol{\xi}, x) = \sum_{k=1}^{d} \phi_k(\boldsymbol{\xi}) x_k$ with prior $d = 11$, $x_k \sim U[-1,1]$, $\|\phi_k\|_\infty = \mathcal{O}(k^{-\frac{3}{2}})$ [Eigel, Pfeffer, Schneider '16]
- Discretisation with bilinear FEs on uniform mesh with $h = 1/64$.
- **Data:** average pressure in 9 locations (**synthetic**, i.e. for some $\boldsymbol{\xi}^*$)

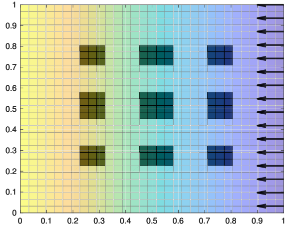
Numerical Example (Inverse Stationary Diffusion Problem)

Model Problem (representative for subsurface flow or structural mechanics)

$$-\nabla\kappa(\boldsymbol{\xi},x)\nabla u(\boldsymbol{\xi},x) = 0 \qquad \boldsymbol{\xi}\in(0,1)^2$$

$$u|_{\xi_1=0} = 1, \qquad u|_{\xi_1=1} = 0,$$

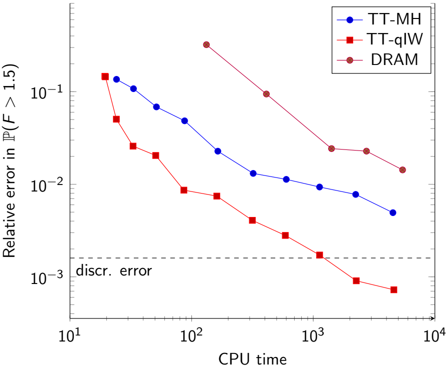
$$\left.\frac{\partial u}{\partial n}\right|_{\xi_2=0} = 0, \quad \left.\frac{\partial u}{\partial n}\right|_{\xi_2=1} = 0.$$



- **Karhunen-Loève expansion** of $\log\kappa(\boldsymbol{\xi},x) = \sum_{k=1}^{d}\phi_k(\boldsymbol{\xi})x_k$ with prior $d=11$, $x_k\sim U[-1,1]$, $\|\phi_k\|_\infty = \mathcal{O}(k^{-\frac{3}{2}})$ [Eigel, Pfeffer, Schneider '16]
- Discretisation with bilinear FEs on uniform mesh with $h=1/64$.
- **Data:** average pressure in 9 locations (**synthetic**, i.e. for some $\boldsymbol{\xi}^*$)
- **QoI** $Q=h(u(\cdot,x))$: probability that flux exceeds 1.5 (**not smooth!**)

Comparison against DRAM (for inverse diffusion problem)

noise level $\sigma_e^2 = 0.01$

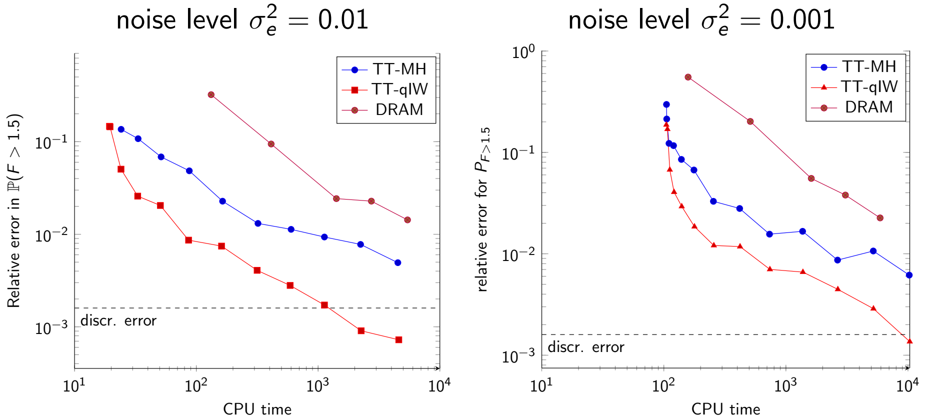


TT-MH TT conditional distribution samples (iid) as proposals for MCMC

TT-qIW TT surrogate for importance sampling with QMC

DRAM Delayed Rejection Adaptive Metropolis [Haario et al, 2006]

Comparison against DRAM (for inverse diffusion problem)

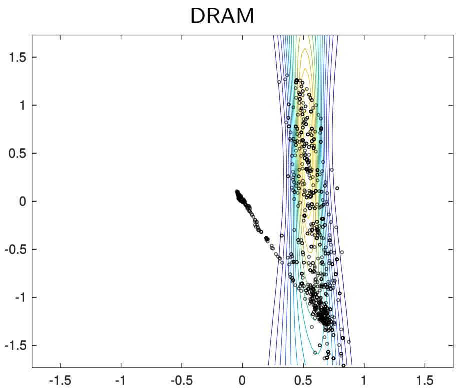


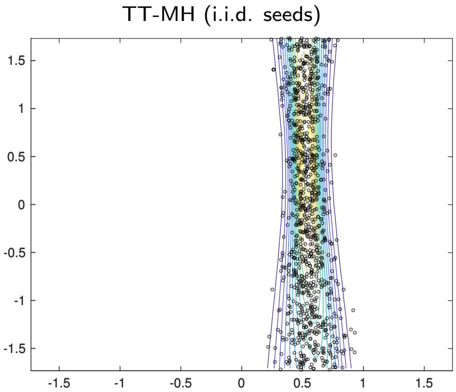
TT-MH TT conditional distribution samples (iid) as proposals for MCMC

