Efficiently Approximating GP Emulators with Inequality Constraints using MC/MCMC

Andrés F. López-Lopera[1], François Bachoc[2], Nicolas Durrande[1,3], and Olivier Roustant[1]

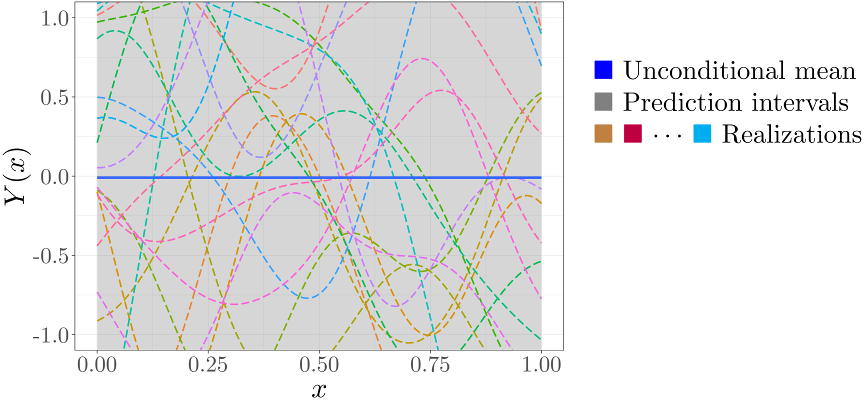
[1]École des Mines de Saint-Étienne (EMSE), France.
[2]Institut de Mathématiques de Toulouse (IMT), France.
[3]PROWLER.io, Cambridge, UK.

This work is funded by the chair of applied mathematics OQUAIDO.

