A Multivariate Gaussian Sampler in High Dimensions

Saïd Moussaoui

Joint work with:

Clément Gilavert and Jérôme Idier

Ecole Centrale de Nantes, IRCCyN, CNRS UMR 6597, Nantes, France

said.moussaoui@irccyn.ec-nantes.fr

MCMC workshop

26th November 2014 • Marseille, France

Reference: C. Gilavert, S. Moussaoui, J. Idier, Efficient Gaussian Sampling for Solving Large-Scale Inverse Problems using MCMC, IEEE Trans. Signal Processing, vol. 63, No. 1, pp. 70-80, 2015 (Arxiv preprint:1409.0606).

Introduction

Draw K samples $\{\boldsymbol{x}_k\}_{k=1}^K$, from a N-dimensional Gaussian distribution

$$\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{R})$$

with a mean vector $\boldsymbol{\mu} \in \mathbb{R}^N$ and a symmetric definite positive covariance matrix $\boldsymbol{R} \in \mathbb{R}^{N \times N}$.

⊛ **Classical approach** `[Wold 1948 ; Scheuer and Stoller, 1962]`

1. perform the Cholesky factorization, $\boldsymbol{R} = \boldsymbol{L}_r \boldsymbol{L}_r^{\mathrm{t}}$, $\rightsquigarrow \mathcal{O}(N^3)$
2. draw a sample from a standard Gaussian distribution, $\boldsymbol{\omega} \sim \mathcal{N}(\boldsymbol{0}_N, \boldsymbol{I}_N)$,
3. retain $\boldsymbol{x} = \boldsymbol{L}_r \boldsymbol{\omega} + \boldsymbol{\mu}$.

$\lhd$ *Main purpose: reduce the computation complexity of MCMC algorithms involving repeated application of high-dimensional Gaussian sampling*

Outline

1. Gaussian sampling for a Bayesian inference
    - Context of inverse problems
    - Main approaches
2. Gaussian sampling within the Reversible Jump MCMC framework
    - The Reversible Jump MCMC framework
    - Proposed sampler
3. Sampler cost optimization and its adaptive tuning
    - Targeting an acceptance rate
    - Optimization of the computation cost
4. Concluding remarks

1. Gaussian sampling for a Bayesian inference

1.1 Context of inverse problems

The *observations* $\boldsymbol{y} \in \mathbb{R}^M$ are expressed according to

$$\boldsymbol{y} = \boldsymbol{H}\,\boldsymbol{x} + \boldsymbol{n} \tag{1}$$

with $\boldsymbol{x} \in \mathbb{R}^N$ the *sought variable* and $\boldsymbol{H} \in \mathbb{R}^{M \times N}$ the *observation matrix (convolution, projection, mixing)*

- Gaussian likelihood: $\boldsymbol{y}|(\boldsymbol{x}, \boldsymbol{\mu}_n, \boldsymbol{R}_n) \sim \mathcal{N}(\boldsymbol{H}\boldsymbol{x} + \boldsymbol{\mu}_n, \boldsymbol{R}_n)$,
- Gaussian prior: $\boldsymbol{x}|(\boldsymbol{\mu}_x, \boldsymbol{R}_x) \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{R}_x)$
  - Simple Gaussian model,
  - Gaussian Markov random fields,
  - Hierarchical Gaussian model (scale/location Gaussian mixtures).

- Inference from the posterior distribution,

$$P(\boldsymbol{x}, \Theta|\boldsymbol{y}) \propto P(\boldsymbol{y}|\boldsymbol{x}, \Theta)P(\boldsymbol{x}|\Theta)P(\Theta) \tag{2}$$

with Θ hyperparameter set, $\Theta = \{\boldsymbol{\mu}_x, \boldsymbol{R}_x, \boldsymbol{\mu}_n, \boldsymbol{R}_n\}$

- Gibbs sampler: for $k = 1, \ldots, K$, $\begin{cases} \text{sample } \Theta_k \sim P(\Theta|\boldsymbol{y}, \boldsymbol{x}_{k-1}) \\ \text{sample } \boldsymbol{x}_k \sim P(\boldsymbol{x}|\boldsymbol{y}, \Theta_k) \end{cases}$

According to the Bayesian model,

$$\boldsymbol{x}|(\boldsymbol{y}, \Theta) \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{R} = \boldsymbol{Q}^{-1}\right)$$

with

$$\boldsymbol{Q} = \boldsymbol{H}^{\mathrm{t}}\boldsymbol{R}_n^{-1}\boldsymbol{H} + \boldsymbol{R}_x^{-1}, \quad \rightsquigarrow \boldsymbol{Q} = \boldsymbol{F}^{\mathrm{t}}\boldsymbol{F} \tag{3}$$

$$\boldsymbol{Q}\boldsymbol{\mu} = \boldsymbol{H}^{\mathrm{t}}\boldsymbol{R}_n^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_n) + \boldsymbol{R}_x^{-1}\boldsymbol{\mu}_x. \quad \rightsquigarrow \boldsymbol{Q}\boldsymbol{\mu} = \boldsymbol{b} \tag{4}$$

◁ *The calculation of the distribution involves the precision matrix* $\boldsymbol{Q}$, *instead of* $\boldsymbol{R}$

◁ *A matrix inversion is necessary to apply the classical sampling approach*

◁ *The posterior mean is given as the solution of a linear system depending on* $\boldsymbol{Q}$

⊛ **Solution 1.** *Avoid high-dimensionnal matrix inversion.* [Rue, 2001]

1. perform the Cholesky factorization of $\boldsymbol{Q} = \boldsymbol{L}_q \boldsymbol{L}_q^{\mathrm{t}}$, instead of $\boldsymbol{R}$,

2. draw a sample from a standard Gaussian distribution, $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}_N, \boldsymbol{I}_N)$,

3. solve $\boldsymbol{L}_q \boldsymbol{z} = \boldsymbol{b}$ and get $\boldsymbol{z}$,

4. retain $\boldsymbol{x}$, solution of $\boldsymbol{L}_q^{\mathrm{t}} \boldsymbol{x} = \boldsymbol{z} + \boldsymbol{\omega}$.