TT-qIW TT surrogate for importance sampling with QMC

DRAM Delayed Rejection Adaptive Metropolis [Haario et al, 2006]

Samples – Comparison TT-CD vs. DRAM





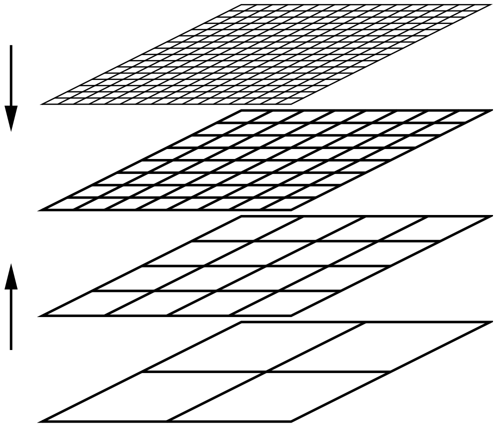
Multilevel Markov Chain Monte Carlo

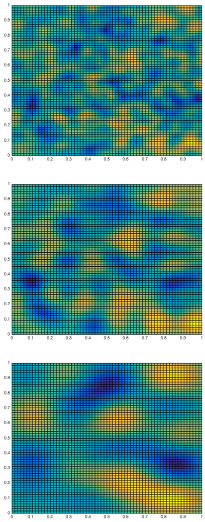
[Dodwell, Ketelsen, RS, Teckentrup, 2015 & 2019],
[Cui, Detommaso, RS, 2019]

Exploiting Model Hierarchy (same inverse diffusion problem)

V_ℓ X_ℓ

L





0

Here $h_\ell = h_0 \times 2^{-\ell}$ and $M_\ell \approx M_0 \times 2^{2\ell}$

Monte Carlo (assuming first π can be sampled – forward problem)

- **Standard Monte Carlo** estimator for $\mathbb{E}[Q]$: (where $Q = h(u(\cdot, x)) \in \mathbb{R}$)

$$\hat{Q}_L^{\mathrm{MC}} := \frac{1}{N} \sum_{i=1}^{N} Q_L^{(i)}, \quad Q_L^{(i)} \text{ i.i.d. samples with Model}(L)$$

Monte Carlo (assuming first π can be sampled – forward problem)

- **Standard Monte Carlo** estimator for $\mathbb{E}[Q]$: (where $Q = h(u(\cdot, x)) \in \mathbb{R}$)

$$\hat{Q}_L^{\mathrm{MC}} := \frac{1}{N} \sum_{i=1}^{N} Q_L^{(i)}, \quad Q_L^{(i)} \text{ i.i.d. samples with Model}(L)$$

- Convergence of plain vanilla MC **(mean square error)**:

$$\underbrace{\mathbb{E}\Big[\big(\hat{Q}_L^{\mathrm{MC}} - \mathbb{E}[Q]\big)^2\Big]}_{=:\ \mathrm{MSE}} = \underbrace{\frac{\mathbb{V}[Q_L]}{N}}_{\textbf{sampling error}} + \underbrace{\Big(\mathbb{E}[Q_L - Q]\Big)^2}_{\textbf{model error ("bias")}}$$

Monte Carlo (assuming first π can be sampled – forward problem)

- **Standard Monte Carlo** estimator for $\mathbb{E}[Q]$: (where $Q = h(u(\cdot, x)) \in \mathbb{R}$)

$$\hat{Q}_L^{\mathrm{MC}} := \frac{1}{N} \sum_{i=1}^{N} Q_L^{(i)}, \quad Q_L^{(i)} \text{ i.i.d. samples with Model}(L)$$

- Convergence of plain vanilla MC **(mean square error)**:

$$\underbrace{\mathbb{E}\left[\left(\hat{Q}_L^{\mathrm{MC}} - \mathbb{E}[Q]\right)^2\right]}_{=:\,\mathrm{MSE}} = \underbrace{\frac{\mathbb{V}[Q_L]}{N}}_{\textbf{sampling error}} + \underbrace{\left(\mathbb{E}[Q_L - Q]\right)^2}_{\textbf{model error (“bias”)}}$$

- **Assuming** $|\mathbb{E}[Q_\ell - Q]| = \mathcal{O}(2^{-\alpha\ell})$ and $\mathbb{E}[\mathrm{Cost}_\ell] = \mathcal{O}(2^{\gamma\ell})$, to get $\mathrm{MSE} = \mathcal{O}(\varepsilon^2)$, we need $L \sim \log_2(\varepsilon^{-1})\alpha^{-1}$ & $N \sim \varepsilon^{-2}$

Monte Carlo (assuming first π can be sampled – forward problem)

