

Algorithme itératif pour l'estimation d'une probabilité de défaillance

Journées des Doctorants, Orléans 2018

- Lucie Bernard** - Institut Denis Poisson - LPSM - STMicroelectronics
Arnaud Guyader - LPSM - Sorbonne Université, Paris
Philippe Leduc - STMicroelectronics, Tours
Florent Malrieu - Institut Denis Poisson - Université Tours

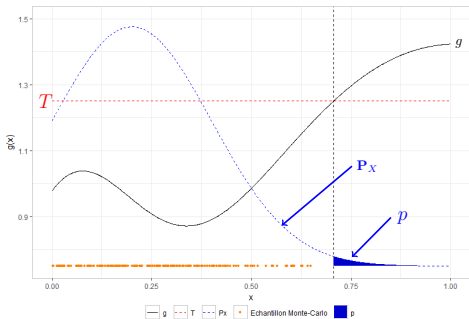


- Soit $A = [0, 1]^d$ et $g : A \rightarrow \mathbb{R}$ une fonction **boîte noire coûteuse à évaluer**.
- À partir d'un **petit nombre d'évaluations** de g , on veut estimer la **probabilité de défaillance** p définie par :

$$p = \mathbb{P}(g(\mathbf{X}) > T),$$

où :

- \mathbf{X} est une variable aléatoire dont la loi \mathbf{P}_X est connue à une constante de normalisation près,
- T est un seuil tel que p est **très petite**.

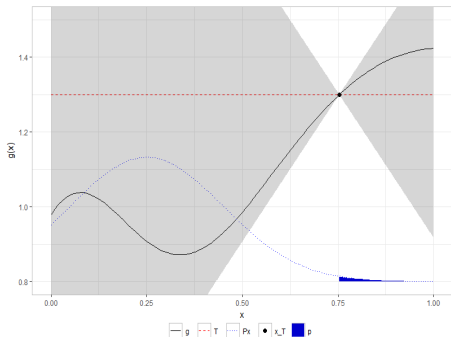
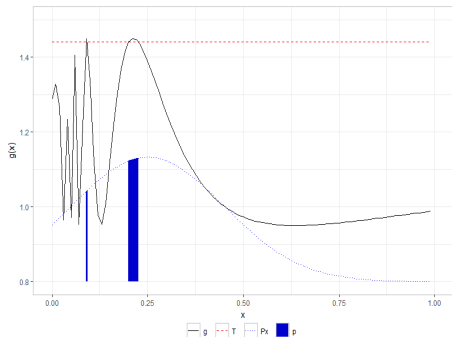


Hypothèse

L'algorithme s'applique sous l'hypothèse que :

(\mathcal{H}_1) : g est **lipschitzienne** de constante M connue :