⊛ **Solution 2.** *Perturbation-Optimization* [Orieux et al., 2012 ; Lalanne 2001]

1. draw a sample from a standard Gaussian distribution, $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}_{M+N}, \boldsymbol{I}_{M+N})$,

2. get a sample from a Gaussian distribution, $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{Q}\boldsymbol{\mu}, \boldsymbol{Q})$, according to $\boldsymbol{\eta} = \boldsymbol{F}^{\mathrm{t}}\boldsymbol{\omega} + \boldsymbol{Q}\boldsymbol{\mu}$,

3. retain $\boldsymbol{x}$, solution of $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{\eta}$.

$\lhd$ *The complexity of both solutions (1 and 2) is $\mathcal{O}(N^3)$ unless matrix $\boldsymbol{Q}$ exhibits an exploitable structure,*

$\lhd$ *Matrix $\boldsymbol{Q}$ depends on Θ and, thus, varies during Gibbs sampler iterations.*

$\circledast$ **Practical alternative.** [Bardsley, 2010 ; Papandreou et al., 2010 ; Tan et al., 2010]

Numerical complexity reduction by applying an early stopped iterative solver (conjugate gradient) in Step 3.

$\triangleleft$ *Question 1. Correctness of the sampler?*

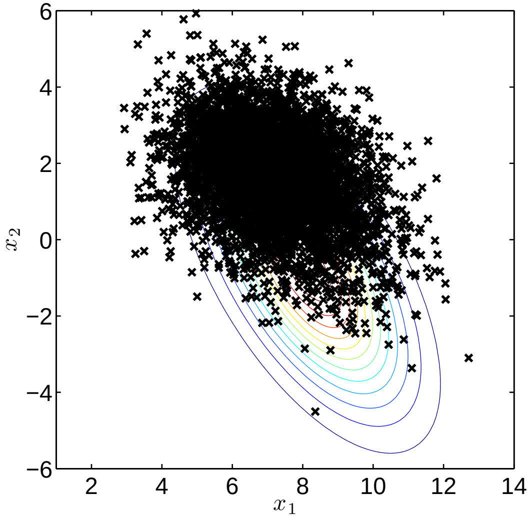
$\triangleleft$ *Question 2. Choice of the truncation level?*

$\lhd$ *Approximate resolution induces an incorrect sampling !*

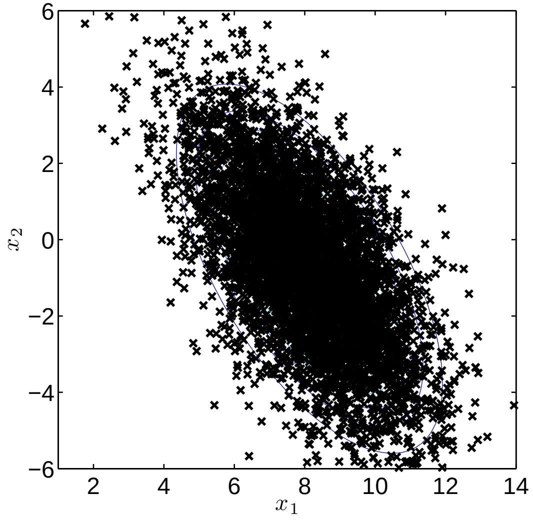
Let $\boldsymbol{Q}$ and $\boldsymbol{\mu}$ be defined by

$$\boldsymbol{Q} = \boldsymbol{R}^{-1} \text{ with } R_{ij} = \sigma^2 \rho^{|i-j|} \text{ and } \mu_i \sim \mathcal{U}[0, 10], \quad (\forall i, j = 1, \dots, N) \qquad (5)$$

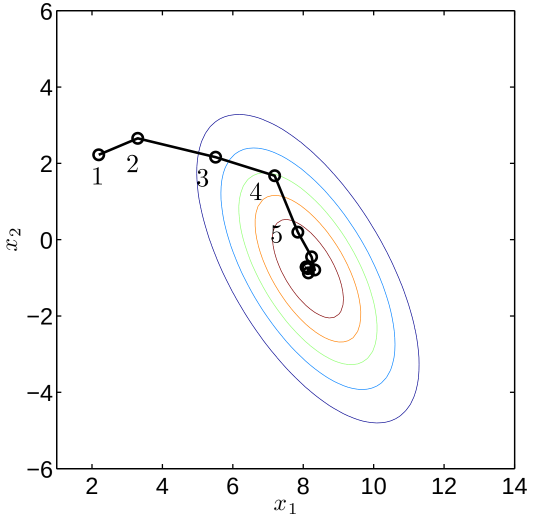
with $N = 20$, $\sigma^2 = 1$ and $\rho = 0.8$. Draw $K = 10,000$ samples.



$J = 4$ CG iterations



$J = 10$ CG iterations



Sample mean

Our proposal: Introduce an accept-reject step to correct this behavior.