- **Standard Monte Carlo** estimator for $\mathbb{E}[Q]$: (where $Q = h(u(\cdot, x)) \in \mathbb{R}$)

$$\hat{Q}_L^{\mathrm{MC}} := \frac{1}{N} \sum_{i=1}^{N} Q_L^{(i)}, \quad Q_L^{(i)} \text{ i.i.d. samples with Model}(L)$$

- Convergence of plain vanilla MC **(mean square error)**:

$$\underbrace{\mathbb{E}\Big[\big(\hat{Q}_L^{\mathrm{MC}} - \mathbb{E}[Q]\big)^2\Big]}_{=:\,\mathrm{MSE}} = \underbrace{\frac{\mathbb{V}[Q_L]}{N}}_{\textbf{sampling error}} + \underbrace{\Big(\mathbb{E}[Q_L - Q]\Big)^2}_{\textbf{model error (“bias”)}}$$

- **Assuming** $|\mathbb{E}[Q_\ell - Q]| = \mathcal{O}(2^{-\alpha\ell})$ and $\mathbb{E}[\mathrm{Cost}_\ell] = \mathcal{O}(2^{\gamma\ell})$, to get $\mathrm{MSE} = \mathcal{O}(\varepsilon^2)$, we need $L \sim \log_2(\varepsilon^{-1})\alpha^{-1}$ & $N \sim \varepsilon^{-2}$

Monte Carlo Complexity Theorem

$\mathrm{Cost}(\hat{Q}_L^{\mathrm{MC}}) = \mathcal{O}(NM_L) = \mathcal{O}\big(\varepsilon^{-2-\gamma/\alpha}\big)$ to obtain $\mathrm{MSE} = \mathcal{O}(\varepsilon^2)$.

Monte Carlo (assuming first π can be sampled – forward problem)

- **Standard Monte Carlo** estimator for $\mathbb{E}[Q]$: (where $Q = h(u(\cdot, x)) \in \mathbb{R}$)

$$\hat{Q}_L^{\mathrm{MC}} := \frac{1}{N} \sum_{i=1}^{N} Q_L^{(i)}, \quad Q_L^{(i)} \text{ i.i.d. samples with Model}(L)$$

- Convergence of plain vanilla MC **(mean square error)**:

$$\underbrace{\mathbb{E}\Big[\big(\hat{Q}_L^{\mathrm{MC}} - \mathbb{E}[Q]\big)^2\Big]}_{=:\ \mathrm{MSE}} = \underbrace{\frac{\mathbb{V}[Q_L]}{N}}_{\textbf{sampling error}} + \underbrace{\Big(\mathbb{E}[Q_L - Q]\Big)^2}_{\textbf{model error (“bias”)}}$$

- **Assuming** $|\mathbb{E}[Q_\ell - Q]| = \mathscr{O}(2^{-\alpha\ell})$ and $\mathbb{E}[\mathrm{Cost}_\ell] = \mathscr{O}(2^{\gamma\ell})$, to get $\mathrm{MSE} = \mathscr{O}(\varepsilon^2)$, we need $L \sim \log_2(\varepsilon^{-1})\alpha^{-1}$ & $N \sim \varepsilon^{-2}$

Monte Carlo Complexity Thm. (2D model problem w. AMG: $\alpha = 1$, $\gamma = 2$)

$\mathrm{Cost}(\hat{Q}_L^{\mathrm{MC}}) = \mathscr{O}(NM_L) = \mathscr{O}\big(\varepsilon^{-2-\gamma/\alpha}\big)$ to obtain $\mathrm{MSE} = \mathscr{O}(\varepsilon^2)$.

Multilevel Monte Carlo [Heinrich, '98], [Giles, '07]

Basic Idea: Note that trivially

$$Q_L = Q_0 + \sum_{\ell=1}^{L} Q_\ell - Q_{\ell-1}$$

Multilevel Monte Carlo [Heinrich, '98], [Giles, '07]

Basic Idea: Note that trivially (due to linearity of $\mathbb{E}$)

$$\mathbb{E}[Q_L] = \mathbb{E}[Q_0] \; + \; \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}] \quad \boxed{\text{Control Variates!!}}$$

Multilevel Monte Carlo [Heinrich, '98], [Giles, '07]

Basic Idea: Note that trivially (due to linearity of $\mathbb{E}$)

$$\mathbb{E}[Q_L] = \mathbb{E}[Q_0] \; + \; \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}] \quad \boxed{\text{Control Variates!!}}$$

Define the following **multilevel MC** estimator for $\mathbb{E}[Q]$:

$$\widehat{Q}_L^{MLMC} := \hat{Q}_0^{\mathrm{MC}} \; + \; \sum_{\ell=1}^{L} \hat{Y}_\ell^{\mathrm{MC}} \quad \text{where} \quad Y_\ell := Q_\ell - Q_{\ell-1}$$

Multilevel Monte Carlo [Heinrich, '98], [Giles, '07]

Basic Idea: Note that trivially (due to linearity of $\mathbb{E}$)

$$\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}] \quad \boxed{\text{Control Variates!!}}$$

Define the following **multilevel MC** estimator for $\mathbb{E}[Q]$:

$$\widehat{Q}_L^{MLMC} := \hat{Q}_0^{\mathrm{MC}} + \sum_{\ell=1}^{L} \hat{Y}_\ell^{\mathrm{MC}} \quad \text{where} \quad Y_\ell := Q_\ell - Q_{\ell-1}$$

Key Observation: (Variance Reduction! Corrections cheaper!)

