Rare event simulation for a static distribution

F. Cérou, P. Del Moral, T. Furon, A. Guyader

Resim 2008, Rennes

This work was partially supported by the French Agence Nationale de la Recherche (ANR), project Nebbiano, number ANR-06-SETI-009

Introduction

$X \sim \mu$, with μ probability measure on $\mathbb{X}$ ($\mathbb{R}^d$, or a discrete space)

We know how to draw samples from μ

Given a function $S : \mathbb{X} \longmapsto \mathbb{R}$, we look at the rare event

$$\mathcal{R} = \{S(X) > \tau\}$$

We want to compute $\mu(\mathcal{R}) = \mathbb{P}(X \in \mathcal{R})$, and draw samples from $\mu_{\mathcal{R}}(dx) = \frac{1}{\mu(\mathcal{R})} \mathbb{1}_{\mathcal{R}}(x)\mu(dx)$

Motivation and examples

Watermarking of digital contents: imbedding/hiding information in a digital file (typically audio or video), such that the change is not noticed, and very hard to remove (robust to any kind of transformation, coding, compression...)

Used for: copy protection or fingerprinting

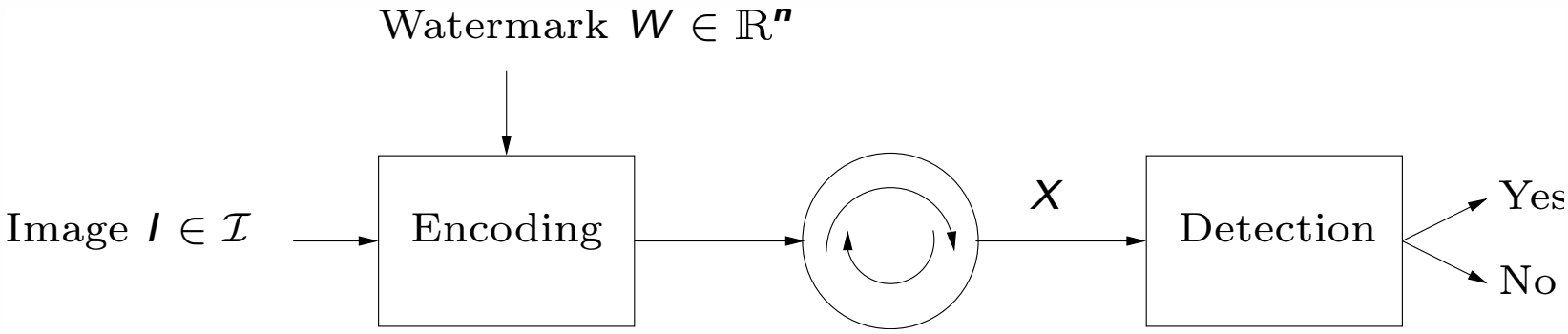


Figure 1: Watermarking

Our rare event occurs when the detection box anwers “yes” but the content is not watermarked

Zero-bit watermarking

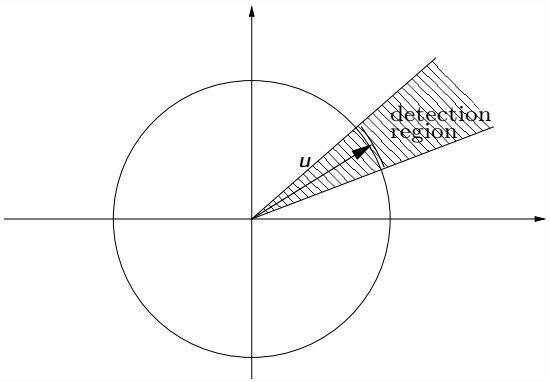


Figure 2: Zero-bit watermarking

- $u \in \mathbb{R}^d$ is a fixed and normalized secret vector.
- A content X is deemed watermarked if $S(X) = \frac{\langle X,u \rangle}{\|X\|} > \tau$.
- Classic Assumption : An unwatermarked content X has a radially symmetric pdf. As S is also radially symmetric, we choose $X \sim N(0, I)$
- False detection : $P_{fd} = \mathbb{P}(S(X) > \tau | X$ unwatermarked$)$

Toy example used to validate the algorithm

Probabilistic fingerprinting codes

Fingerprinting:

- Principle : Some personal identification sequence $F_i \in \{0,1\}^m$ is hidden in the copy of each user.
- Benefit : Find a dishonest user via his fingerprint
- False Detections : Accusing an innocent (false alarm) or accusing none of the colluders (false negative)

Tardos probabilistic codes:

- Fingerprint : $X = [X_1, \ldots, X_m]$, $X_i \sim \mathcal{B}(p_i)$ and $p_i \sim f(p)$ (same p_i's for all users)
- Pirated Copy : $y = [y_1, \ldots, y_m] \in \{0,1\}^m$
- Accusation procedure : $S(X) = \sum_{i=1}^{m} y_i g_i(X_i) \gtrless \tau$

The choice of f and the g_i's is crucial (but not discussed here)

Collusions

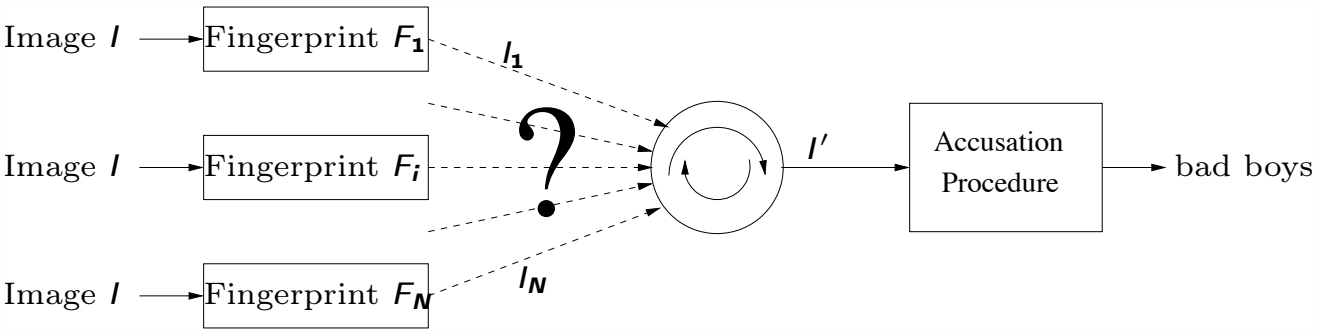


Figure 3: Collusion

Several users compare their digital content: they are not exactly the same...

Stategies to build up a new file, different from all the users' ones:

- majority vote
- random choice on parts
- put the detected bits equal to 0
- ...

Multilevel approach

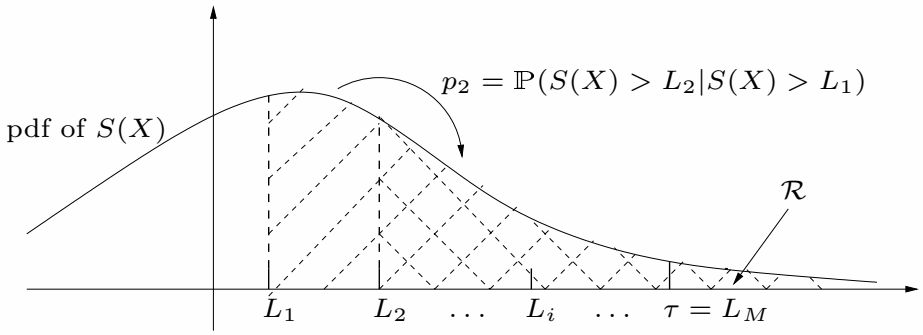


Figure 4: Multilevel

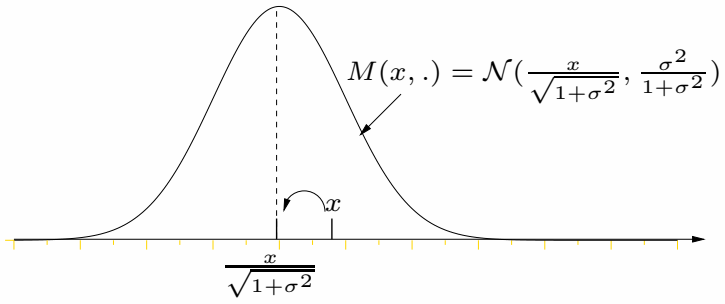
- Ingredients : fix M and $L_1 < \cdots < L_M = \tau$ so that each $p_i = \mathbb{P}(S(X) > L_i | S(X) > L_{i-1})$ is not too small.
- Bayes decomposition : $\alpha = p_1 p_2 \ldots p_M$.
- Unrealistic case : suppose you can estimate each p_i independently with classic Monte-Carlo : $p_i \approx \hat{p}_i = N_i/N$.
- Multilevel Estimator : $\hat{\alpha}_N = \hat{p}_1 \hat{p}_2 \ldots \hat{p}_M$.

