Master's thesis: Numerical comparison of MCMC methods for Quantum tomography

Danila Mokeev
Supervisors: Estelle Massart, Andrew Thompson and Tameem Adel

21st of June 2024

Ecole Polytechnique de Louvain

Scope of this thesis

Topic: Markov chain Monte Carlo (MCMC) methods in Quantum tomography

Research questions:

1. How do these methods perform in different experimental setups?
2. Why do some methods perform better than others?

Purpose:

- Enable new directions of research
- Help researchers make an informed choice for their use case

Thesis contributions

1. Numerically compare 2 MCMC algorithms, the prob-estimator and the Projected Langevin algorithm
2. Propose 2 new algorithms to understand the impact of the prior and the algorithm on the accuracy

Table of Contents

Table of Contents

Motivation behind Quantum tomography

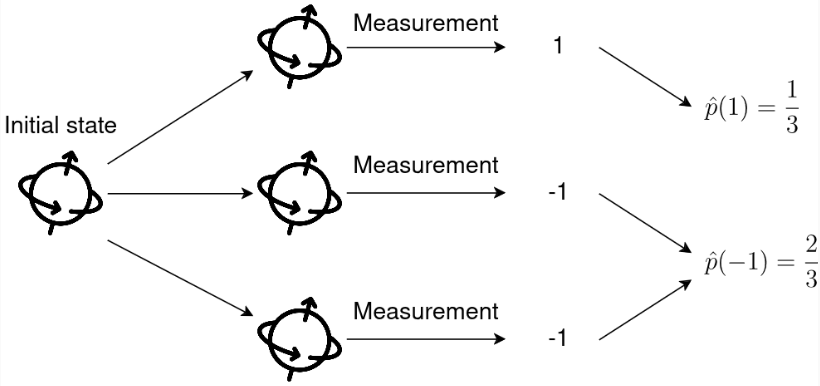
Quantum tomography is a process to reconstruct the state of a quantum system.

There are some challenges to consider:

- Quantum systems are inherently probabilistic
- A measurement can only be made once
- We can only measure the position or momentum, but not both

Quantum tomography: a diagram

Quantum tomography allows to address the existing challenges



Quantum tomography: mathematical description (1)

The Born rule states that

$$p(m) = \mathsf{tr}(\rho P_m) \tag{1}$$

with

- $p(m)$ the probability of occurence of m
- P_m the projector matrix associated to the eigenvalue m of an *observable* O
- ρ the *density matrix* representing the quantum state

The size of ρ is $2^n \times 2^n$ with n the number of qubits.

Quantum tomography: mathematical description (2)

If we flatten the matrices

$$\vec{P} = \begin{bmatrix} P_{11} & P_{12} & \cdots \end{bmatrix} \tag{2}$$

$$A = \begin{bmatrix} \vec{P}_1 \\ \vec{P}_2 \\ \vec{P}_3 \\ \vdots \end{bmatrix} \quad \vec{\rho} = \begin{bmatrix} \rho_{11} \\ \rho_{12} \\ \rho_{13} \\ \vdots \end{bmatrix} \tag{3}$$

then we can estimate ρ by solving the resulting system of equations

$$A\vec{\rho} = \hat{p} \tag{4}$$

Most common methods

- Direct methods:
$$\hat{\rho} = (A^T A)^{-1} A^T \hat{p} \tag{5}$$
- Optimization-based methods:
$$\hat{\rho} = \mathsf{argmin}_{\vec{\rho}} ||A\vec{\rho} - \hat{p}||_2 \tag{6}$$
- Pauli basis expansion:
$$\hat{\rho} = \sum_{b \in \{I,x,y,z\}^n} \rho_b \sigma_b \tag{7}$$
- Bayesian methods, and in particular MCMC methods
$$\hat{\rho} = \frac{1}{N} \sum_{i=1}^{N} \rho_i \quad \text{with } \rho_i \sim \pi(\rho|\mathbf{D}) \tag{8}$$