Level L: $\mathbb{V}[Q_L - Q_{L-1}] \to 0$ as $L \to \infty \Rightarrow N_L = \mathcal{O}(1)$ (best case)

Multilevel Monte Carlo [Heinrich, '98], [Giles, '07]

Basic Idea: Note that trivially (due to linearity of $\mathbb{E}$)

$$\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}] \quad \boxed{\text{Control Variates!!}}$$

Define the following **multilevel MC** estimator for $\mathbb{E}[Q]$:

$$\widehat{Q}_L^{MLMC} := \hat{Q}_0^{\mathrm{MC}} + \sum_{\ell=1}^{L} \hat{Y}_\ell^{\mathrm{MC}} \quad \text{where} \quad Y_\ell := Q_\ell - Q_{\ell-1}$$

Key Observation: (Variance Reduction! Corrections cheaper!)

Level L: $\mathbb{V}[Q_L - Q_{L-1}] \to 0$ as $L \to \infty \;\Rightarrow\; N_L = \mathscr{O}(1)$ (best case)

Level 0: $N_0 \sim N$ **but** $\mathrm{Cost}_0 = \mathscr{O}(M_0) = \mathscr{O}(1)$

Multilevel Monte Carlo [Heinrich, '98], [Giles, '07]

Basic Idea: Note that trivially (due to linearity of $\mathbb{E}$)

$$\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}] \quad \boxed{\text{Control Variates!!}}$$

Define the following **multilevel MC** estimator for $\mathbb{E}[Q]$:

$$\widehat{Q}_L^{MLMC} := \hat{Q}_0^{\mathrm{MC}} + \sum_{\ell=1}^{L} \hat{Y}_\ell^{\mathrm{MC}} \quad \text{where} \quad Y_\ell := Q_\ell - Q_{\ell-1}$$

Key Observation: (Variance Reduction! Corrections cheaper!)

Level L: $\mathbb{V}[Q_L - Q_{L-1}] \to 0$ as $L \to \infty$ $\Rightarrow$ $N_L = \mathcal{O}(1)$ (best case)

$\vdots$

Level ℓ: N_ℓ optimised to "balance" with cost on levels 0 and L

$\vdots$

Level 0: $N_0 \sim N$ **but** $\mathrm{Cost}_0 = \mathcal{O}(M_0) = \mathcal{O}(1)$

Complexity Theorem [Giles, '07], [Cliffe, Giles, RS, Teckentrup, '11]

Assume approximation error $\mathcal{O}(2^{-\alpha\ell})$, Cost/sample $\mathcal{O}(2^{\gamma\ell})$ **and**

$$\mathbb{V}[Q_\ell - Q_{\ell-1}] = \mathcal{O}(2^{-\beta\ell}) \qquad \text{(strong error/variance reduction)}$$

Then there exist L, $\{N_\ell\}_{\ell=0}^{L}$ to obtain $\mathrm{MSE} = \mathcal{O}(\varepsilon^2)$ with

$$\mathrm{Cost}(\widehat{Q}_L^{MLMC}) \;=\; \mathcal{O}\left(\varepsilon^{-2-\max\left(0,\frac{\gamma-\beta}{\alpha}\right)}\right) \; + \text{possible log-factor}$$

using **dependent** or **independent** estimators $\hat{Q}_0^{\mathrm{MC}}$, and $\left(\hat{Y}_\ell^{\mathrm{MC}}\right)_{\ell=1}^{L}$.

Complexity Theorem [Giles, '07], [Cliffe, Giles, RS, Teckentrup, '11]

Assume approximation error $\mathcal{O}(2^{-\alpha\ell})$, Cost/sample $\mathcal{O}(2^{\gamma\ell})$ **and**

$$\mathbb{V}[Q_\ell - Q_{\ell-1}] = \mathcal{O}(2^{-\beta\ell}) \qquad \text{(strong error/variance reduction)}$$

Then there exist L, $\{N_\ell\}_{\ell=0}^{L}$ to obtain $\mathrm{MSE} = \mathcal{O}(\varepsilon^2)$ with

$$\mathrm{Cost}(\widehat{Q}_L^{MLMC}) \; = \; \mathcal{O}\left(\varepsilon^{-2 - \max\left(0, \frac{\gamma-\beta}{\alpha}\right)}\right) \; + \text{ possible log-factor}$$

using **dependent** or **independent** estimators $\hat{Q}_0^{\mathrm{MC}}$, and $\left(\hat{Y}_\ell^{\mathrm{MC}}\right)_{\ell=1}^{L}$.