$$|g(x) - g(x')| \leq M \|x - x'\|_{\infty}, \quad \forall (x, x') \in A^2.$$

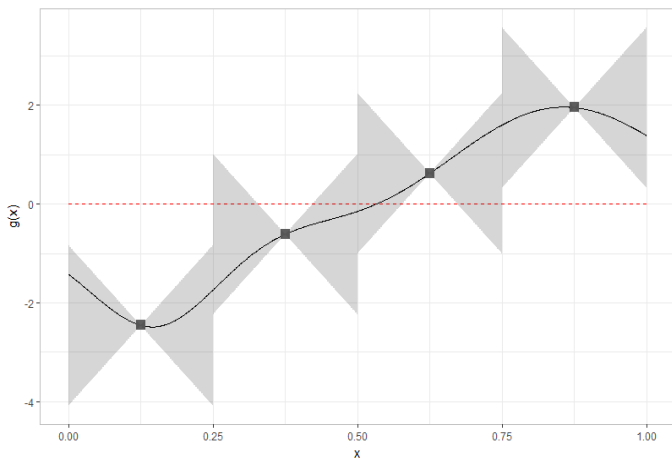


Hypothèse

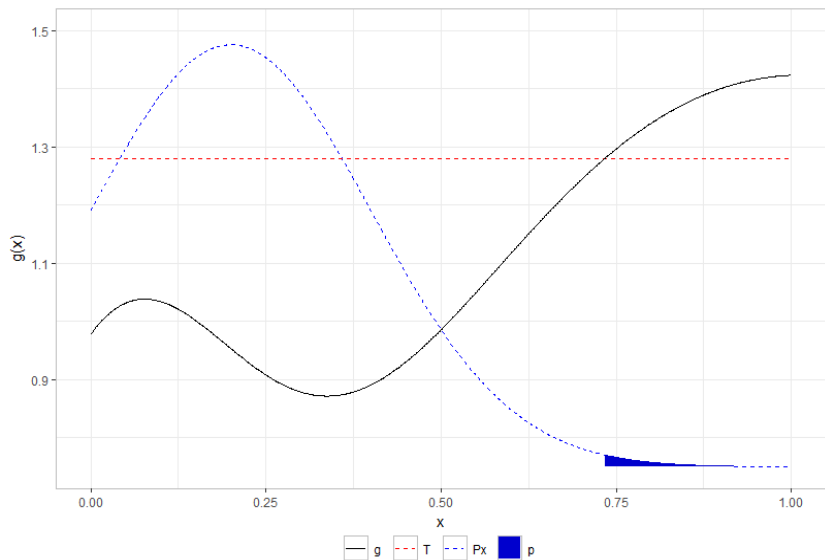
L'algorithme s'applique sous l'hypothèse que :

(\mathcal{H}_1) : g est **lipschitzienne** de constante M connue :

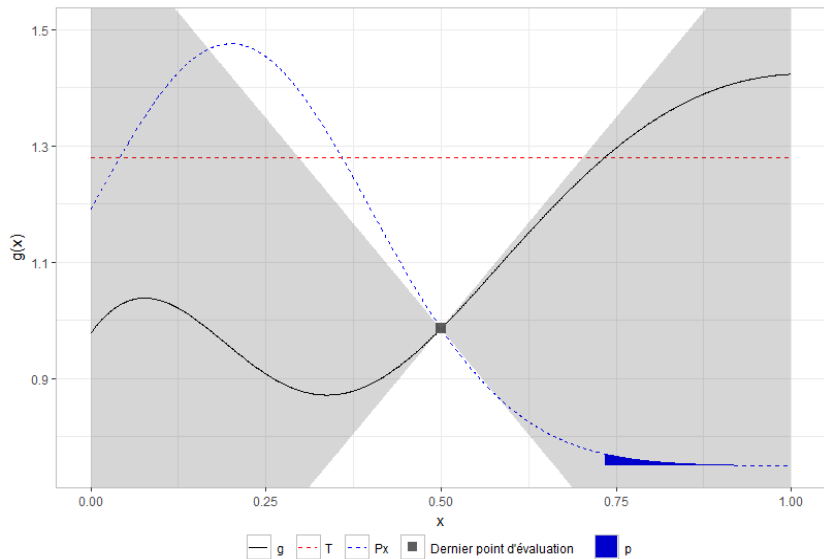
$$|g(\mathbf{x}) - g(\mathbf{x}')| \leq M \|\mathbf{x} - \mathbf{x}'\|_{\infty}, \quad \forall (\mathbf{x}, \mathbf{x}') \in A^2.$$