Existing methods: our focus in this thesis

- Direct methods:

$$\hat{\rho} = (A^T A)^{-1} A^T \hat{p} \tag{9}$$

- Optimization-based methods:

$$\hat{\rho} = \mathrm{argmin}_{\vec{\rho}} ||A\vec{\rho} - \hat{p}|| \tag{10}$$

- Pauli basis expansion:

$$\hat{\rho} = \sum_{b \in \{I,x,y,z\}^n} \rho_b \sigma_b \tag{11}$$

- Bayesian methods, and in particular MCMC methods

$$\hat{\rho} = \frac{1}{N} \sum_{i=1}^{N} \rho_i \quad \text{with } \rho_i \sim \pi(\rho|\mathbf{D}) \tag{12}$$

Table of Contents

Bayesian framework

In the Bayesian framework:

$$\underbrace{\pi(\rho|\mathbf{D})}_{\text{Posterior}} \propto \underbrace{\pi(\mathbf{D}|\rho)}_{\text{Likelihood}} \underbrace{\pi(\rho)}_{\text{Prior}} \tag{13}$$

Recall that each term is a distribution!

In the context of Quantum tomography:

- Likelihood $\pi(\mathbf{D}|\rho) = \exp(-||A\vec{\rho} - \hat{p}||_2^2)$
- Prior $\pi(\rho)$ is method specific

Markov chain Monte Carlo methods

- Markov chain Monte Carlo (MCMC) methods *sample* from $\pi(\rho|\mathbf{D})$.
- They build a Markov chain of samples $\rho_1, \rho_2, \ldots$ such that

$$f(x) = \pi(\rho|\mathbf{D}) \tag{14}$$

  with the equilibrium distribution $f(x)$ of the chain
- The density matrix is then calculated as

$$\tilde{\rho} = \mathbb{E}[\rho] = \int \rho\pi(\rho|\mathbf{D})d\rho \tag{15}$$

$$\Leftrightarrow \hat{\rho} = \frac{1}{N}\sum_{i=1}^{N}\rho_i \quad \text{with } \rho_i \sim \pi(\rho|\mathbf{D}) \tag{16}$$

An example: Metropolis-Hastings algorithm

mcmc.gif

Advantages of MCMC algorithms

Why are we interested in MCMC methods?

- Prior $\pi(\rho)$: additional information about the density matrix - low-rank for example
- Uncertainty quantification: working with distributions instead of point estimates

Table of Contents

Prob-estimator: prior

- Introduced by Mai and Alquier in 2017 [MA17]
- Sum of rank-1 matrices:

$$\rho = \sum_{i=1}^{d} \gamma_i V_i {V_i}^{\dagger}$$

- The prior $\pi_1(\gamma_1 \dots \gamma_d)$ is a Dirichlet distribution. A typical draw leads to a sparse vector.