Running example (for smooth fctls. & AMG): $\alpha \approx 1$, $\beta \approx 2$, $\gamma \approx 2$

$$\mathrm{Cost}(\widehat{Q}_L^{MLMC}) \; = \; \mathcal{O}\left(\varepsilon^{-\max\left(2, \frac{\gamma}{\alpha}\right)}\right) \; = \; \mathcal{O}\left(\max(N_0, M_L)\right) \; \approx \; \mathcal{O}(\varepsilon^{-2})$$

Complexity Theorem [Giles, '07], [Cliffe, Giles, RS, Teckentrup, '11]

Assume approximation error $\mathcal{O}(2^{-\alpha\ell})$, Cost/sample $\mathcal{O}(2^{\gamma\ell})$ **and**

$$\mathbb{V}[Q_\ell - Q_{\ell-1}] = \mathcal{O}(2^{-\beta\ell}) \qquad \text{(strong error/variance reduction)}$$

Then there exist L, $\{N_\ell\}_{\ell=0}^{L}$ to obtain $\text{MSE} = \mathcal{O}(\varepsilon^2)$ with

$$\text{Cost}(\widehat{Q}_L^{MLMC}) = \mathcal{O}\left(\varepsilon^{-2-\max\left(0,\frac{\gamma-\beta}{\alpha}\right)}\right) + \text{possible log-factor}$$

using **dependent** or **independent** estimators $\hat{Q}_0^{\text{MC}}$, and $\left(\hat{Y}_\ell^{\text{MC}}\right)_{\ell=1}^{L}$.

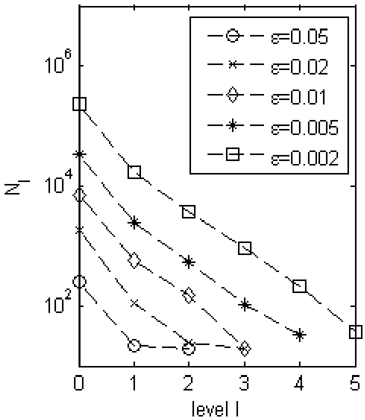
Running example (for smooth fctls. & AMG): $\alpha \approx 1$, $\beta \approx 2$, $\gamma \approx 2$

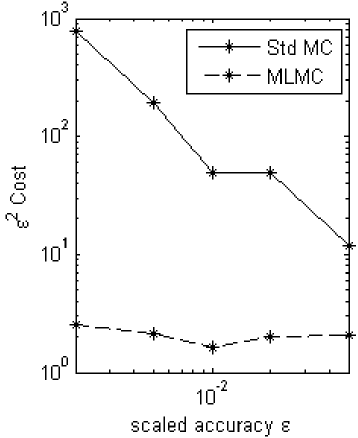
$$\text{Cost}(\widehat{Q}_L^{MLMC}) = \mathcal{O}\left(\varepsilon^{-\max\left(2,\frac{\gamma}{\alpha}\right)}\right) = \mathcal{O}\left(\max(N_0, M_L)\right) \approx \mathcal{O}(\varepsilon^{-2})$$

Optimality: Asymptotic cost of **<u>one</u>** deterministic solve (to tol$=\varepsilon$) **!**

Numerical Example (Multilevel MC)

Running example with $Q = \|u\|_{L_2(D)}$





$h_0 = \frac{1}{8}$; lognormal diffusion coeff. w. exponential covariance ($\sigma^2 = 1$, $\lambda = 0.3$)

Inverse Problem: Multilevel Markov Chain Monte Carlo

Posterior distribution for PDE model problem (Bayes):

$$\pi^\ell(x_\ell | y^{\mathrm{obs}}) \; \eqsim \; \exp(-\|y^{\mathrm{obs}} - F_\ell(x_\ell)\|^2_{\Sigma^{\mathrm{obs}}}) \, \pi_{\mathrm{prior}}(x_\ell)$$

Inverse Problem: Multilevel Markov Chain Monte Carlo

Posterior distribution for PDE model problem (Bayes):

$$\pi^\ell(x_\ell|y^{\mathrm{obs}}) \;\eqsim\; \exp(-\|y^{\mathrm{obs}} - F_\ell(x_\ell)\|^2_{\Sigma^{\mathrm{obs}}})\,\pi_{\mathrm{prior}}(x_\ell)$$

What were the **key ingredients** of “standard” multilevel Monte Carlo?

Inverse Problem: Multilevel Markov Chain Monte Carlo

Posterior distribution for PDE model problem (Bayes):

$$\pi^\ell(x_\ell|y^{\mathrm{obs}}) \;\eqsim\; \exp(-\|y^{\mathrm{obs}} - F_\ell(x_\ell)\|^2_{\Sigma^{\mathrm{obs}}})\,\pi_{\mathrm{prior}}(x_\ell)$$

What were the **key ingredients** of "standard" multilevel Monte Carlo?

- **Telescoping sum:** $\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}]$
- Models on coarser levels **much cheaper** to solve ($M_0 \ll M_L$).
- $\mathbb{V}[Q_\ell - Q_{\ell-1}] \stackrel{\ell\to\infty}{\longrightarrow} 0$ as $\implies$ much **fewer samples** on finer levels.

Inverse Problem: Multilevel Markov Chain Monte Carlo

Posterior distribution for PDE model problem (Bayes):

$$\pi^\ell(x_\ell|y^{\mathrm{obs}}) \;\eqsim\; \exp(-\|y^{\mathrm{obs}} - F_\ell(x_\ell)\|^2_{\Sigma^{\mathrm{obs}}})\, \pi_{\mathrm{prior}}(x_\ell)$$

What were the **key ingredients** of "standard" multilevel Monte Carlo?