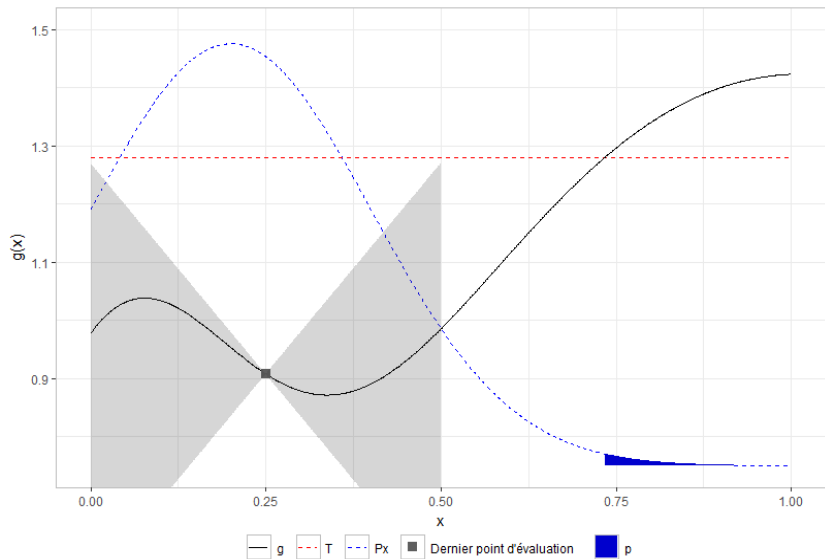
Exemple en dimension 1



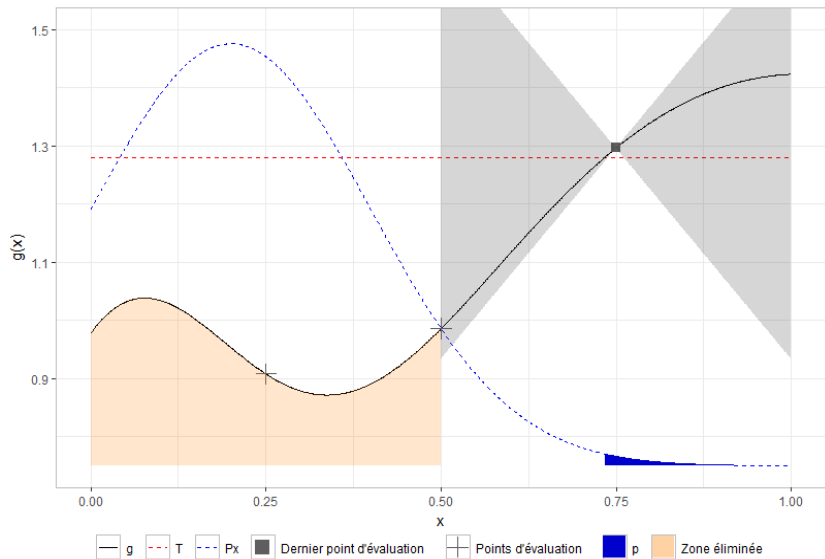
Exemple en dimension 1 : Étape 1



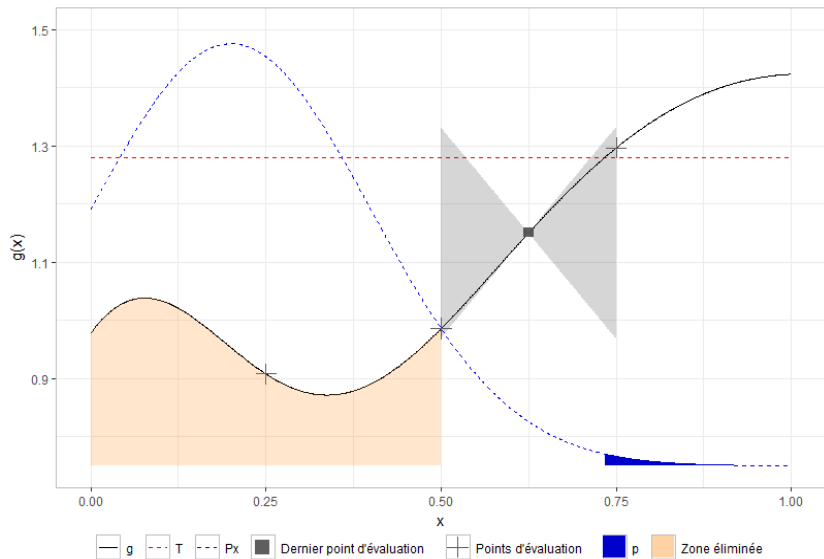
Exemple en dimension 1 : Étape 2