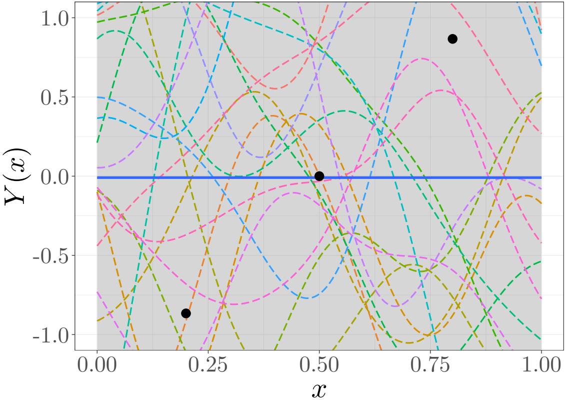
13th MCQMC

July 5, 2018

Gaussian process regression models: motivation



$$Y \sim \mathcal{GP}(0, k(x, x'))$$



$Y \sim \mathcal{GP}(0, k(x, x'))$

Unconditional mean

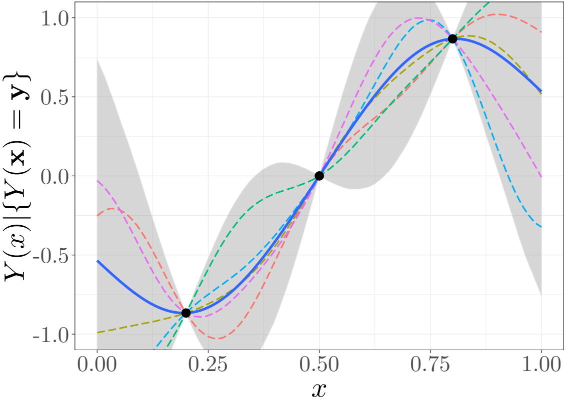
Prediction intervals

$\cdots$ Realizations

● Interpolation conditions

$$\mathcal{D} = (x_i, y_i)_{i=1}^n = (\mathbf{x}, \mathbf{y})$$

Gaussian process regression models: motivation



Unconditional mean

Prediction intervals

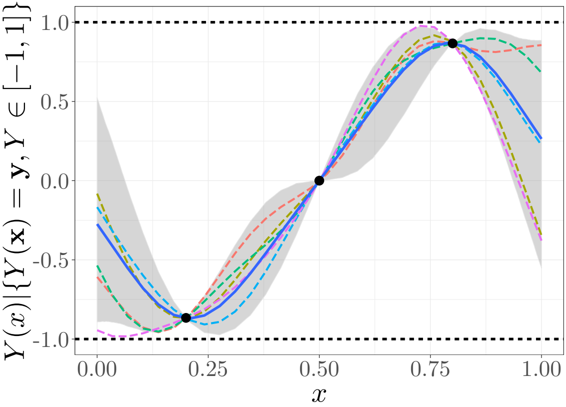
$\cdots$ Realizations

- Interpolation conditions

$$\mathcal{D} = (x_i, y_i)_{i=1}^{n} = (\mathbf{x}, \mathbf{y})$$

$$Y|\{Y(\mathbf{x}) = \mathbf{y}\} \sim \mathcal{GP}(m(x), c(x, x'))$$

Gaussian process regression models: motivation



Unconditional mean

Prediction intervals

· · · Realizations

● Interpolation conditions

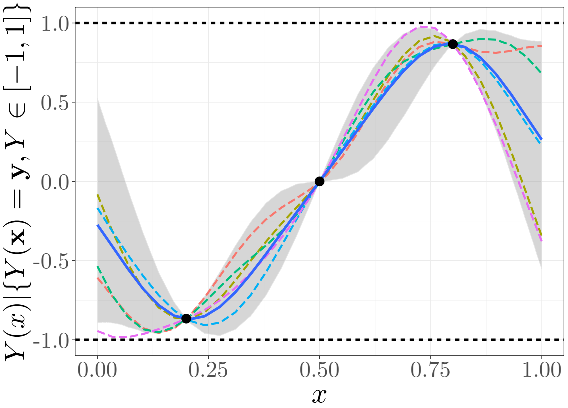
$$\mathcal{D} = (x_i, y_i)_{i=1}^n = (\mathbf{x}, \mathbf{y})$$

- - Boundedness condition

$$Y \in [-1, 1]$$

$$Y | \{Y(\mathbf{x}) = \mathbf{y}, Y \in [-1,1]\} \not\sim \mathcal{GP}(\mu_c(x), c_c(x, x'))$$

Gaussian process regression models: motivation



Unconditional mean

Prediction intervals

$\cdots$ Realizations

● Interpolation conditions

$$\mathcal{D} = (x_i, y_i)_{i=1}^n = (\mathbf{x}, \mathbf{y})$$

- - Boundedness condition

$$Y \in [-1, 1]$$

$$Y|\{Y(\mathbf{x}) = \mathbf{y}, Y \in [-1,1]\} \not\sim \mathcal{GP}(\mu_c(x), c_c(x, x'))$$

$\Rightarrow$ But it can be approximated efficiently via MC or MCMC!

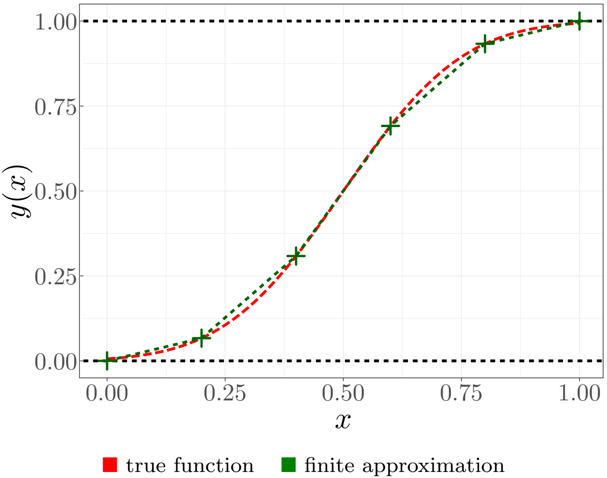
Table of Contents

Table of Contents

Table of Contents

Finite-dimensional Gaussian approximation

Finite representation: also bounded and monotonic.



Imposing constraints on the knots is enough (Maatouk and Bay, 2017).

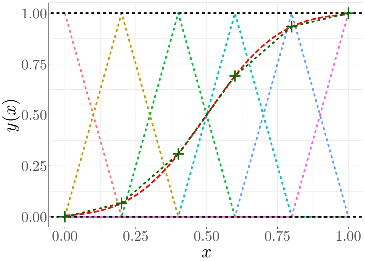
Finite-dimensional Gaussian approximation

Let the finite-dimensional Gaussian approximation be defined as

$$Y_m(x) = \sum_{j=1}^{m} \xi_j \phi_j(x), \text{ s.t. } \begin{cases} Y_m(x_i) = y_i & \text{(interpolation conditions)}, \\ \boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u} & \text{(linear inequality conditions)}, \end{cases}$$

where

- $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_m]^\top \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}_\theta)$ with covariance matrix $\boldsymbol{\Gamma}_\theta$; and
- $\phi_j : [0, 1] \to \mathbb{R}$ are hat functions as in (Maatouk and Bay, 2017):



Finite-dimensional Gaussian approximation

Then, since linearity preserves Gaussian distributions, quantifying uncertainty on Y_m relies on simulating the truncated Gaussian vector

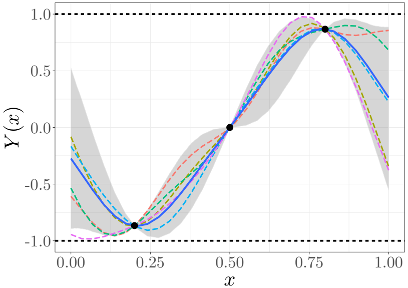
$$\boldsymbol{\Lambda}\boldsymbol{\xi} | \{\boldsymbol{\Phi}\boldsymbol{\xi} = \mathbf{y}, \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u}\} \sim \mathcal{TN}(\boldsymbol{\Lambda}\boldsymbol{\mu}, \boldsymbol{\Lambda}\boldsymbol{\Sigma}\boldsymbol{\Lambda}^\top, \boldsymbol{l}, \boldsymbol{u}), \qquad (1)$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean vector and covariance matrix of the conditional distribution $\boldsymbol{\xi} | \{\boldsymbol{\Phi}\boldsymbol{\xi} = \mathbf{y}\} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$:

$$\boldsymbol{\mu} = \boldsymbol{\Gamma}\boldsymbol{\Phi}^\top (\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^\top)^{-1} \mathbf{y},$$
$$\boldsymbol{\Sigma} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma}\boldsymbol{\Phi}^\top (\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^\top)^{-1} \boldsymbol{\Phi}\boldsymbol{\Gamma}.$$