2. Sampling within the Reversible Jump MCMC framework

2.1 Reversible Jump MCMC [Green, 1995 ; Waagepetersen and Sorensen, 2001]

- Construct a Markov chain whose distribution asymptotically converges to the target distribution $P_{\boldsymbol{X}}(\cdot)$.

- Introduce an auxiliary variable $\boldsymbol{z} \in \mathbb{R}^L \sim f_{\boldsymbol{Z}}(\boldsymbol{z}|\boldsymbol{x}_{\text{old}})$ and define a differentiable transformation $\phi(\cdot)$

$$\boldsymbol{\phi} : \left(\mathbb{R}^N \times \mathbb{R}^L\right) \mapsto \left(\mathbb{R}^N \times \mathbb{R}^L\right)$$
$$(\boldsymbol{x}_{\text{old}}, \boldsymbol{z}) \mapsto (\boldsymbol{x}, \boldsymbol{s})$$

that must be reversible $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{s}) = (\boldsymbol{x}_{\text{old}}, \boldsymbol{z})$.

- The transition from $\boldsymbol{x}_{\mathrm{old}}$ to $\boldsymbol{x}_{\mathrm{new}}$ is governed by an acceptance probability

$$\alpha(\boldsymbol{x}_{\mathrm{old}}, \boldsymbol{x}) = \min\left(1, \frac{P_{\boldsymbol{X}}(\boldsymbol{x}) P_{\boldsymbol{Z}}(\boldsymbol{s}|\boldsymbol{x})}{P_{\boldsymbol{X}}(\boldsymbol{x}_{\mathrm{old}}) P_{\boldsymbol{Z}}(\boldsymbol{z}|\boldsymbol{x}_{\mathrm{old}})} |J_{\boldsymbol{\phi}}(\boldsymbol{x}_{\mathrm{old}}, \boldsymbol{z})|\right).$$

where $J_{\boldsymbol{\phi}}(\cdot)$ is the Jacobian determinant of $\boldsymbol{\phi}(\cdot)$.

2.2 Gaussian sampling within RJMCMC

To sample from a Gaussian distribution $\boldsymbol{x} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{Q}^{-1}\right)$,

- An auxiliary variable $\boldsymbol{z} \in \mathbb{R}^N$ is sampled from

$$P_{\boldsymbol{Z}}(\boldsymbol{z}|\boldsymbol{x}_{\text{old}}) = \mathcal{N}\left(\boldsymbol{A}\boldsymbol{x}_{\text{old}} + \boldsymbol{c}, \boldsymbol{B}\right). \tag{6}$$

The choice of $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, $\boldsymbol{B} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{c} \in \mathbb{R}^{N\times N}$ will be discussed later.

- The deterministic move is performed using the transformation $\boldsymbol{\phi}(\cdot)$, such that

$$\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{s} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\phi}_1(\boldsymbol{x}_{\text{old}}, \boldsymbol{z}) \\ \boldsymbol{\phi}_2(\boldsymbol{x}_{\text{old}}, \boldsymbol{z}) \end{pmatrix} = \begin{pmatrix} -\boldsymbol{x}_{\text{old}} + \boldsymbol{f}(\boldsymbol{z}) \\ \boldsymbol{z} \end{pmatrix}, \tag{7}$$

with functions $\left(\boldsymbol{f} : \mathbb{R}^N \mapsto \mathbb{R}^N\right)$, $\left(\boldsymbol{\phi}_1 : (\mathbb{R}^N \times \mathbb{R}^N) \mapsto \mathbb{R}^N\right)$ and $\left(\boldsymbol{\phi}_2 : (\mathbb{R}^N \times \mathbb{R}^N) \mapsto \mathbb{R}^N\right)$.

$\lhd$ *f($\boldsymbol{z}$) must be independent from $\boldsymbol{x}_{old}$ to ensure the reversibility condition*

⊛ **Proposition** [Gilavert 2014]

Let an auxiliary variable $\boldsymbol{z}$ *be obtained according to* (6) *and a proposed sample* $\boldsymbol{x}$ *according to the transformation defined by* (7)*. Then the acceptance probability is*

$$\alpha(\boldsymbol{x}_{old}, \boldsymbol{x}) = \min\left(1, e^{-\boldsymbol{r}(\boldsymbol{z})^{\mathrm{t}}(\boldsymbol{x}_{old}-\boldsymbol{x})}\right),$$

with $\boldsymbol{r}(\boldsymbol{z}) = \boldsymbol{Q}\boldsymbol{\mu} + \boldsymbol{A}^{\mathrm{t}}\boldsymbol{B}^{-1}(\boldsymbol{z}-\boldsymbol{c}) - \frac{1}{2}\left(\boldsymbol{Q} + \boldsymbol{A}^{\mathrm{t}}\boldsymbol{B}^{-1}\boldsymbol{A}\right)\boldsymbol{f}(\boldsymbol{z})$.

In particular, the acceptance probability equals one when $\boldsymbol{f}(\boldsymbol{z})$ *is the exact solution of the linear system*

$$\frac{1}{2}\left(\boldsymbol{Q} + \boldsymbol{A}^{\mathrm{t}}\boldsymbol{B}^{-1}\boldsymbol{A}\right)\boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{Q}\boldsymbol{\mu} + \boldsymbol{A}^{\mathrm{t}}\boldsymbol{B}^{-1}(\boldsymbol{z}-\boldsymbol{c}). \qquad (8)$$

⊛ **Proof.** See paper [Gilavert 2015].