The Shaker

- Recall : $X \sim \mu$ on $\mathbb{X}$.
- Ingredient : a μ reversible transition kernel $K(x, dx')$ on $\mathbb{X}$:

$$\forall (x, x') \in \mathbb{X}^2 \qquad \mu(dx)K(x, dx') = \mu(dx')K(x', dx).$$

- Consequence : $\mu K = \mu$.
- Example : if $X \sim \mathcal{N}(0, 1)$ then $X' = \frac{X+\sigma W}{\sqrt{1+\sigma^2}} \sim \mathcal{N}(0, 1)$, i.e. $K(x, dx') \sim \mathcal{N}(\frac{x}{\sqrt{1+\sigma^2}}, \frac{\sigma^2}{1+\sigma^2})(dx')$ is a "good shaker".



Feynman-Kac representation

$A_k = S^{-1}(]L_k, +\infty[)$

$M_k^K(x, dy) = K(x, dy)\mathbb{1}_{A_k}(y) + K(x, A_k^c)\,\delta_x(dy)$

$\mu_k(dx) = \frac{1}{\mu(A_k)}\,\mathbb{1}_{A_k}(x)\,\mu(dx)$ the normalized restriction of μ on A_k

μ_k invariant by M_k^K

X_k Markov chain with initial distribution μ and transitions M_k^K

For every test function φ, for $k \in \{0, \ldots, n\}$, we have the following Feynman-Kac representation

$$\mu_{k+1}(\varphi) = \frac{\mathbb{E}[\varphi(X_k)\prod_{j=0}^{k}\mathbb{1}_{A_{j+1}}(X_j)]}{\mathbb{E}[\prod_{j=0}^{k}\mathbb{1}_{A_{j+1}}(X_j)]}.$$

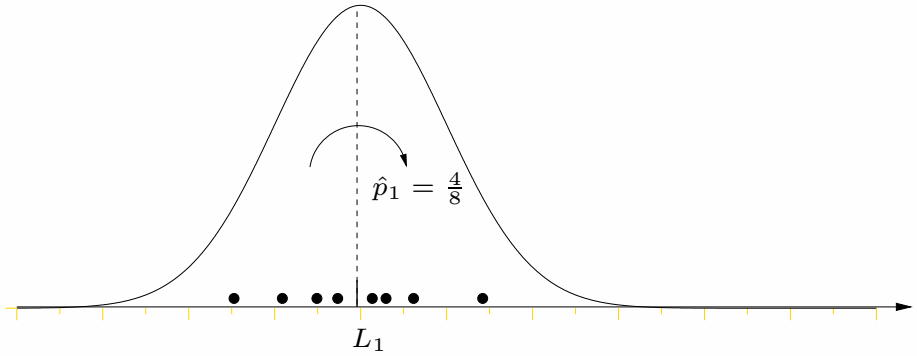
Algorithm

- Initialization : Simulate an i.i.d. sample $\xi_0^1, \ldots, \xi_0^N \sim \mu$.
- Estimate $\hat{p}_1 = \frac{1}{N} \sum \mathbb{1}_{A_1}(\xi_0^1)$
- Selection : $\hat{\xi}_0^i = \xi_0^i$ if $S(\xi_0^i) > L_1$, else pick at random among the N_1 selected particles.
- Mutation : $\tilde{\xi}_0^i \sim M(\hat{\xi}_0^i, dx')$ and