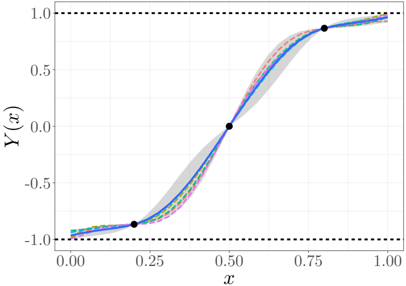
$\Rightarrow$ Posterior distribution (1) can be approximated via MC/MCMC.

1D example under boundedness constraints



$$
\underbrace{\begin{bmatrix} -1 \\ -1 \\ -1 \\ \vdots \\ -1 \\ -1 \end{bmatrix}}_{\boldsymbol{l}}
\leq
\underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}}_{\boldsymbol{\Lambda}}
\underbrace{\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \vdots \\ \xi_{m-1} \\ \xi_m \end{bmatrix}}_{\boldsymbol{\xi}}
\leq
\underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}}_{\boldsymbol{u}}
$$

1D example under boundedness & monotonicity constraints



$$\underbrace{\begin{bmatrix} -1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}}_{\boldsymbol{l}} \leq \underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}}_{\boldsymbol{\Lambda}} \underbrace{\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \vdots \\ \xi_{m-1} \\ \xi_m \end{bmatrix}}_{\boldsymbol{\xi}} \leq \underbrace{\begin{bmatrix} 1 \\ \infty \\ \infty \\ \vdots \\ \infty \\ 1 \end{bmatrix}}_{\boldsymbol{u}}$$

Table of Contents

MINES Saint-Étienne

INSTITUT DE MATHÉMATIQUES DE TOULOUSE

Table of Contents

MCQMC-2014: Maatouk and Bay (2016) proposed a rejection sampler using the mode of $\boldsymbol{\xi}$ (MAP solution). Let $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$, then

$$\boldsymbol{\mu}^* = \underset{\boldsymbol{\xi} \in C}{argmin} \ \frac{1}{2} \boldsymbol{\xi}^\top \boldsymbol{\Gamma}^{-1} \boldsymbol{\xi},$$

where $\mathcal{C} = \{\mathbf{c} \in \mathbb{R}^m; \ \forall \ k = 1, \ldots, q \ : \ \ell_k \leq \sum_{j=1}^m \lambda_{k,j} c_j \leq u_k\}$. Then, a valid proposal pdf for rejection sampling (RS) on $\mathcal{C}$ is

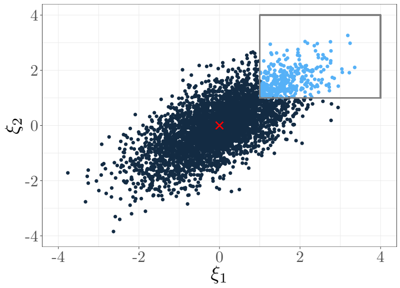
$$g(\boldsymbol{\xi} | \boldsymbol{\mu}^*, \boldsymbol{\Gamma}) = \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Gamma}).$$

Rejection sampling from the mode (RSM)

$$\begin{bmatrix}\xi_1\\ \xi_2\end{bmatrix} \sim \mathcal{TN}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1.0 & 0.7\\ 0.7 & 1.0\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}4\\4\end{bmatrix}\right).$$

Number of simulations: 10^3

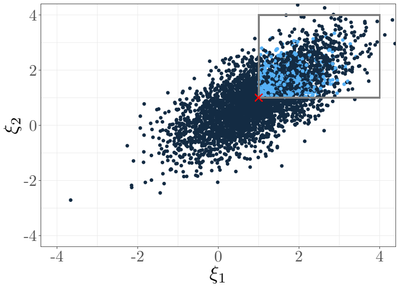
Standard RS



$\mu = [0, 0]$

Acceptance rate: 9%

RSM



$\mu^*_{\mathrm{RSM}} = [1, 1]$

Acceptance rate: 14%

Rejection sampling from the mode (RSM)

Advantage

- ✓ RMS is an exact approach.
- ✓ It provides uncorrelated samples.

Disadvantage

- ✗ It still yields small acceptance rates.
- ✗ Curse of rejection samplers: not usable in high dimensions.
- ✗ It requires two rejection steps.

Table of Contents

The SOV method from (Genz, 1992) allows sampling from truncated multivariate normals by simulating truncated univariate normals.