⊛ Consequence

Setting $\boldsymbol{A} = \boldsymbol{B} = \boldsymbol{Q}$ and $\boldsymbol{c} = \boldsymbol{Q}\boldsymbol{\mu}$

- defines $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{Q}\boldsymbol{x}_{\text{old}} + \boldsymbol{Q}\boldsymbol{\mu}, \boldsymbol{Q})$, which can also be expressed as $\boldsymbol{z} = \boldsymbol{Q}\boldsymbol{x}_{\text{old}} + \boldsymbol{\eta}$,

- simplifies equation (8) to a linear system $\boldsymbol{Q}\boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{z}$,

- cancels the correlation between successive samples when the acceptance probability equals one,

- by substituting $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z}) - \boldsymbol{x}_{\text{old}}$ in (8), the latter becomes $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{\eta}$.

2.3 Reversible Jump Perturbation-Optimization algorithm

1. Sample $\boldsymbol{\eta} \sim \mathcal{N}\left(\boldsymbol{Q}\boldsymbol{\mu}, \boldsymbol{Q}\right)$.

2. Solve the linear system $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{\eta}$ in an approximate way. Let $\widehat{\boldsymbol{x}}$ denote the obtained solution and $\boldsymbol{r}(\boldsymbol{z}) = \boldsymbol{\eta} - \boldsymbol{Q}\widehat{\boldsymbol{x}}$.

3. With probability $\min\left(1, e^{-\boldsymbol{r}(\boldsymbol{z})^{\mathrm{t}}(\boldsymbol{x}_{\mathrm{old}} - \widehat{\boldsymbol{x}})}\right)$, set $\boldsymbol{x}_{\mathrm{new}} = \widehat{\boldsymbol{x}}$, otherwise set $\boldsymbol{x}_{\mathrm{new}} = \boldsymbol{x}_{\mathrm{old}}$.

$\triangleleft$ *To ensure reversibility of the deterministic move, the initial point $\boldsymbol{x}_0$ of the solver must be such that $\boldsymbol{u}_0 = \boldsymbol{x}_0 + \boldsymbol{x}_{old}$ does not depend on $\boldsymbol{x}_{old}$.*

$\triangleleft$ *Consequence: setting $\boldsymbol{x}_0 = \boldsymbol{0}$ or $\boldsymbol{x}_0 = \boldsymbol{x}_{old}$ is not allowed, while $\boldsymbol{x}_0 = -\boldsymbol{x}_{old}$ is the default choice corresponding to $\boldsymbol{u}_0 = \boldsymbol{0}$.*

⊛ Comparison with the T-PO algorithm

Similarly to the T-PO, the proposed RJPO algorithm relies on the approximate resolution of the same linear system $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{\eta}$, but with two additional features:

- An accept-reject strategy to ensure the sampler convergence,
- An initial point $\boldsymbol{x}_0$ of the linear solver such that $\boldsymbol{x}_0 + \boldsymbol{x}_{\text{old}}$ does not depend on $\boldsymbol{x}_{\text{old}}$.

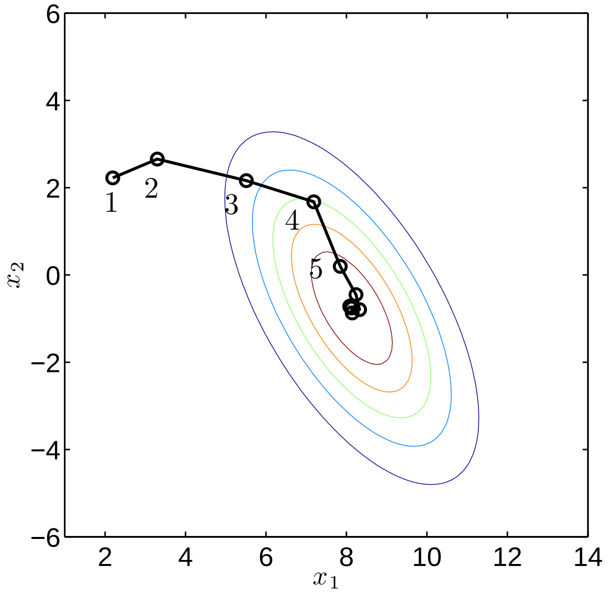
⊛ Implementation issues

- The linear conjugate gradient algorithm is used for the system resolution since it permits a matrix-free implementation with reduced memory requirements,
- A stopping rule based on a threshold on the relative residual norm is applied:

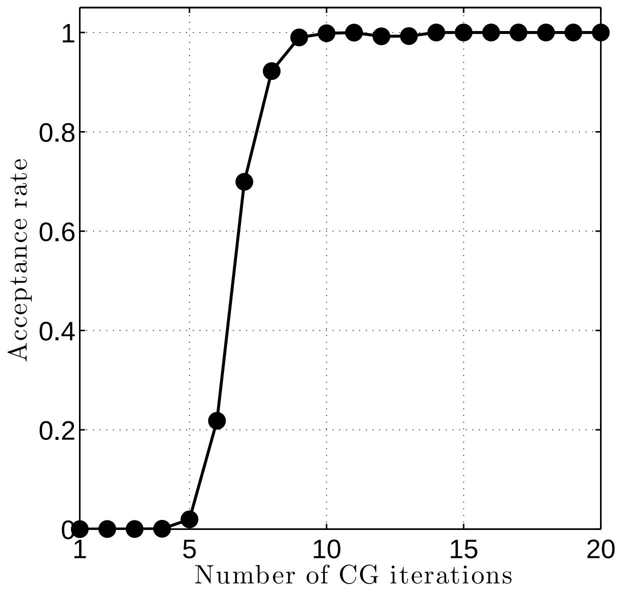
$$\epsilon = \frac{\|\boldsymbol{\eta} - \boldsymbol{Q}\boldsymbol{x}\|_2}{\|\boldsymbol{\eta}\|_2}.$$