$$\gamma = \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix}$$

- The prior $\pi_2(V_i)$ is a unit sphere distribution

$$\|V_i\|_2 = 1$$

Prob-estimator: algorithm

Algorithm: Prob-estimator algorithm

for $t \leftarrow 1 : T$ **do**

```
// Iterate over each dimension i
```

for $i \leftarrow 1 : d$ **do**

1. Sample γ_i^* from $\pi_1(\gamma_i)$
2. Update $\gamma^{(t)}$ with accept/reject step

end

for $i \leftarrow 1 : d$ **do**

1. Sample V_i^* from $\pi_2(V_i)$
2. Update $V^{(t)}$ with an accept/reject step

end

end

Prob-estimator: algorithm

Algorithm: Prob-estimator algorithm

for $t \leftarrow 1 : T$ **do**

    // Iterate over each dimension i

    for $i \leftarrow 1 : d$ **do**

        1. Sample γ_i^* from $\pi_1(\gamma_i)$
        2. Update $\gamma^{(t)}$ with accept/reject step

    end

    for $i \leftarrow 1 : d$ **do**

        1. Sample V_i^* from $\pi_2(V_i)$
        2. Update $V^{(t)}$ with an accept/reject step

    end

end

Prob-estimator: algorithm

Algorithm: Prob-estimator algorithm

for $t \leftarrow 1 : T$ **do**

```
// Iterate over each dimension i
```

for $i \leftarrow 1 : d$ **do**

1. Sample γ_i^* from $\pi_1(\gamma_i)$
2. Update $\gamma^{(t)}$ with accept/reject step

end

for $i \leftarrow 1 : d$ **do**

1. Sample V_i^* from $\pi_2(V_i)$
2. Update $V^{(t)}$ with an accept/reject step

end

end

Prob-estimator: algorithm

Algorithm: Prob-estimator algorithm

for $t \leftarrow 1 : T$ **do**

    `// Iterate over each dimension` i

    for $i \leftarrow 1 : d$ **do**

        1. Sample γ_i^* from $\pi_1(\gamma_i)$
        2. Update $\gamma^{(t)}$ with accept/reject step

    end

    for $i \leftarrow 1 : d$ **do**

        1. Sample V_i^* from $\pi_2(V_i)$
        2. Update $V^{(t)}$ with an accept/reject step

    end

end

Prob-estimator: algorithm

Algorithm: Prob-estimator algorithm

```
for t ← 1 : T do
    // Iterate over each dimension i
    for i ← 1 : d do
        1. Sample γ_i^* from π_1(γ_i)
        2. Update γ^(t) with accept/reject step
    end
    for i ← 1 : d do
        1. Sample V_i^* from π_2(V_i)
        2. Update V^(t) with an accept/reject step
    end
end
```

Projected Langevin: prior

- Introduced by Adel, Chrétien, Massart, Thompson in 2024 [Ade+24]
- Burer-Monteiro factorization: $\rho = YY^\dagger$, with $\text{rank}(Y) = r$
- Low-rank prior: spectral scaled Student-t distribution

$$\pi(Y) = \prod_{j=1}^{r} (\theta^2 + \underbrace{s_j(Y)^2}_{j\text{th singular value of } Y})^{-(2d+r+2)/2} \tag{17}$$

- Promotes sparsity among the eigenvalues leading to a low rank
- Very similar to the Student-t distribution

Projected Langevin: algorithm

Algorithm: Projected Langevin algorithm

for $t \leftarrow 1 : T$ **do**

1. Sample $\tilde{w}^{(t)} \sim N(\mathbf{0}, \mathbf{I})$
2. $\tilde{Y}^{(t)} \leftarrow \tilde{Y}^{(t-1)} - \eta^{(t)} \underbrace{\nabla \pi(\tilde{Y}^{(t-1)} | \mathbf{D})}_{\textbf{gradient}} + \frac{\sqrt{2\eta^{(t)}}}{\beta} \tilde{w}^{(t)}$

end

The gradient allows us to explore the regions of high density faster.

Projected Langevin: algorithm

Algorithm: Projected Langevin algorithm

for $t \leftarrow 1 : T$ **do**

1. Sample $\tilde{w}^{(t)} \sim N(\mathbf{0}, \mathbf{I})$

2. $\tilde{Y}^{(t)} \leftarrow \tilde{Y}^{(t-1)} - \eta^{(t)} \underbrace{\nabla \pi(\tilde{Y}^{(t-1)} | \mathbf{D})}_{\textbf{gradient}} + \frac{\sqrt{2\eta^{(t)}}}{\beta} \tilde{w}^{(t)}$

end

The gradient allows us to explore the regions of high density faster.

Projected Langevin: algorithm

Algorithm: Projected Langevin algorithm

for $t \leftarrow 1 : T$ **do**

1. Sample $\tilde{w}^{(t)} \sim N(\mathbf{0}, \mathbf{I})$
2. $\tilde{Y}^{(t)} \leftarrow \tilde{Y}^{(t-1)} - \eta^{(t)} \underbrace{\nabla \pi(\tilde{Y}^{(t-1)} | \mathbf{D})}_{\textbf{gradient}} + \frac{\sqrt{2\eta^{(t)}}}{\beta} \tilde{w}^{(t)}$

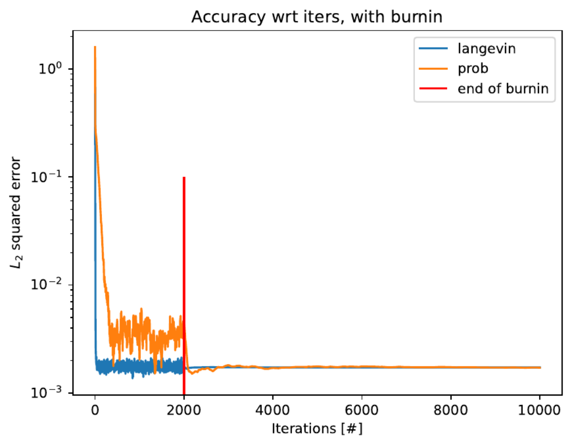
end

The gradient allows us to explore the regions of high density faster.