Consider the LQ-decomposition

$$\mathbf{\Lambda} = \mathbf{L}\mathbf{Q}^\top,$$

where

- $\mathbf{L}$: (full rank) lower triangular matrix with non-negative entries,
- $\mathbf{Q}$: orthonormal matrix.

Let $\boldsymbol{\xi} \sim \mathcal{TN}(\mathbf{0}, \mathbf{I}, \boldsymbol{l}, \boldsymbol{u})$ and $\mathbf{z} = \mathbf{Q}^\top \boldsymbol{\xi}$, then

$$\boldsymbol{l} \leq \mathbf{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u} \quad \Rightarrow \quad \boldsymbol{l} \leq \mathbf{L}\mathbf{z} \leq \boldsymbol{u}.$$

One can observe,

$$\begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{bmatrix} \leq \begin{bmatrix} L_{1,1} & 0 & 0 & \dots & 0 \\ L_{2,1} & L_{2,2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{m,1} & L_{m,2} & L_{m,3} & \cdots & L_{m,m} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix} \leq \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}.$$

Then, the importance sampling density can be written as

$$g(\mathbf{z}) = g(z_1)g(z_2|z_1)\cdots g(z_m|z_1\cdots z_{m-1}), \quad s.t. \quad \boldsymbol{l} \leq \mathbf{Lz} \leq \boldsymbol{u},$$

where

$$g(z_k|z_1,\cdots z_{k-1}) \sim \mathcal{TN}(0,1,\widetilde{l}_k(z_1,\cdots,z_{k-1}),\widetilde{u}_k(z_1,\cdots,z_{k-1})).$$

Finally, samples of $\mathbf{z}$ can be generated via importance sampling (IS).

One can observe,

$$\begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{bmatrix} \leq \begin{bmatrix} L_{1,1} & 0 & 0 & \dots & 0 \\ L_{2,1} & L_{2,2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{m,1} & L_{m,2} & L_{m,3} & \cdots & L_{m,m} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix} \leq \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}.$$

Then, the importance sampling density can be written as

$$g(\mathbf{z}) = g(z_1)g(z_2|z_1)\cdots g(z_m|z_1\cdots z_{m-1}), \quad s.t. \quad \boldsymbol{l} \leq \mathbf{L}\mathbf{z} \leq \boldsymbol{u},$$

where

$$g(z_k|z_1,\cdots z_{k-1}) \sim \mathcal{TN}(0,1,\widetilde{l}_k(z_1,\cdots,z_{k-1}),\widetilde{u}_k(z_1,\cdots,z_{k-1})).$$

Finally, samples of $\mathbf{z}$ can be generated via importance sampling (IS).

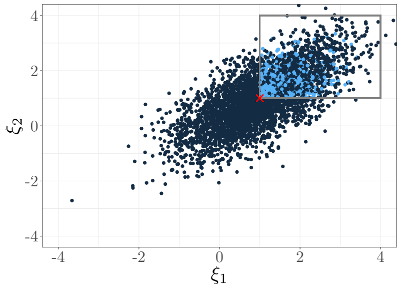
$\Rightarrow$ This approach was further investigated in (Botev, 2017) for rare events via exponential tilting (ExpT) (see, e.g., (L'Ecuyer et al., 2010)).

Separation of variable (SOV)

$$\begin{bmatrix}\xi_1\\ \xi_2\end{bmatrix} \sim \mathcal{TN}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1.0 & 0.7\\ 0.7 & 1.0\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}4\\4\end{bmatrix}\right).$$

Number of simulations: 10^3

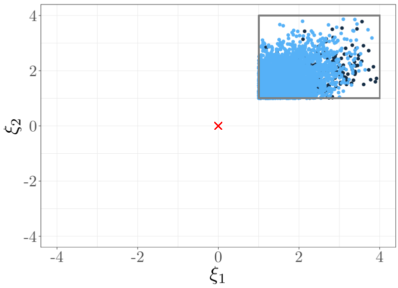
RSM



$\mu^*_{\mathrm{RSM}} = [1, 1]$

Acceptance rate: 14%

ExpT



$\mu = [0, 0]$

Acceptance rate: 94%

Table of Contents

Markov Chain Monte Carlo (MCMC)

MCMC assumes that the sample-path performs a Markov Chain.

$$\boldsymbol{\xi}_0 \longrightarrow \boldsymbol{\xi}_1 \longrightarrow \boldsymbol{\xi}_2 \longrightarrow \cdots \longrightarrow \boldsymbol{\xi}_N$$

Advantage

- ✓ Samples can be obtained with higher acceptance rate.
- ✓ There are MCMC algorithms that scale well to high dimensions.

Disadvantage

- ✗ Typically, MCMC is only an approximatation.
- ✗ Correlated samples: some of the samples have to be discarded.
- ✗ Burn-in the Markov chain.

Markov Chain Monte Carlo (MCMC)

MCMC assumes that the sample-path performs a Markov Chain.

$$\boldsymbol{\xi}_0 \longrightarrow \boldsymbol{\xi}_1 \longrightarrow \boldsymbol{\xi}_2 \longrightarrow \cdots \longrightarrow \boldsymbol{\xi}_N$$

Advantage

- ✓ Samples can be obtained with higher acceptance rate.
- ✓ There are MCMC algorithms that scale well to high dimensions.