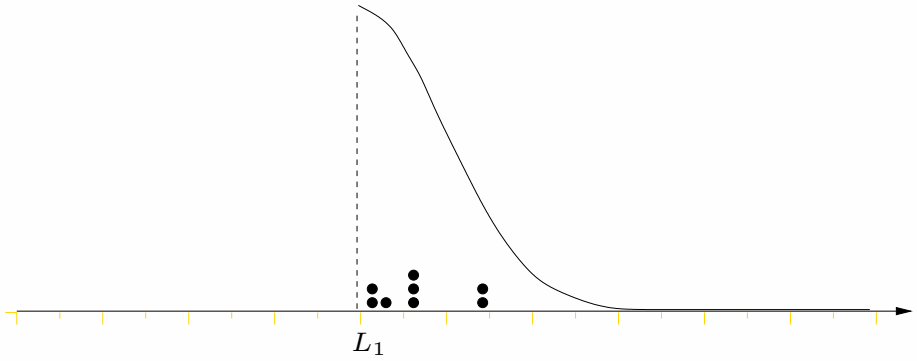
$$\forall i \in \{1, \ldots, N\} \qquad \xi_1^i = \begin{cases} \tilde{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) > L_1 \\ \hat{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) \leq L_1 \end{cases}$$

- Consider next level and iterate until the rare event is reached

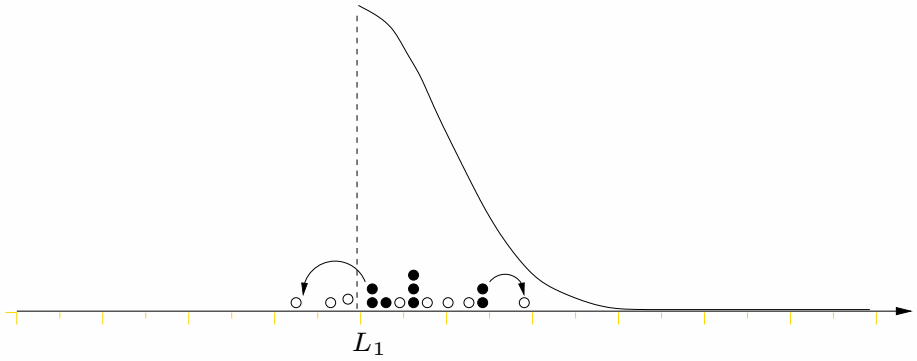
Algorithm



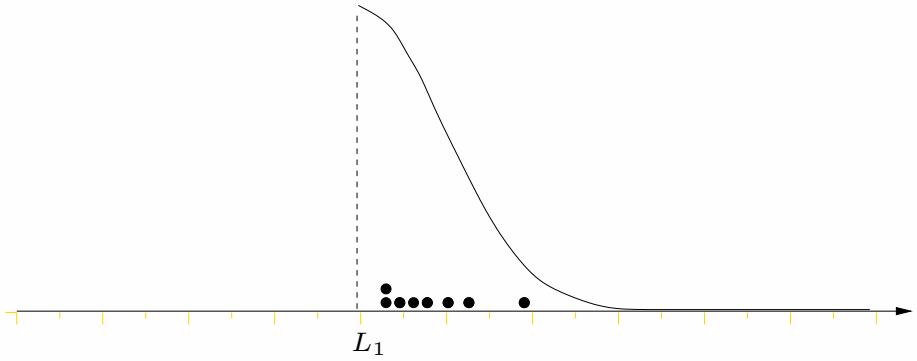
Algorithm



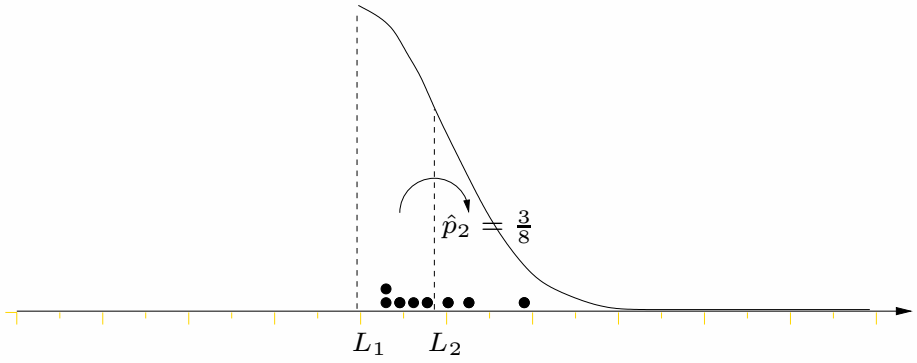
Algorithm



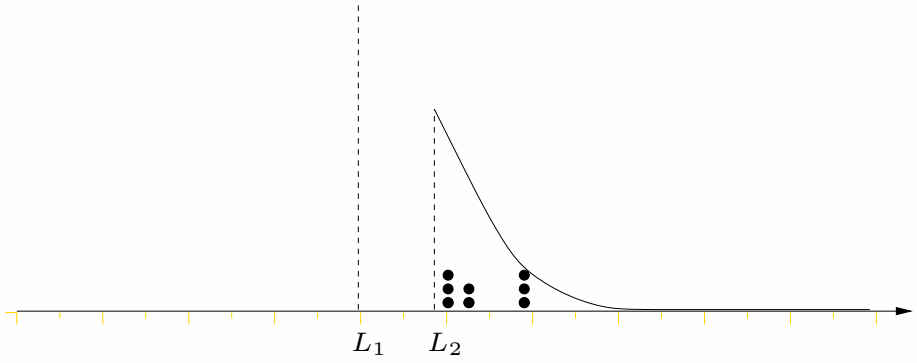
Algorithm



Algorithm



Algorithm



Implementation issues

Choice of K Depends on the model: Metropolis-Hastings, Gibbs sampler, Gaussian case (zero-bit watermarking), i.i.d. on some random sites (Tardos codes)

Trade-off between two drawbacks :

- "shaking effect" too large : most proposed mutations are refused.
- "shaking effect" too small : particles almost don't move.

Levels L_k Adaptive levels with fixed rate of success: p_0 quantile on $S(\xi_k^j)$ to set L_{k+1}, $p_0 = 0.75$ or 0.8 is a good choice