Table of Contents

Convergence plot

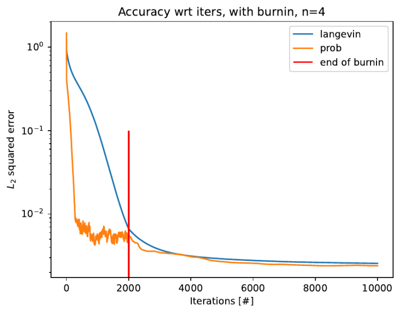
Note: the error corresponds to $||\rho - \hat{\rho}||_2^2$



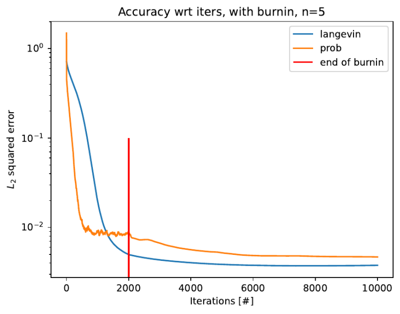
$\Longrightarrow$ Projected Langevin converges faster for $n = 3$ qubits

Convergence speed for $n = 4, 5$

Reminder: a larger n means a larger density matrix



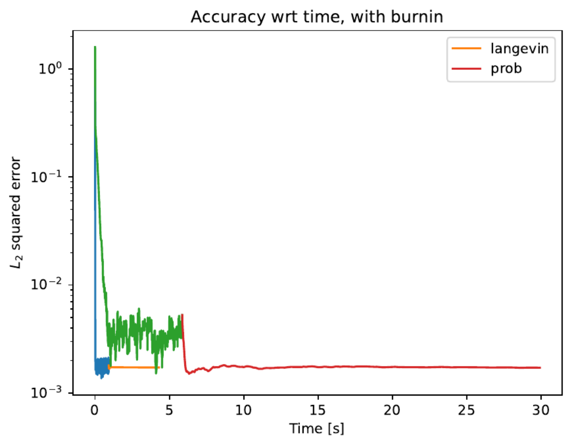
(a) $n = 4$



(b) $n = 5$

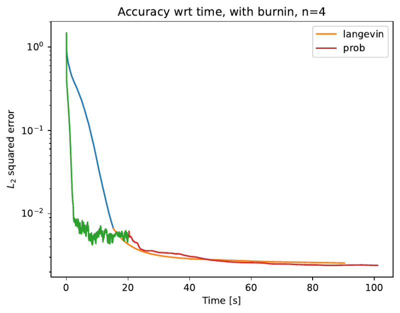
$\Longrightarrow$ Projected Langevin converges slower than previously

Computation time for $n = 3$

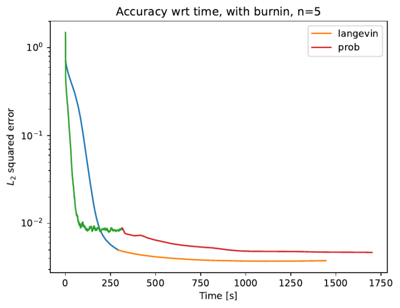


$\Longrightarrow$ For $n = 3$, Projected Langevin takes much less time

Computation time for $n = 4, 5$



(a) $n = 4$



(b) $n = 5$

$\Longrightarrow$ When n increases, Projected Langevin becomes as slow as the prob-estimator due to the gradient cost.

Introducing 2 new methods

What makes Projected Langevin perform better ?

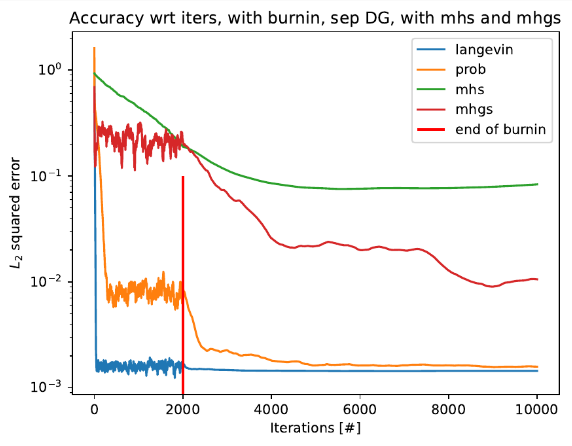
To answer this question, we introduce 2 new algorithms:

1. Metropolis-Hastings with Student-t prior (MHS)
2. Metropolis-Hastings with Gibbs with Student-t prior (MHGS)

They combine:

- The algorithm from the prob-estimator
- The prior from the Projected Langevin algorithm

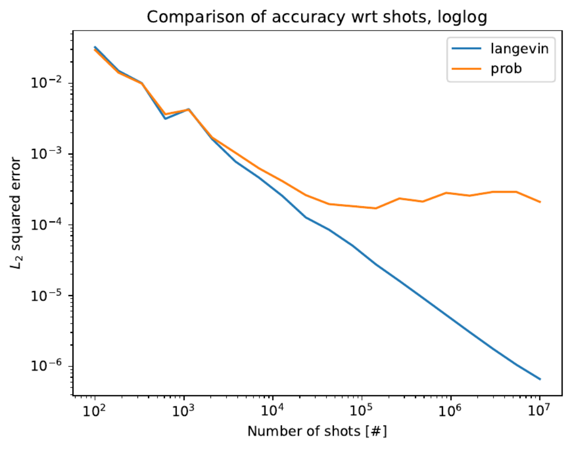
Convergence comparison



$\Longrightarrow$ The prior itself is not a solution, and must be paired with the right algorithm to be fast and accurate

Impact of the number of shots

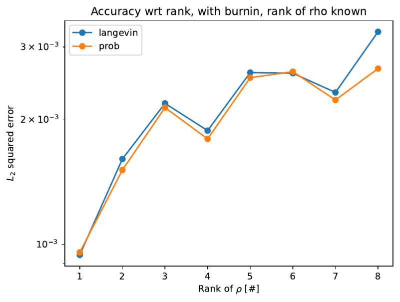
Shot: measurement we perform on a clone of the state



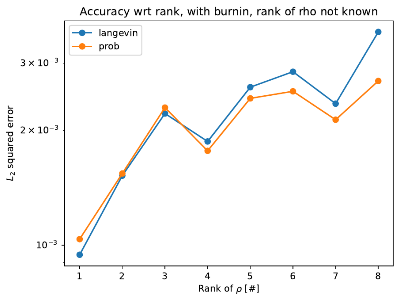
$\Longrightarrow$ The prob-estimator does not scale!

Impact of knowing the rank of ρ

Reminder: for Projected Langevin, $\rho = YY^\dagger$, with $\text{rank}(Y) = r$



(a) Rank of ρ known



(b) Rank of ρ not known

$\Longrightarrow$ The information about the rank only marginally affects the accuracy

Summary

- Quantum tomography is not yet a solved problem, especially for large systems
- MCMC methods are a promising direction of research, thanks to uncertainty quantification and prior information
- The choice of the algorithm might have more impact on the convergence speed and accuracy than the prior

Future work

- Extend tests to larger n to check for robustness of conclusions
- Try other gradient-based algorithms to see the impact on convergence speed (for example HMC)
- Try other priors to see the impact on computation time (the gradient may be faster to compute) and accuracy

References

[MA17] The Tien Mai and Pierre Alquier. **Pseudo-Bayesian quantum tomography with rank-adaptation.** 2017.

[Ade+24] Tameem Adel, Stéphane Chrétien, Estelle Massart, and Andrew Thompson. **“A projected Langevin sampling algorithm for quantum tomography”.** 2024.

Prob-estimator: full

It can be seen as eigendecomposition, without the orthogonality property:

$$\rho = U\Lambda U^\dagger \tag{18}$$

Prior:

$$\pi(\rho) = \pi_1(\gamma_1, \ldots, \gamma_d)\prod_{i=1}^{d}\pi_{2,i}(V_i) \tag{19}$$

Likelihood:

$$\pi(\mathbf{D}|\rho) = \pi(\rho, \mathbf{D}) = \exp(-\lambda\ell(\rho, \mathbf{D})) \tag{20}$$

with:

$$\ell(\rho, \mathbf{D}) = \sum_{\mathbf{a}\in\mathcal{E}^n}\sum_{\mathbf{s}\in\mathcal{R}^n}\left[\mathrm{tr}(\rho P_{\mathbf{s}}^{\mathbf{a}}) - \hat{p}_{\mathbf{a},\mathbf{s}}\right]^2 \tag{21}$$