Disadvantage

- ✗ Typically, MCMC is only an approximatation.
- ✗ Correlated samples: some of the samples have to be discarded.
- ✗ Burn-in the Markov chain.

Some efficient MCMC methods for truncated multivariate normals:

- Gibbs sampling (Taylor and Benjamini, 2017).
- Exact Hamiltonian Monte Carlo (Pakman and Paninski, 2014).

Comparison between proposed MCMC techniques w.r.t. the RSM approach

	RSM	ExpT	Gibbs	HMC
Exact method	✓	✓	✗	✓
Non parametric	✓	✓	✓	✗(✓)
Acceptance rate	✗	✗-✓	**100%**	✓
Speed	✗	✗-✓	✓	✓
Uncorrelated samples	✓	✓	✗	✗
R Package	`constrKriging`	`TruncatedNormal`	`tmvtnorm`	`tmg`

RSM: Rejection Sampling from the Mode (Maatouk and Bay, 2016)
ExpT: Exponential Tilting (Botev, 2017)
Gibbs sampling (Taylor and Benjamini, 2017)
HMC: Hamiltonian Monte Carlo (Pakman and Paninski, 2014)

Table of Contents

Numerical results

- We simulate constrained samples under different conditions according to (López-Lopera et al., 2017).

Numerical results

- We simulate constrained samples under different conditions according to (López-Lopera et al., 2017).
- For MCMC methods, we use the mode as initial state of the chains.
- We "burn" the first 100 realization.
- We fix the MCMC hyperparameters in order to obtain less correlated samples (e.g. thinning for Gibb sampling).

Numerical results

- We simulate constrained samples under different conditions according to (López-Lopera et al., 2017).
- For MCMC methods, we use the mode as initial state of the chains.
- We "burn" the first 100 realization.
- We fix the MCMC hyperparameters in order to obtain less correlated samples (e.g. thinning for Gibb sampling).
- We assess algorithms via Effective Sample Size (ESS) criterion:

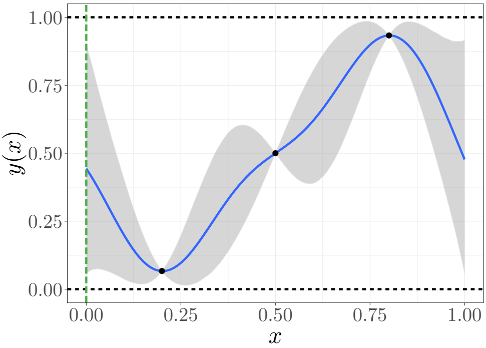
$$\mathrm{ESS} = \frac{N}{1 + 2\sum_{k=1}^{N} \widehat{\rho}_k}$$

  with N the sample size and $\widehat{\rho}_k$ a positive and convex estimator of the sample autocorrelation ρ_k with lag k (Geyer, 1992).
- Time normalised (TN)-ESS:

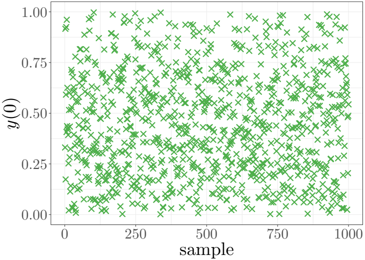
$$\text{TN-ESS} = \frac{\mathrm{ESS}}{\mathrm{Time}\ [s]}.$$

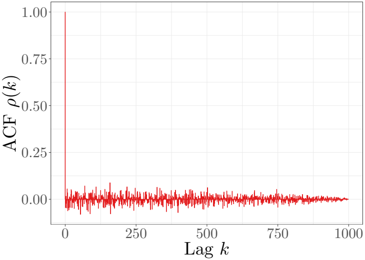
Toy example 1: boundedness conditions

Example under boundedness constraints: **RSM**



$m = 100,\ q = 100,\ N = 10^3$





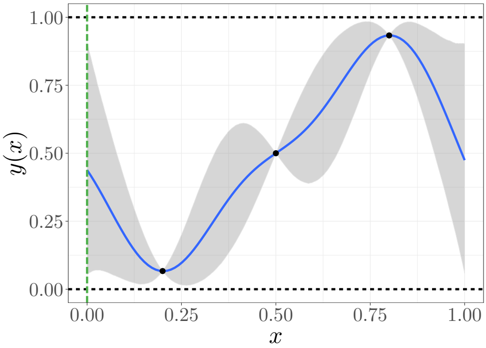
Method	CPU Time [s]	ESS [$\times 10^4 s^{-1}$] ($q_{10\%}, q_{50\%}, q_{90\%}$)	TN-ESS ($q_{10\%}$)
RSM	1207.4	(0.94, 0.99, 1.00)	0.001