Less dependent sample We can iterate the kernel M_k^K several times to improve the variability of the particles. From well known results on Metropolis-Hastings, the sample is getting more and more independent. Rule: iterate until 90 or 95% have actually moved to an accepted transition

Asymptotic variance

Best achievable asymptotic variance:

- Multilevel Estimator : $\hat{\alpha}_N = \hat{p}_1 \hat{p}_2 \dots \hat{p}_M$.
- Fluctuations : If the $\hat{p}_i$'s are independent, then

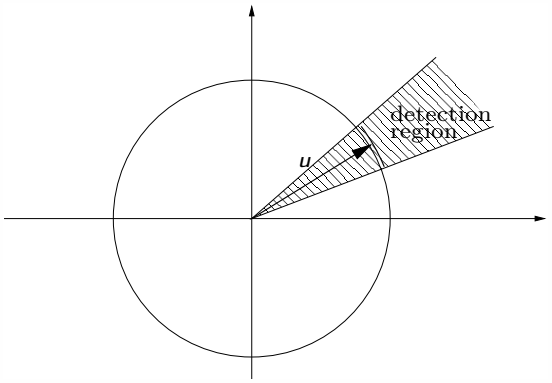
$$\sqrt{N} \cdot \frac{\hat{\alpha}_N - \alpha}{\alpha} \xrightarrow[N \to \infty]{\mathcal{L}} \mathcal{N}(0, \sum_{i=1}^{M} \frac{1 - p_i}{p_i}).$$

- Constrained Minimization :

$$\arg \min_{p_1, \dots, p_M} \sum_{i=1}^{M} \frac{1 - p_i}{p_i} \qquad \text{s.t.} \qquad \prod_{i=1}^{M} p_i = \alpha.$$

- Optimum : $p_1 = \dots = p_M = \alpha^{1/M}$.

Simulations : The Model



- **The model :** $X \sim \mathcal{N}(0, I_{20})$.
- **Rare event :** $\alpha = \mathbb{P}(\frac{\langle X,u \rangle}{\|X\|} > 0.95)$.
- **Numerical computation :** $\alpha = 4.704 \cdot 10^{-11}$.
- **Parameter :** $p = 3/4 \rightsquigarrow \alpha = r \times p^M = .83 \times (3/4)^{82}$.