- **Telescoping sum:** $\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}]$
- Models on coarser levels **much cheaper** to solve ($M_0 \ll M_L$).
- $\mathbb{V}[Q_\ell - Q_{\ell-1}] \xrightarrow{\ell\to\infty} 0$ as $\implies$ much **fewer samples** on finer levels.

But Important! In MCMC the target distribution π^ℓ **depends on** ℓ:

$$\mathbb{E}_{\pi^L}[Q_L] = \mathbb{E}_{\pi^0}[Q_0] + \sum_\ell \mathbb{E}_{\pi^\ell}[Q_\ell] - \mathbb{E}_{\pi^{\ell-1}}[Q_{\ell-1}]$$

Inverse Problem: Multilevel Markov Chain Monte Carlo

Posterior distribution for PDE model problem (Bayes):

$$\pi^\ell(x_\ell | y^{\mathrm{obs}}) \eqsim \exp(-\|y^{\mathrm{obs}} - F_\ell(x_\ell)\|^2_{\Sigma^{\mathrm{obs}}})\, \pi_{\mathrm{prior}}(x_\ell)$$

What were the **key ingredients** of "standard" multilevel Monte Carlo?

- **Telescoping sum:** $\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}]$
- Models on coarser levels **much cheaper** to solve ($M_0 \ll M_L$).
- $\mathbb{V}[Q_\ell - Q_{\ell-1}] \xrightarrow{\ell\to\infty} 0$ as $\Longrightarrow$ much **fewer samples** on finer levels.

But Important! In MCMC the target distribution π^ℓ **depends on** ℓ:

$$\mathbb{E}_{\pi^L}[Q_L] = \underbrace{\mathbb{E}_{\pi^0}[Q_0]}_{\text{standard MCMC}} + \sum_\ell \underbrace{\mathbb{E}_{\pi^\ell}[Q_\ell] - \mathbb{E}_{\pi^{\ell-1}}[Q_{\ell-1}]}_{\text{multilevel MCMC (NEW)}}$$

$$\widehat{Q}^{\mathrm{MLMetH}}_{h,s} := \frac{1}{N_0}\sum_{n=1}^{N_0} Q_0(z^n_{0,0}) + \sum_{\ell=1}^{L} \frac{1}{N_\ell}\sum_{n=1}^{N_\ell}\left(Q_\ell(z^n_{\ell,\ell}) - Q_{\ell-1}(z^n_{\ell,\ell-1})\right)$$

Multilevel Markov Chain Monte Carlo – Algorithm

[Dodwell, Ketelsen, RS, Teckentrup, JUQ 2015 or SIREV 2019]

ALGORITHM 2 (Multilevel Metropolis Hastings MCMC for $Q_\ell - Q_{\ell-1}$)

At states $z^n_{\ell,0}, \ldots, z^n_{\ell,\ell}$ of $\ell + 1$ Markov chains on levels $0, \ldots, \ell$:

1. $k = 0$: Set $x^0_0 := z^n_{\ell,0}$. Generate samples $x^i_0 \sim \pi^0$ (coarse posterior) via basic **Metropolis-Hastings**.
2. $k > 0$: Set $x^0_k := z^n_{\ell,k}$. Generate samples $x^i_k \sim \pi^k$ as follows:

   (a) Propose $x'_k = x^{(i+1)t_{k-1}}_{k-1}$

   Subsample to reduce correlation!

   (b) Accept x'_k with probability

   $$\alpha^{\mathrm{ML}}_\ell(x'_k|x^i_k) = \min\left(1, \frac{\pi^k(x'_k)\,\mathrm{q}^{\mathrm{ML}}_k(x^n_k|x'_k)}{\pi^k(x^n_k)\,\mathrm{q}^{\mathrm{ML}}(x'_k\,|\,x^n_k)}\right)$$

   i.e. set $x^{i+1}_k = x'_k$ with prob. $\alpha^{\mathrm{ML}}_\ell(x'_k|x^i_k)$; otherwise $x^{i+1}_k = x^i_k$

Multilevel Markov Chain Monte Carlo – Algorithm

[Dodwell, Ketelsen, RS, Teckentrup, JUQ 2015 or SIREV 2019]

ALGORITHM 2 (Multilevel Metropolis Hastings MCMC for $Q_\ell - Q_{\ell-1}$)

At states $z^n_{\ell,0}, \ldots, z^n_{\ell,\ell}$ of $\ell + 1$ Markov chains on levels $0, \ldots, \ell$:

1. $k = 0$: Set $x^0_0 := z^n_{\ell,0}$. Generate samples $x^i_0 \sim \pi^0$ (coarse posterior) via basic **Metropolis-Hastings**.
2. $k > 0$: Set $x^0_k := z^n_{\ell,k}$. Generate samples $x^i_k \sim \pi^k$ as follows:

   (a) Propose $x'_k = x^{(i+1)t_{k-1}}_{k-1}$

   Subsample to reduce correlation!