MINES Saint-Étienne

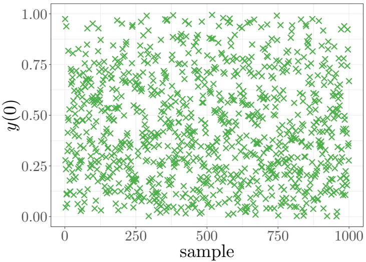
INSTITUT DE MATHÉMATIQUES DE TOULOUSE

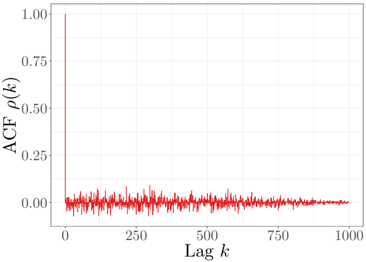
Toy example 1: boundedness conditions

Example under boundedness constraints: **ExpT**



$$m = 100, \; q = 100, \; N = 10^3$$

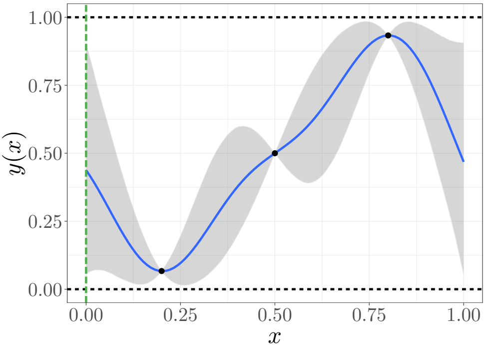




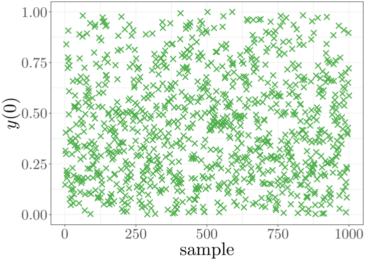
Method	CPU Time $[s]$	ESS $[\times 10^4 s^{-1}]$ $(q_{10\%}, q_{50\%}, q_{90\%})$	TN-ESS $(q_{10\%})$
RSM	1207.4	(0.94, 0.99, 1.00)	0.001
ExpT	3.3	(0.94, 0.99, 1.00)	0.285

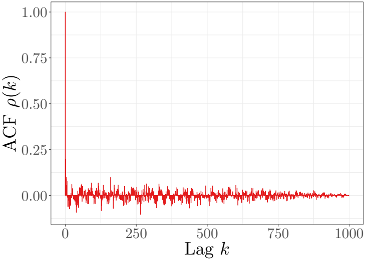
Toy example 1: boundedness conditions

Example under boundedness constraints: **Gibbs**



$m = 100, \; q = 100, \; N = 10^3$





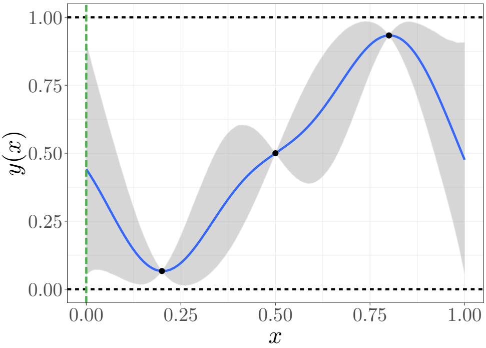
Method	CPU Time $[s]$	ESS $[\times 10^4 s^{-1}]$ $(q_{10\%}, q_{50\%}, q_{90\%})$	TN-ESS $(q_{10\%})$
RSM	1207.4	(0.94, 0.99, 1.00)	0.001
ExpT	3.3	(0.94, 0.99, 1.00)	0.285
Gibbs	9.6	(0.57, 0.75, 0.81)	0.059