Posterior:

$$\pi(\nu|\mathbf{D}) \propto \exp(-\lambda\ell(\nu, \mathbf{D}))\pi(\nu) \tag{22}$$

Projected Langevin: full

Prior:

$$\nu_\theta(Y) = C_\theta \det(\theta^2 I_d + YY^\dagger)^{-(2d+r+2)/2} \tag{23}$$

Likelihood:

$$L(Y, \mathbf{D}) = \sum_{i=1}^{M} (\hat{p}_m - \mathrm{tr}(A_m Y Y^\dagger))^2 \tag{24}$$

Posterior:

$$\hat{\nu}_{\lambda,\theta}(Y, \mathbf{D}) = \exp(-f_{\lambda,\theta}(Y, \mathbf{D})) \tag{25}$$

with

$$f_{\lambda,\theta}(Y, \mathbf{D}) = \lambda \sum_{i=1}^{M} (\hat{p}_m - \mathrm{tr}(A_m Y Y^\dagger))^2 + \frac{2d+r+2}{2} \log\det(\theta^2 I_d + YY^\dagger) \tag{26}$$

Prob-estimator: full algorithm

Algorithm 0: Prob-estimator algorithm

$\gamma^{(0)} \in \mathbb{R}^{d\times 1} \leftarrow Y^{(0)}/(\sum_{i=1}^{d} Y_i^{(0)})$

for $t \leftarrow 1 : T$ **do**

$Y^{(t)} \leftarrow Y^{(t-1)}$

for $i \leftarrow 1 : d$ **do**

$\tilde{Y} \leftarrow Y^{(t)}$

Sample $y \sim U(-0.5, 0.5)$

$\tilde{Y}_i \leftarrow Y_i^{(t)} \exp(y)$

$\tilde{\gamma} \leftarrow \tilde{Y}/(\sum_{j=1}^{d} \tilde{Y}_j)$

$$Y_i^{(t)} \leftarrow \begin{cases} \tilde{Y}_i & \text{with probability} \min\{R(\tilde{Y}, Y^{(t)}, \tilde{\gamma}, \gamma^{(t)}, V^{(t-1)}, \lambda, \alpha), 1\} \\ Y_i^{(t)} & \text{otherwise} \end{cases}$$

$\gamma^{(t)} \leftarrow Y^{(t)}/(\sum_{k=1}^{d} Y_k^{(t)})$

end

Projected Langevin: full algorithm

Algorithm 1: Projected Langevin algorithm

Input : $T \in \mathbb{N}, Y^{(0)} \in \mathbb{C}^{d\times r}, \{\eta^{(k)} | k \in 1 \ldots T\}, \beta \in \mathbb{R}, \theta \in \mathbb{R}, \lambda \in \mathbb{R}$

Output: $\tilde{Y} \in \mathbb{R}^{2d\times 2r}$

$\tilde{Y}^{(0)} \leftarrow \psi(Y^{(0)})$

for $k \leftarrow 1 : T$ **do**

$w_R^{(k)}, w_I^{(k)} \sim N(0,1)^{d\times r}$ `// Sample from the standard normal of size` $d \times r$

$w^{(k)} \leftarrow w_R^{(k)} + i w_I^{(k)}$

$\tilde{w}^{(k)} \leftarrow \psi(w^{(k)})$

$$\tilde{Y}^{(k)} \leftarrow \tilde{Y}^{(k-1)} - \eta^{(k)} \nabla f(\tilde{Y}^{(k-1)}, \theta, \lambda) + \frac{\sqrt{2\eta^{(k)}}}{\beta} \tilde{w}^{(k)}$$

$\tilde{Y} \leftarrow \frac{1}{k} \tilde{Y}^{(k)} + (1 - \frac{1}{k}) \tilde{Y}$

end

MHS algorithm

Algorithm 2: Metropolis-Hastings with Student-t prior

for $k \leftarrow 1 : T$ **do**

Sample $\tilde{Y}^* \sim p_1(\tilde{Y}|\tilde{Y}^{(k-1)})$

Sample $u \sim U(0,1)$

$\alpha \leftarrow \min\left\{\log A_1(\tilde{Y}^*, \tilde{Y}^{(k-1)}, \theta, \lambda), \log(1)\right\}$