1) Application to the toy example

⊛ Acceptance probability



Sample mean



acceptance rate

◁ *The acceptance rate curve indicates a required minimal number of CG iterations,*

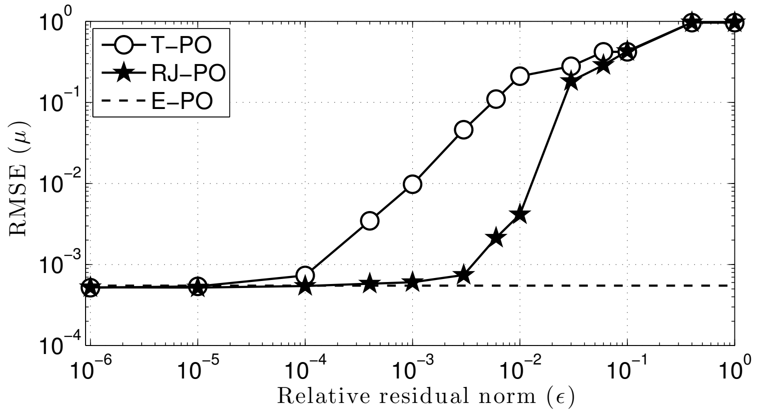
◁ *How to choose the appropriate truncation level to maximize efficiency?*

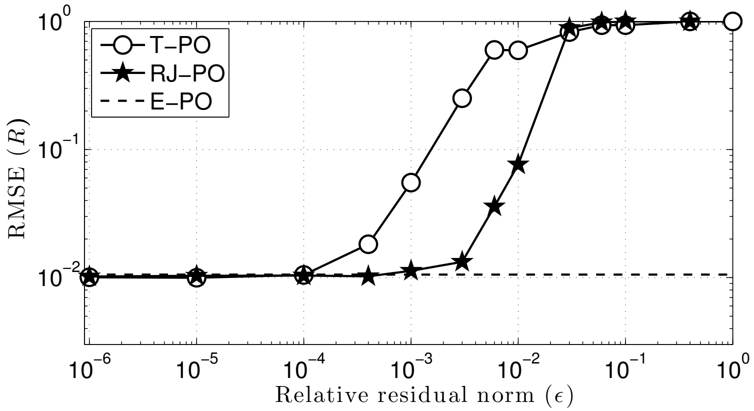
2) Influence of the relative residual norm

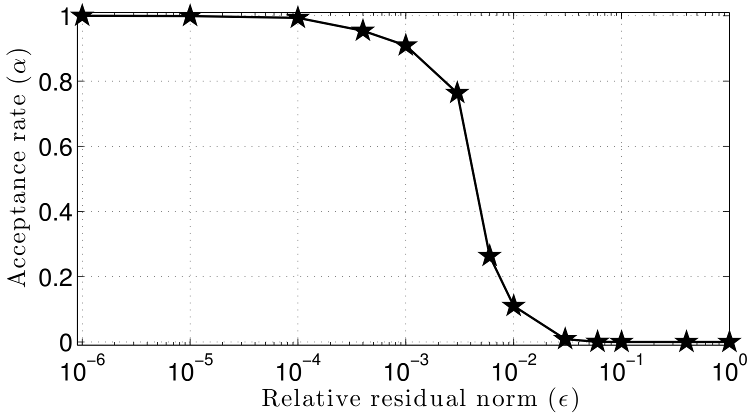
⊛ Estimation error

$$\mathrm{RMSE}(\boldsymbol{\mu}) = \frac{\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2}{\|\boldsymbol{\mu}\|_2} \quad \text{and} \quad \mathrm{RMSE}(\boldsymbol{R}) = \frac{\|\boldsymbol{R} - \hat{\boldsymbol{R}}\|_F}{\|\boldsymbol{R}\|_F},$$

where $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{R}}$ the empirical estimates using 10^5 generated Markov chain samples.