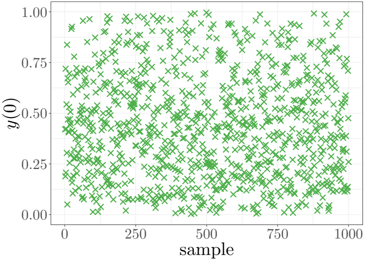
MINES Saint-Étienne

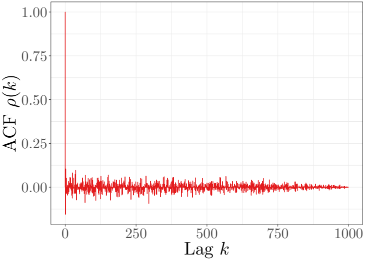
INSTITUT DE MATHÉMATIQUES DE TOULOUSE

Example under boundedness constraints: **HMC**

$m = 100,\ q = 100,\ N = 10^3$



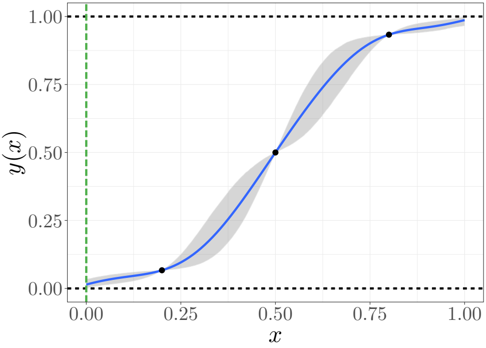


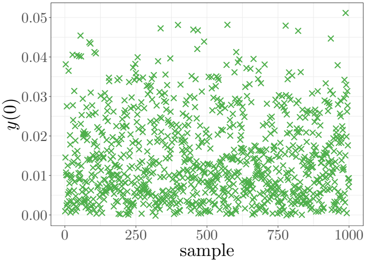


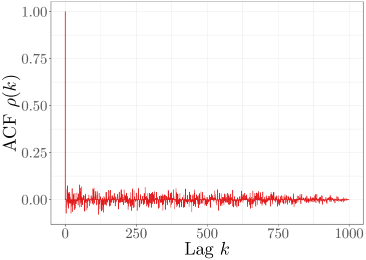
Method	CPU Time $[s]$	ESS $[\times 10^4 s^{-1}]$ $(q_{10\%}, q_{50\%}, q_{90\%})$	TN-ESS $(q_{10\%})$
RSM	1207.4	(0.94, 0.99, 1.00)	0.001
ExpT	3.3	(0.94, 0.99, 1.00)	0.285
Gibbs	9.6	(0.57, 0.75, 0.81)	0.059
HMC	**2.6**	(0.88, 0.91, 0.94)	**0.338**

Example under boundedness & monotonicity constraints: **ExpT**

$$m = 100,\ q = 200,\ N = 10^3$$



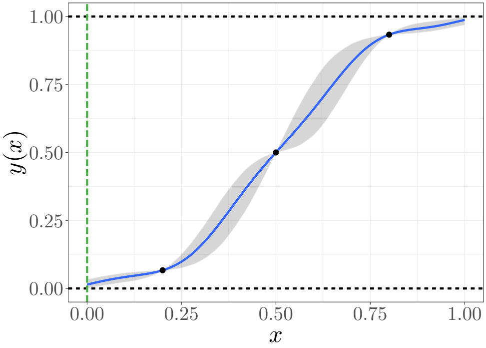


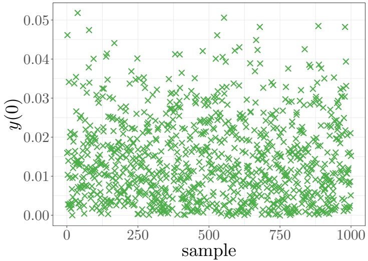


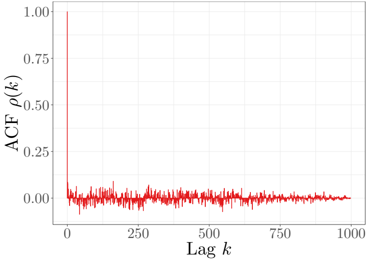
Method	CPU Time $[s]$	ESS $[\times 10^4 s^{-1}]$ $(q_{10\%}, q_{50\%}, q_{90\%})$	TN-ESS $(q_{10\%})$
RSM	-	-	-
ExpT	73.7	(0.87, 1.00, 1.00)	0.012

Example under boundedness & monotonicity constraints: **Gibbs**

$$m = 100,\ q = 200,\ N = 10^3$$





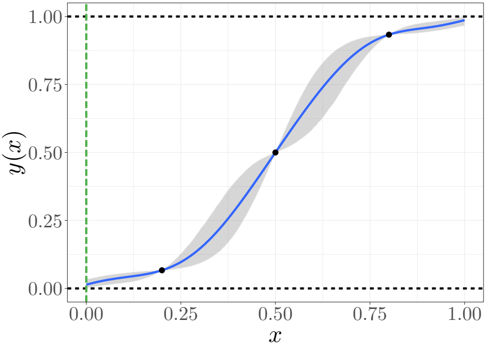


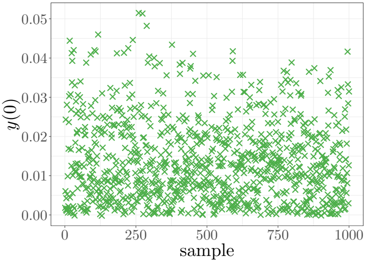
Method	CPU Time [s]	ESS [$\times 10^4 s^{-1}$] ($q_{10\%}, q_{50\%}, q_{90\%}$)	TN-ESS ($q_{10\%}$)
RSM	-	-	-
ExpT	73.7	(0.87, 1.00, 1.00)	0.012
Gibbs	29.0	(0.18, 0.40, 0.96)	0.006

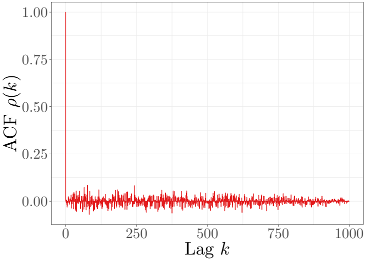
Toy example 2: boundedness & monotonicity conditions