if $\log(u) \leq \alpha$ **then**

```
// Accept
```
$\tilde{Y}^*$

$\tilde{Y}^{(k)} \leftarrow \tilde{Y}^*$

else

```
// Reject
```
$\tilde{Y}^*$

$\tilde{Y}^{(k)} \leftarrow \tilde{Y}^{(k-1)}$

end

$\tilde{Y} \leftarrow \frac{1}{k}\tilde{Y}^{(k)} + (1 - \frac{1}{k})\tilde{Y}$

end

MHGS algorithm

for $k \leftarrow 1 : T$ **do**

$\tilde{Y}^{(k)} \leftarrow \tilde{Y}^{(k-1)}$

for $i \leftarrow 1 : 2d$ **do**

for $j \leftarrow 1 : 2r$ **do**

Sample $y^* \sim p_2(y|\tilde{Y}_{ij}^{(k)})$

Sample $u \sim U(0, 1)$

$\alpha \leftarrow \min\left\{\log A_2(y^*, \tilde{Y}_{ij}^{(k)}, \theta, \lambda), \log(1)\right\}$

if $\log(u) \leq \alpha$ **then**

$\tilde{Y}_{ij}^{(k)} \leftarrow y^*$

else

$\tilde{Y}_{ij}^{(k)} \leftarrow \tilde{Y}_{ij}^{(k)}$

end

end

end

Impact of shots: prob-estimator

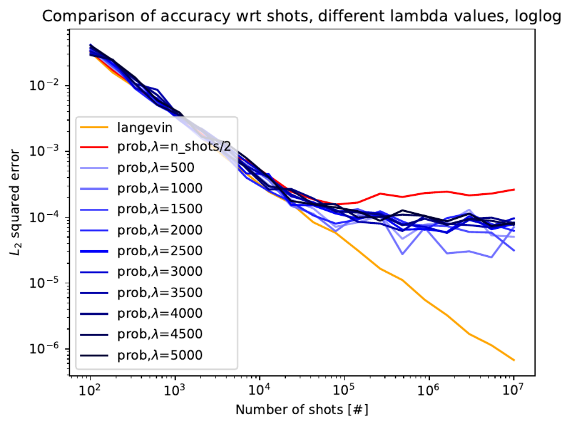
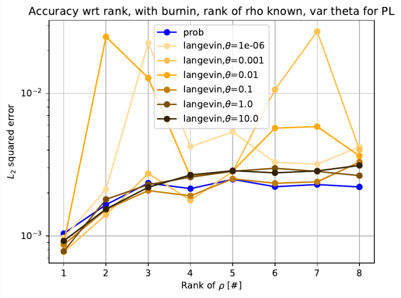
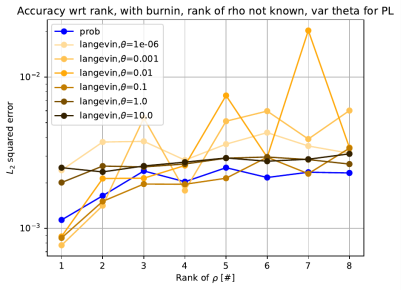


Figure 4: Impact of the number of used shots on the error with varying parameter λ for the prob-estimator with $n = 3$

Impact of knowing the rank



(a) Rank of ρ known



(b) Rank of ρ not known

Figure 5: Rank knowledge plot for $n = 3$ and different combinations of θ

Physics aspects

- Wave function collapse
- Heisenberg uncertainty principle
- No-cloning theorem

Questions and answers

- Why do you think that Projected Langevin converges faster? It is simply a property of the algorithm, langevin converges faster due to the use of the posterior
- Why do you say that the posterior is costly to calculate? Because it involves a matrix inversion, which is costly. Even though the authors use a trick, it is still expensive to calculate.
- Why do you think the prob-estimator is slow? Again, property of the method. The fact that we iterate over each dimesion reduces the speed.
- Why on the graph with the rank of ρ, we see that the error for a low rank is lower ? Probably comes from the fact that we use a low rank prior. I agree that intuitively it makes sense for higher ranks to be approximated better, as by sampling random values, chances are that the resulting approximation will be of full rank.

Questions and answers (2)

- Why do you think mhgs works better? Perhaps due to the fact that it iterates over the dimensions a la gibbs, and this results in a better accuracy, similar to the prob-estimator