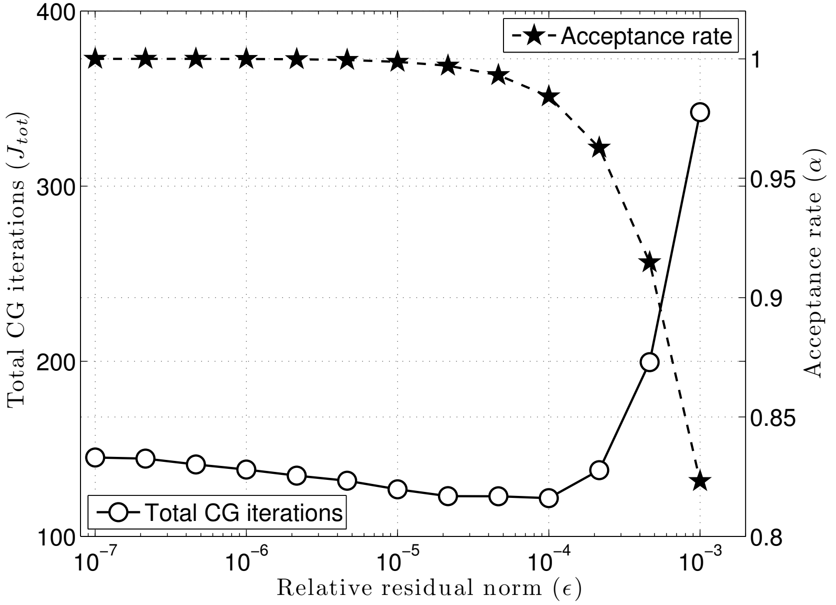


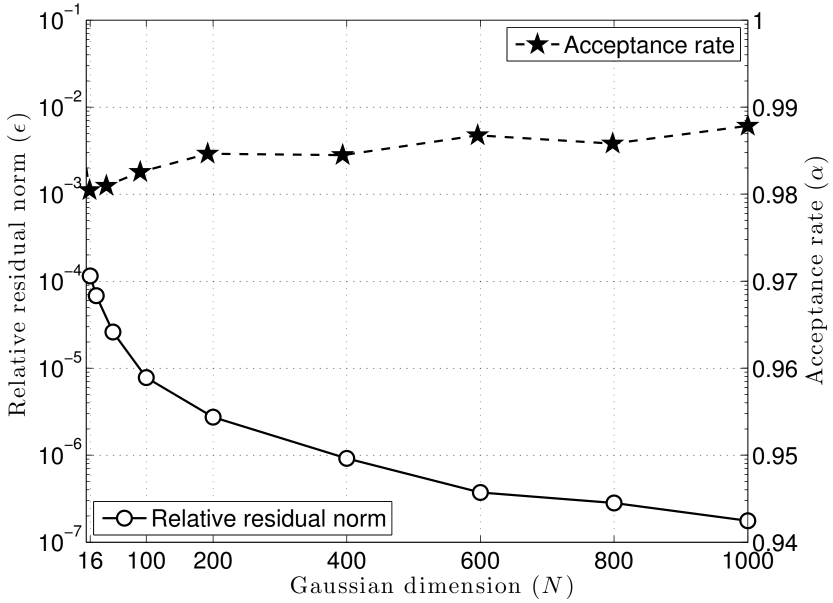




⊛ Convergence diagnosis and computation cost

- Assess the total (cumulated) number of conjugate gradient iterations before convergence (diagnosis based on Gelman-Rubin convergence criterion on 100 parallel chains)





⊲ *A lower acceptance rate induces a higher number of iterations due to slow convergence.*

⊲ *A minimal cost can be reached and it corresponds to an acceptance rate of almost one.*

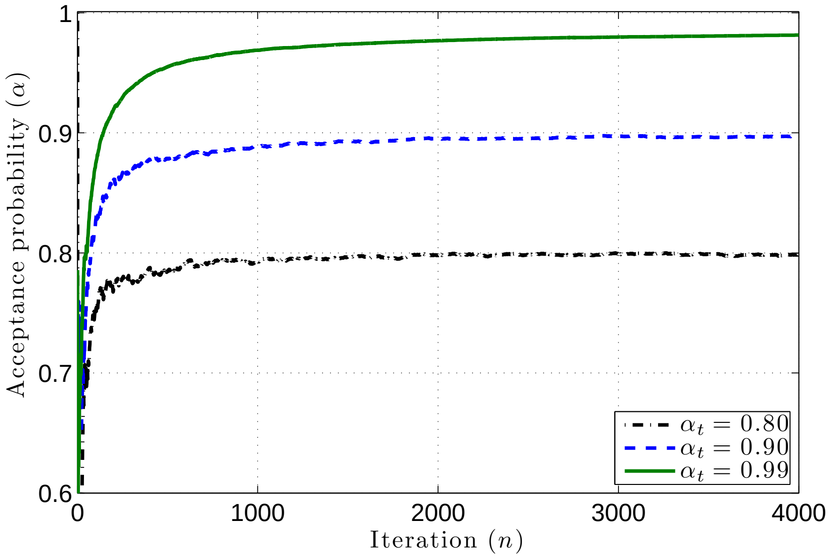
3. Adaptive tuning of the Gaussian sampler

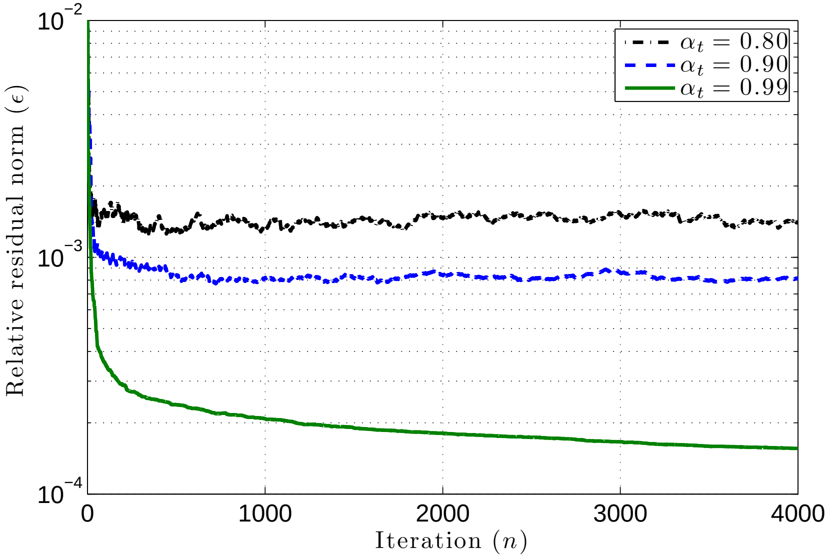
3.1 Targeting a predefined acceptance rate

Adjust recursively the relative residual norm ϵ to achieve a desired acceptance rate α_t using a stochastic approximation procedure [Andrieu and Robert 2001]

$$\log(\epsilon_{k+1}) = \log(\epsilon_k) + \gamma_k \left[\alpha(\boldsymbol{x}_k, \boldsymbol{x}) - \alpha_t\right] \tag{9}$$

with $\gamma_k = K_0\, k^{-\beta}$. (Take $K_0 = 1$ and $\beta = 0.5$.)