Example under boundedness & monotonicity constraints: **HMC**

$m = 100,\ q = 200,\ N = 10^3$





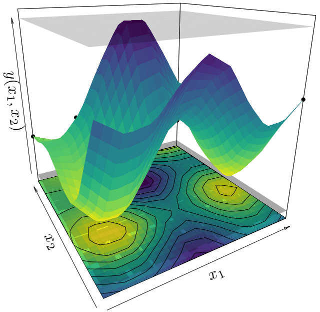


Method	CPU Time $[s]$	ESS $[\times 10^4 s^{-1}]$ $(q_{10\%}, q_{50\%}, q_{90\%})$	TN-ESS $(q_{10\%})$
RSM	-	-	-
ExpT	73.7	(0.87, 1.00, 1.00)	0.012
Gibbs	29.0	(0.18, 0.40, 0.96)	0.006
HMC	**12.6**	(0.77, 0.84, 0.97)	**0.061**

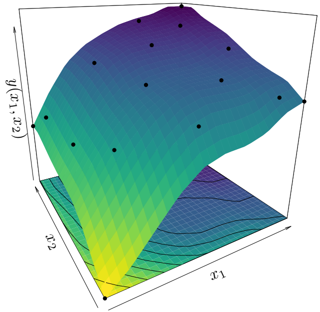
MINES Saint-Étienne

Toy example 3: boundedness conditions on 2D

2D Examples under boundedness constr.



2D Examples under monotonicity constr.



Method	2D boundedness example			2D monotonicity example		
	CPU Time $[s]$	ESS $[\times 10^4 s^{-1}]$ $(q_{10\%}, q_{50\%}, q_{90\%})$	TN-ESS $(q_{10\%})$	CPU Time $[s]$	ESS $[\times 10^4 s^{-1}]$ $(q_{10\%}, q_{50\%}, q_{90\%})$	TN-ESS $(q_{10\%})$
RSM	-	-	-	-	-	-
ExpT	0.9	(0.90, 1.00, 1.00)	1.009	1488.3	(0.93, 1.00, 1.00)	0.001
Gibbs	9.7	(0.85, 0.96, 1.00)	0.088	-	-	-
HMC	**0.6**	(0.83, 0.93, 1.00)	**1.493**	**8.6**	(0.80, 0.90, 1.00)	**0.093**

$$m_1 = m_2 = 10, N = 10^3$$

Table of Contents

Conclusions

- We further investigated the approach from (Maatouk and Bay, 2017) under linear inequality constraints.
- We suggested Hamiltonian MC to approximate the posterior of $\boldsymbol{\xi}$.
- We implemented the R package: `lineqGPR` (available on July).

✦ A.F. López-Lopera, F. Bachoc, N. Durrande, and O. Roustant (2017). *Finite-dimensional Gaussian approximation with linear inequality constraints.* ArXiv preprint

✦ A.F. López-Lopera. *LineqGPR: Gaussian process regression models with linear inequality constraints*, 2018.

Conclusions

- We further investigated the approach from (Maatouk and Bay, 2017) under linear inequality constraints.
- We suggested Hamiltonian MC to approximate the posterior of $\boldsymbol{\xi}$.
- We implemented the R package: `lineqGPR` (available on July).

✦ A.F. López-Lopera, F. Bachoc, N. Durrande, and O. Roustant (2017). *Finite-dimensional Gaussian approximation with linear inequality constraints.* ArXiv preprint

✦ A.F. López-Lopera. *LineqGPR: Gaussian process regression models with linear inequality constraints*, 2018.

Future work

- Zig-Zag/Bouncy samplers for truncated multivariate normals?

Table of Contents

References I

Botev, Z. I. (2017). The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(1):125–148.

Genz, A. (1992). Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1:141–150.

Geyer, C. J. (1992). Practical Markov Chain Monte Carlo. *Statistical Science*, 7(4):473–483.

L'Ecuyer, P., Blanchet, J. H., Tuffin, B., and Glynn, P. W. (2010). Asymptotic robustness of estimators in rare-event simulation. *ACM Trans. Model. Comput. Simul.*, 20(1):6:1–6:41.

López-Lopera, A. F., Bachoc, F., Durrande, N., and Roustant, O. (2017). Finite-dimensional Gaussian approximation with linear inequality constraints. *ArXiv e-prints*.

Maatouk, H. and Bay, X. (2016). *A New Rejection Sampling Method for Truncated Multivariate Gaussian Random Variables Restricted to Convex Sets*, pages 521–530. Springer International Publishing, Cham.

Maatouk, H. and Bay, X. (2017). Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences*, 49(5):557–582.

Pakman, A. and Paninski, L. (2014). Exact Hamiltonian Monte Carlo for truncated multivariate Gaussians. *Journal of Computational and Graphical Statistics*, 23(2):518–542.

Taylor, J. and Benjamini, Y. (2017). RestrictedMVN: multivariate normal restricted by affine constraints. `https://cran.r-project.org/web/packages/restrictedMVN/index.html`. [Online; 02-Feb-2017].