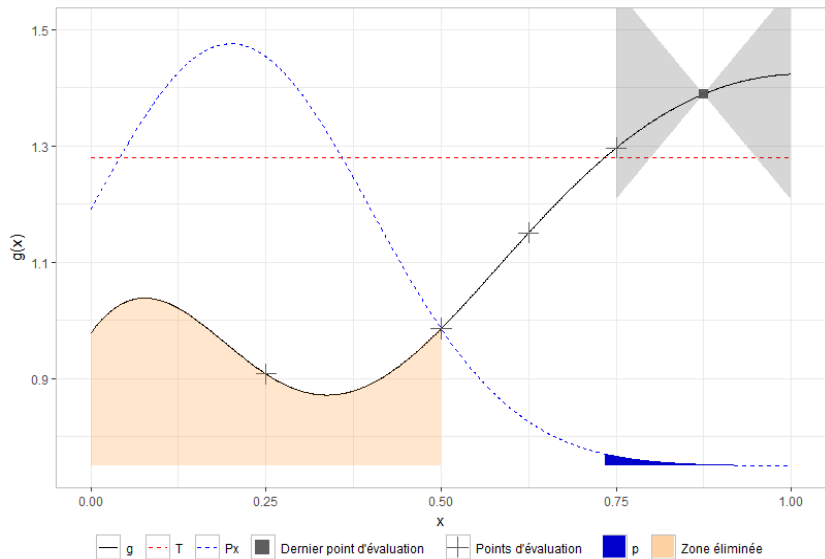
Exemple en dimension 1 : Étape 2



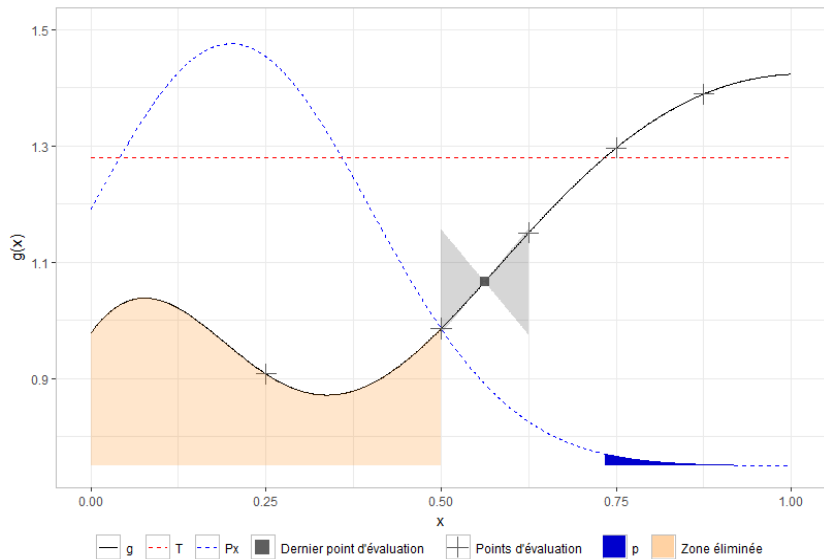
Exemple en dimension 1 : Étape 3



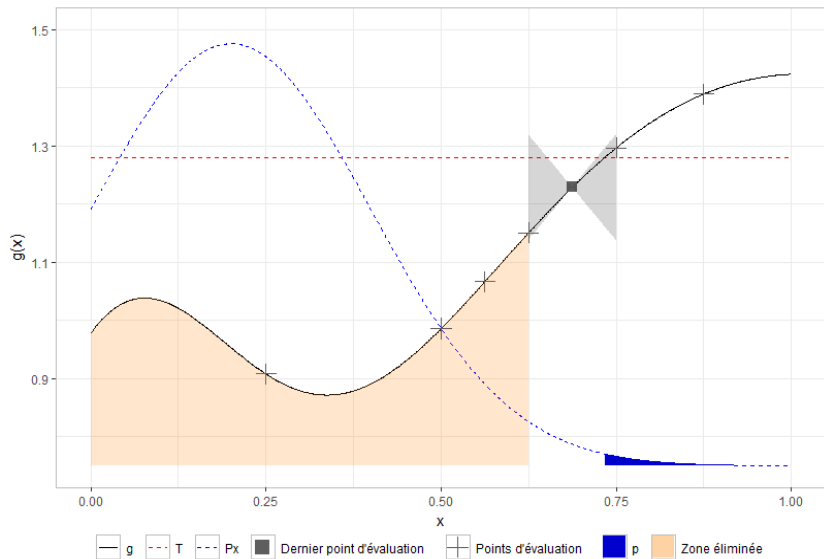