3.2 Minimization of the computation cost per effective sample

1) Statistical efficiency

- Effective sample size (ESS) [Goodman and Sokal, 1989]: number n_{eff} of independent samples, that would yield the same estimation variance in approximating the Bayesian estimator as $n_{\max}$ samples of the simulated chain:

$$n_{\text{eff}} = \frac{n_{\max}}{1 + 2 \sum_{k=1}^{\infty} \rho_k} \tag{10}$$

where ρ_k the autocorrelation coefficient at lag k. For a first-order autoregressive chain, $\rho_k = \rho^k$,

$$n_{\text{eff}} = n_{\max} \frac{1-\rho}{1+\rho} \Longrightarrow \text{ESS Ratio} = \frac{n_{\text{eff}}}{n_{\max}}. \tag{11}$$

- It defines how many iterations $n_{\max}$ are needed for each resolution accuracy in order to get chains having the same effective sample size.

2) Computation cost per effective sample

We propose to define the *computing cost per effective sample* (CCES) as

$$\mathrm{CCES} = \frac{J_{\mathrm{tot}}}{n_{\mathrm{eff}}} = J \cdot \frac{1+\rho}{1-\rho}. \tag{12}$$

where $J = J_{\mathrm{tot}}/n_{\max}$ is the average number of CG iterations per sample.

- The chain correlation ρ is an implicit function of the acceptance rate α. It has two terms:
  - With a probability $(1-\alpha)$, the accept-reject procedure produces identical ($i.e.$, maximally correlated) samples in case of rejection.
  - In case of acceptance, the new sample is slightly correlated with the previous one, because of the early stopping of the CG algorithm.

- The correlation induced in the case of acceptance is negligible compared to the correlation induced by rejection.

Thus, $\rho = (1 - \alpha)$ and

$$\mathrm{CCES} = J\,\frac{1+\rho}{1-\rho} = \frac{2-\alpha}{\alpha}$$

- The best tuning of the relative residual norm leading to the lowest CCES satisfies

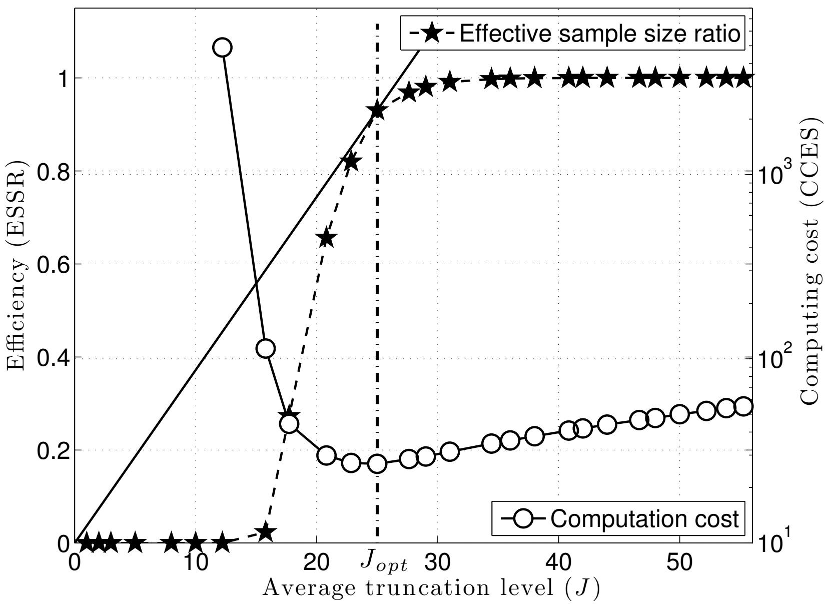
$$J\,\frac{d\alpha}{dJ} - \alpha + \frac{\alpha^2}{2} = 0.$$

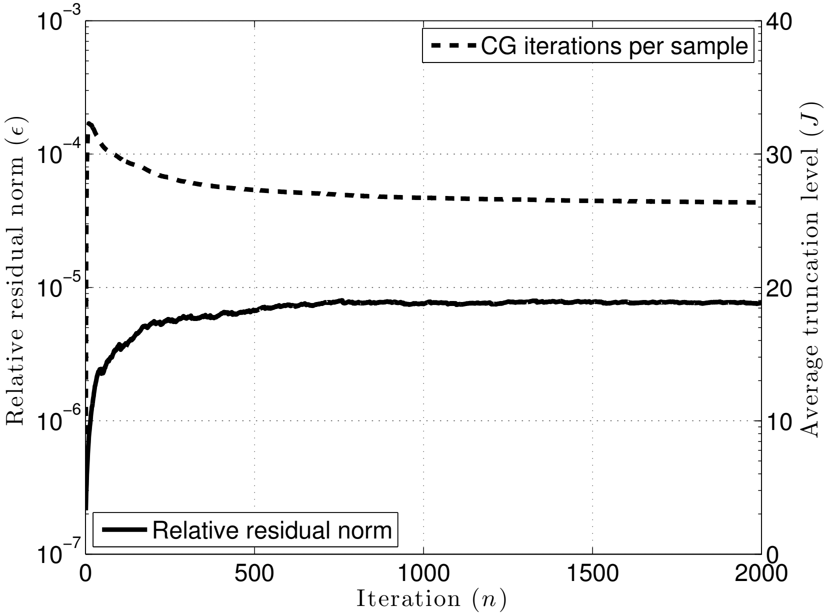
$\triangleleft$ *The solution can not be calculated analytically.*