Numerical results

credit: V. Bahuon

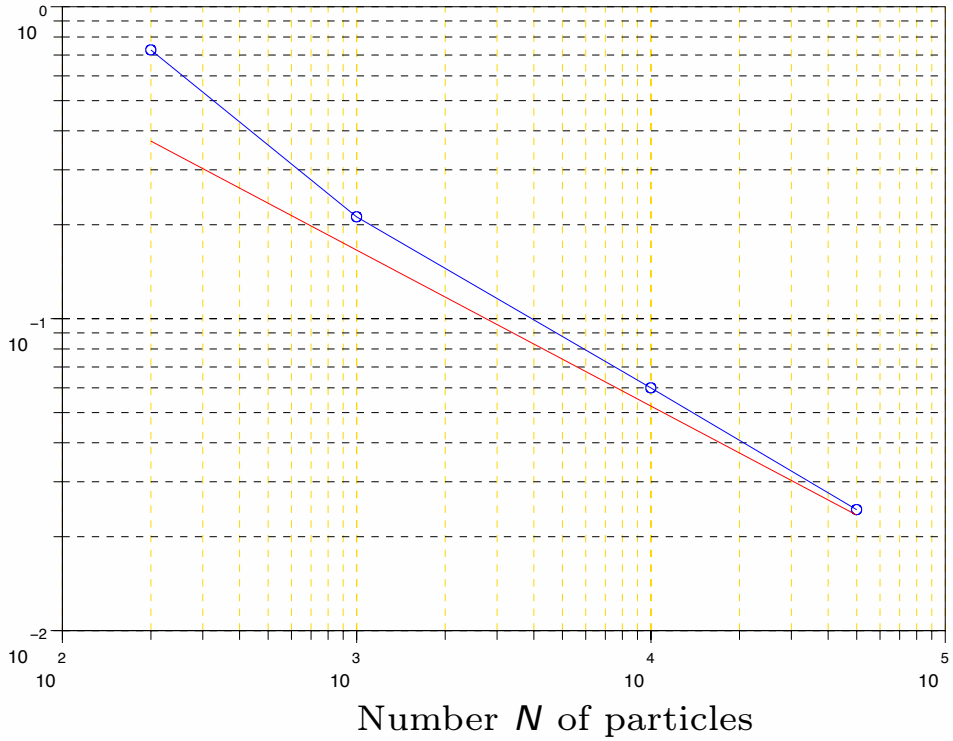


Figure 5: Relative standard deviation.

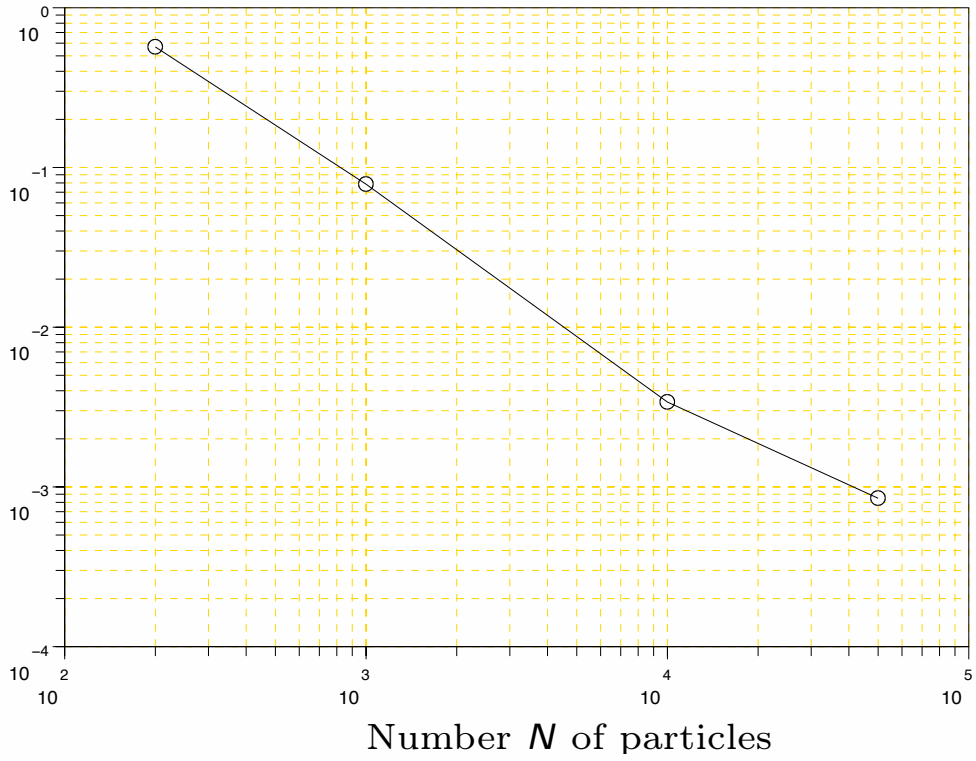


Figure 6: Relative bias.

Perspectives

Find other similar applications (e.g. probabilistic counting algotihms in a large discrete set)

Work in progress: non asymptotic variance estimates