   (b) Accept x'_k with probability

   $$\alpha^{\mathrm{ML}}_\ell(x'_k | x^i_k) = \min\left(1, \frac{\pi^k(x'_k)\pi^{k-1}(x^n_k)}{\pi^k(x^n_k)\pi^{k-1}(x'_k)}\right)$$

   JS Liu, 2001

   i.e. set $x^{i+1}_k = x'_k$ with prob. $\alpha^{\mathrm{ML}}_\ell(x'_k | x^i_k)$; otherwise $x^{i+1}_k = x^i_k$

   (c) Set $z^{n+1}_{\ell,k} := x^{T_k}_k$ with $T_k := \prod_{j=k}^{\ell-1} t_j$.

Comments

- Each $\{z_{\ell,k}^n\}_{n\geq 1}$ is a **Markov chain** targeting π^k, $k = 0, \ldots, \ell$.
- In the limit of infinite subsampling rate, the chains are unbiased and the multilevel algorithm is **consistent** (no bias between levels).

  (In practice, with subsampling rate $\approx$ IACT the bias is negligible.)

Main Theoretical Results from [Dodwell, Ketelsen, RS, Teckentrup, '15]

$$\mathbb{E}_{\pi^\ell,\pi^\ell}\left[1 - \alpha_\ell^{\mathrm{ML}}(\cdot|\cdot)\right] = \mathcal{O}(h_\ell^{1-\delta}) \quad \forall\delta > 0. \quad \text{(exponential covariance)}$$

$$\mathbb{V}_{\pi^\ell,\pi^{\ell-1}}\left[Q_\ell(z_{\ell,\ell}^n) - Q_{\ell-1}(z_{\ell,\ell-1}^n)\right] = \mathcal{O}(h_\ell^{1-\delta}) \quad \forall\delta > 0$$

- Algorithm is a type of **surrogate transition method** [Liu 2001] related also to **delayed acceptance** [Christen, Fox, '05]
- **But crucially**, it also exploits the variance reduction idea of **MLMC** and the paper provides **actual rates** for the diffusion problem!

More Sophisticated Proposals – Multilevel DILI
[Cui, Detommaso, RS, arXiv:1910.12431]

- **Original work: pCN random walk** proposal
  [Cotter, Dashti, Stuart '12] (no grad./Hessian info)

More Sophisticated Proposals – Multilevel DILI
[Cui, Detommaso, RS, arXiv:1910.12431]

- **Original work: pCN random walk** proposal
  [Cotter, Dashti, Stuart '12] (no grad./Hessian info)

- **Better: DILI** [Cui, Law, Marzouk, '16]:
  (dimension-independent likelihood-informed)

  Samples from preconditioned Langevin eqn.
  using **low-rank Hessian approximation (LIS)**
  at a number of points (incl. MAP point)

More Sophisticated Proposals – Multilevel DILI

[Cui, Detommaso, RS, arXiv:1910.12431]

- **Original work: pCN random walk** proposal
  [Cotter, Dashti, Stuart '12] (no grad./Hessian info)

- **Better: DILI** [Cui, Law, Marzouk, '16]:
  (dimension-independent likelihood-informed)

  Samples from preconditioned Langevin eqn.
  using **low-rank Hessian approximation (LIS)**
  at a number of points (incl. MAP point)

More Sophisticated Proposals – Multilevel DILI

[Cui, Detommaso, RS, arXiv:1910.12431]

- **Original work: pCN random walk** proposal
  [Cotter, Dashti, Stuart '12] (no grad./Hessian info)
- **Better: DILI** [Cui, Law, Marzouk, '16]:
  (dimension-independent likelihood-informed)

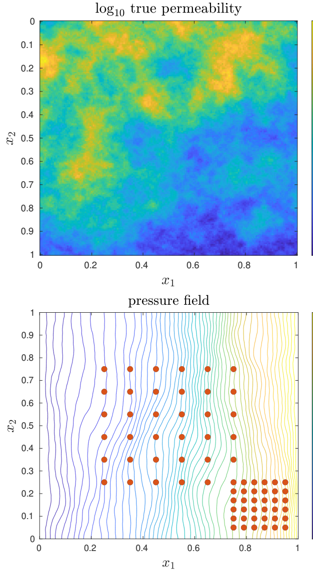
  Samples from preconditioned Langevin eqn.
  using **low-rank Hessian approximation (LIS)**
  at a number of points (incl. MAP point)
- [Cui et al, '19]: **Hierarchical** construction
  of **LIS** (which is significantly **cheaper!**) and
  combination of DILI with MLMCMC.

More Sophisticated Proposals – Multilevel DILI

[Cui, Detommaso, RS, arXiv:1910.12431]

- **Original work: pCN random walk** proposal [Cotter, Dashti, Stuart '12] (no grad./Hessian info)
- **Better: DILI** [Cui, Law, Marzouk, '16]: (dimension-independent likelihood-informed)

  Samples from preconditioned Langevin eqn. using **low-rank Hessian approximation (LIS)** at a number of points (incl. MAP point)
- [Cui et al, '19]: **Hierarchical** construction of **LIS** (which is significantly **cheaper!**) and combination of DILI with MLMCMC.
- **Numerical experiment:** much higher dimensional and more complicated than above, using lognormal prior.