Exemple en dimension 1 : Étape 3



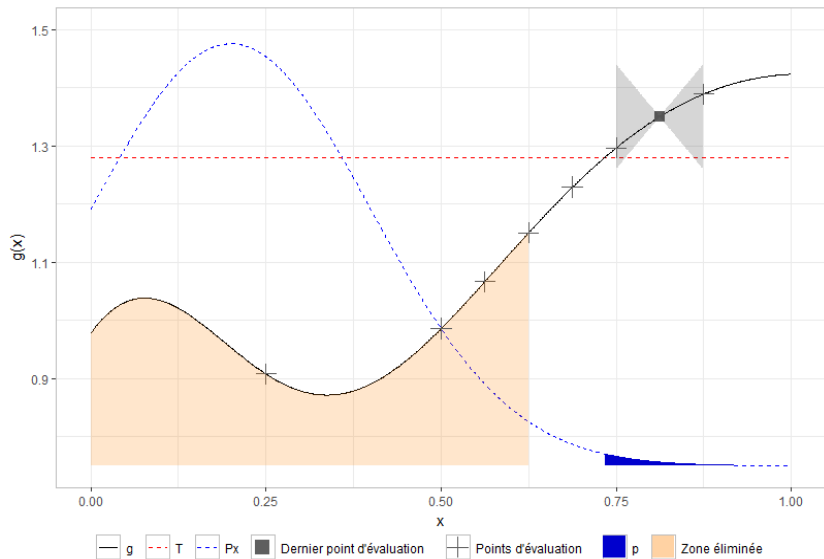
Exemple en dimension 1 : Étape 4



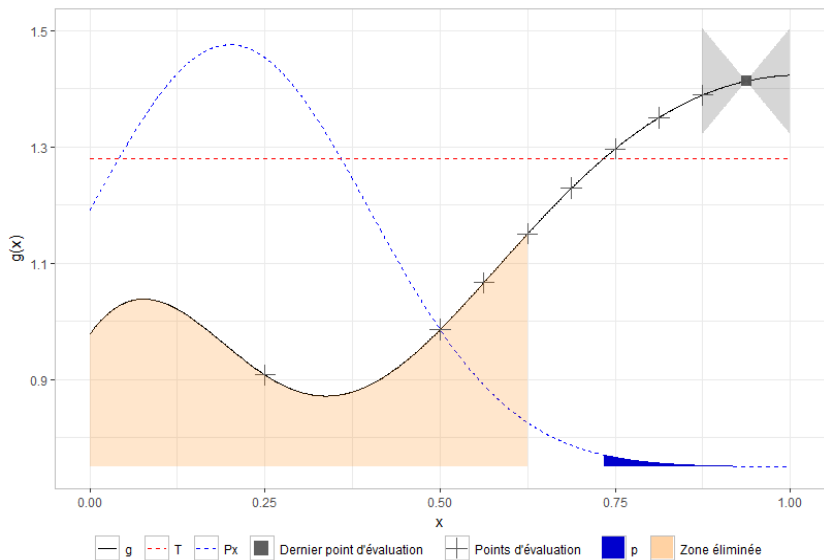
Exemple en dimension 1 : Étape 4



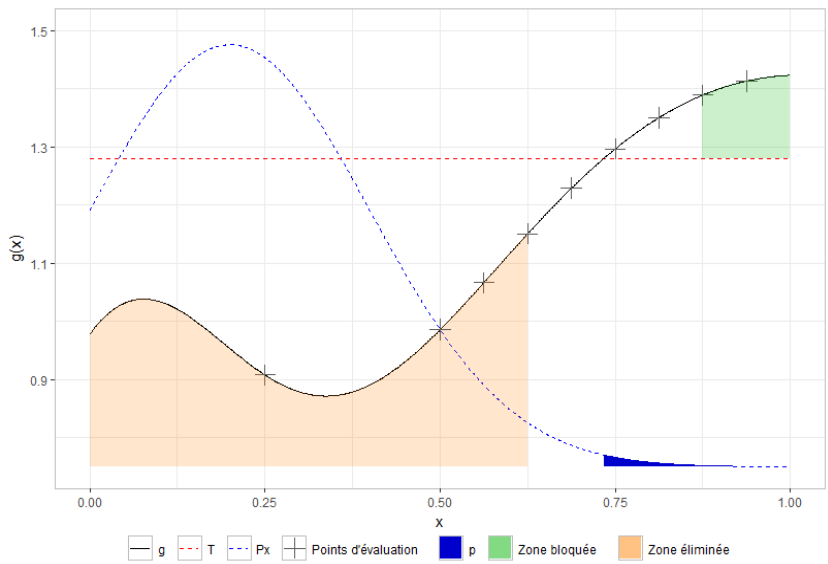