⊛ Adaptive tuning

The stochastic approximation procedure is now applied to adaptively adjust the optimal value of ϵ,

$$\log \epsilon_{k+1} = \log \epsilon_k + \gamma_k \left(J_k \frac{d\alpha_k}{dJ} - \alpha_k + \frac{\alpha_k^2}{2} \right), \tag{13}$$

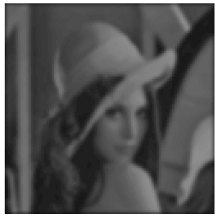




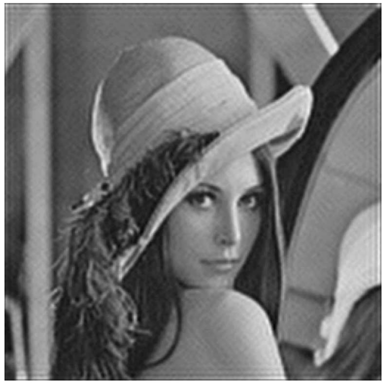
◁ *There is no need to define the target acceptance probability.*

3.3 Example of image superresolution

Original image



One observation



Reconstructed image

Having 5 images of size 128×128 pixels ($M = 81920$) we reconstruct the original one of size 256×256 ($N = 65536$ pixels).

- The noise is assumed zero mean, Gaussian with an unknown precision matrix $\boldsymbol{Q}_n = \gamma_n \boldsymbol{I}$,
- A zero-mean Gaussian distribution $\boldsymbol{x}$, with a precision matrix $\boldsymbol{Q}_x = \gamma_x \boldsymbol{D}^{\mathrm{t}} \boldsymbol{D}$ assigned to $\boldsymbol{x}$, with $\boldsymbol{D}$ a circulant convolution matrix associated to a Laplacian filter.

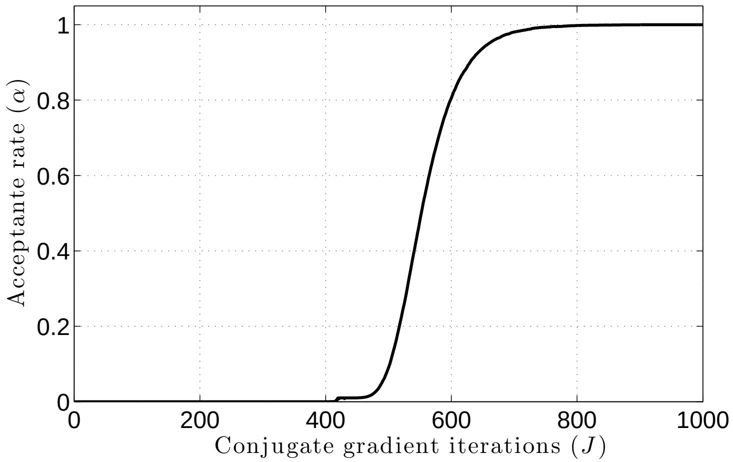
- Non-informative Jeffrey's priors are assigned to the two hyper-parameters γ_n and γ_x.

- A Gibbs sampler is run for 1000 iterations and a burn-in period of 100 iterations.

⊛ **Sampled posterior statistics**. Mean (standard deviation)

	γ_n	$\gamma_x \times 10^{-4}$	x_i
Cholesky	102.1 (0.56)	6.1 (0.07)	104.6 (9.06)
T-PO $\epsilon = 10^{-4}$	0.3 (0.06)	$45\ (0.87)$	102.2 (3.30)
T-PO $\epsilon = 10^{-6}$	6.8 (0.04)	$32\ (0.22)$	104.8 (2.34)
T-PO $\epsilon = 10^{-8}$	71.7 (0.68)	$21\ (0.29)$	102.7 (2.51)
RJPO $\alpha_t = 0.99$	101.2 (0.55)	$6.1\ (0.07)$	101.9 (8.89)

⊛ Acceptance rate



⊛ Computation time and memory usage

- The computation time per sample, on a Intel Core i7-3770 with 8 GB of RAM and a 64bit system :
  - Cholesky sampler: 20.3s and the required memory is about 6 GB
  - RJPO algorithm: 15.1s and the memory usage is less than 200 MB
- This last result is due to the use of a conjugate gradient on which each matrix-vector product is performed without explicitly writing the matrix $\boldsymbol{Q}$.

4. Conclusions

- Convergent multivariate Gaussian sampling suitable for high-dimensional problems

- Empirical analysis of the statistical efficiency,

- Set an adaptive tuning allowing to optimize the computation cost.

⊛ **Open questions**

- Establish a link between the proposed strategy and random walk Metropolis Hastings (RWMH), Metropolis Adjusted Langevin Algorithm (MALA)?

- Use of the computation cost per effective sample for an adaptive scaling of RWMH and MALA?