Numerical Comparison: IACTs & CPU Times

Refined parameters

Level ℓ	0	1	2	3
iact(pCN)	4300	45	48	24
iact(DILI)	34	11	3.6	2.0

$Q_\ell(z^n_{\ell,\ell}) - Q_{\ell-1}(z^n_{\ell,\ell-1})$

Level ℓ	0	1	2	3
iact(pCN)	4100	4.9	2.8	1.9
iact(DILI)	9.0	4.6	2.4	1.8

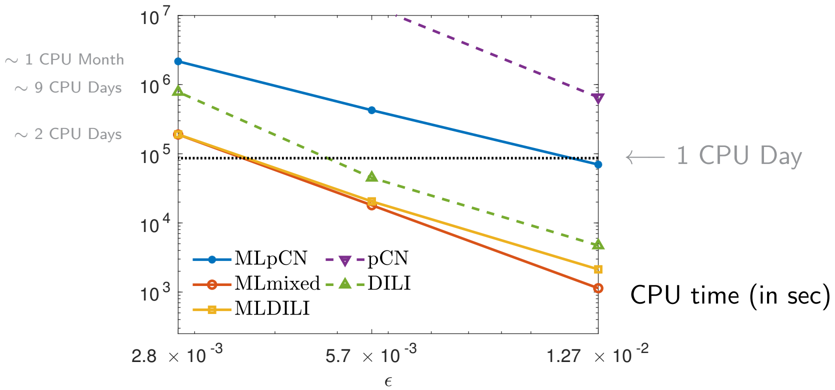
Numerical Comparison: IACTs & CPU Times

Refined parameters

Level ℓ	0	1	2	3
iact(pCN)	4300	45	48	24
iact(DILI)	34	11	3.6	2.0

$Q_\ell(z^n_{\ell,\ell}) - Q_{\ell-1}(z^n_{\ell,\ell-1})$

Level ℓ	0	1	2	3
iact(pCN)	4100	4.9	2.8	1.9
iact(DILI)	9.0	4.6	2.4	1.8



Conclusions

- Large-scale **PDE-constrained** Bayesian inference with **sparse** data
- **Idea 1:** Characterise complex/intractable distributions by constructing *deterministic couplings*
- **Variational Inference:** Optimisation of Kullback-Leibler divergence
  (Many types: sparse, decomposable, neural nets, polynomial, kernel-based)

Conclusions

- Large-scale **PDE-constrained** Bayesian inference with **sparse** data
- **Idea 1:** Characterise complex/intractable distributions by constructing *deterministic couplings*
- **Variational Inference:** Optimisation of Kullback-Leibler divergence
  (Many types: sparse, decomposable, neural nets, polynomial, kernel-based)
- **Alternative:** Low-rank tensor factorisation and conditional distribution sampling (Rosenblatt transform) [Stats & Comput, 2019]
  - Scales with dimension; comparable comput. efficiency to NNs
  - Unbiased estimates via Metropolisation or importance weighting

Conclusions

- Large-scale **PDE-constrained** Bayesian inference with **sparse** data
- **Idea 1:** Characterise complex/intractable distributions by constructing *deterministic couplings*
- **Variational Inference:** Optimisation of Kullback-Leibler divergence
  (Many types: sparse, decomposable, neural nets, polynomial, kernel-based)
- **Alternative:** Low-rank tensor factorisation and conditional distribution sampling (Rosenblatt transform) [Stats & Comput, 2019]
  - Scales with dimension; comparable comput. efficiency to NNs
  - Unbiased estimates via Metropolisation or importance weighting
- **Idea 2:** Use **model hierarchies** – Multilevel MCMC [SINUM, 2019]
  - Variance reduction and much better complexities (proven!)
  - Better IACT on fine levels through surrogate transition method
  - Further acceleration (especially on coarsest level) by using DILI

References

1. Dolgov, Anaya-Izquierdo, Fox, RS, *Approximation and sampling of multivariate probability distributions in the tensor train decomposition*, Statistics & Comput. **30**, 2020 [arXiv:1810.01212]
2. Dodwell, Ketelsen, RS, Teckentrup, *A hierarchical multilevel Markov chain MC algorithm [...]*, SIAM/ASA J Uncertain Q **3**, 2015 [arXiv:1303.7343]
3. Cui, Detommaso, RS, *Multilevel dimension-independent likelihood-informed MCMC for large-scale inverse problems*, submitted, 2019 [arXiv:1910.12431]
4. Moselhy, Marzouk, *Bayesian inference with optimal maps*, J Comput Phys 231, 2012 [arXiv:1109.1516]
5. Rezende, Mohamed, *Variational inference with normalizing flows*, ICML'15 Proc. 32nd Inter. Conf. Machine Learning, Vol. 37, 2015 [arXiv:1505.05770]
6. Marzouk, Moselhy, Parno, Spantini, *Sampling via measure transport: An introduction*, Handbook of UQ (Ghanem et al, Eds), 2016 [arXiv:1602.05023]
7. Detommaso, Cui, Spantini, Marzouk, RS, *A Stein variational Newton method*, NIPS 2018, Vol. 31, 2018 [arXiv:1806.03085]
8. Kruse, Detommaso, RS, Köthe, *HINT: Hierarchical invertible neural transport for density estimation & Bayesian inference*, 2019 [arXiv:1905.10687]