Exemple en dimension 1 : Étape 4



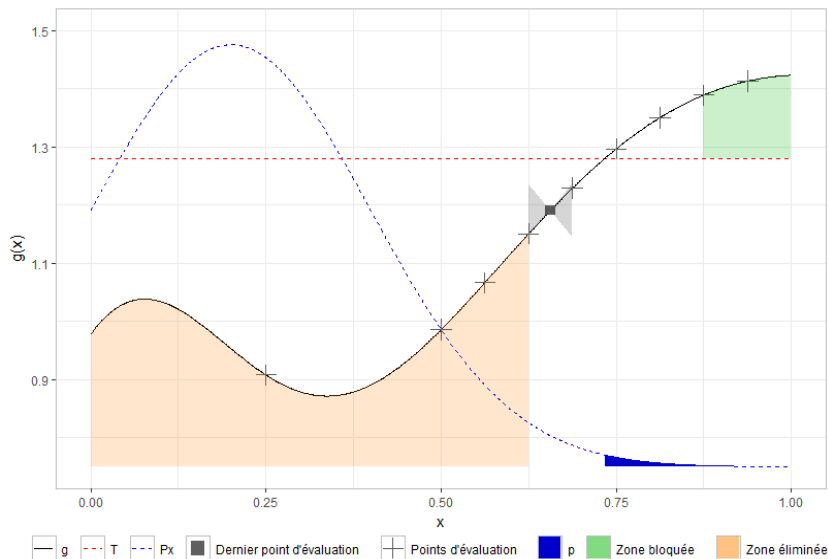
Exemple en dimension 1 : Étape 4



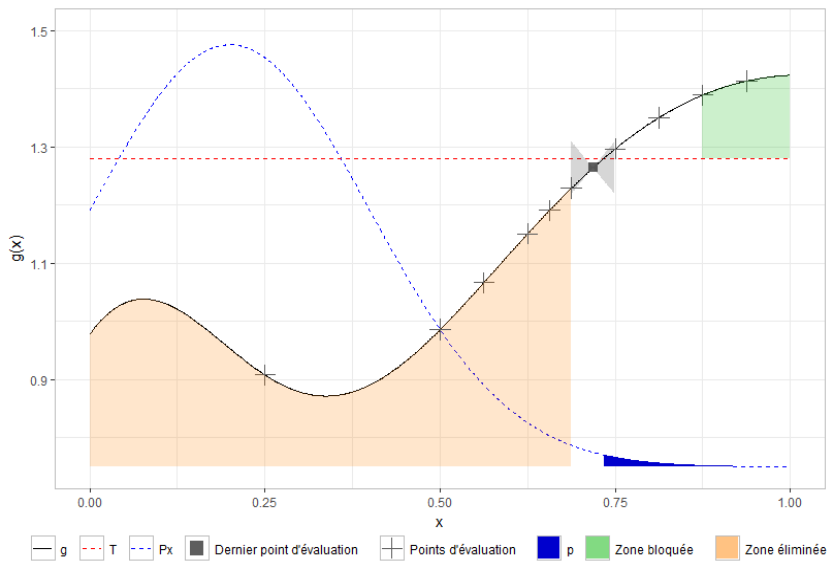
Exemple en dimension 1 : Étape 4



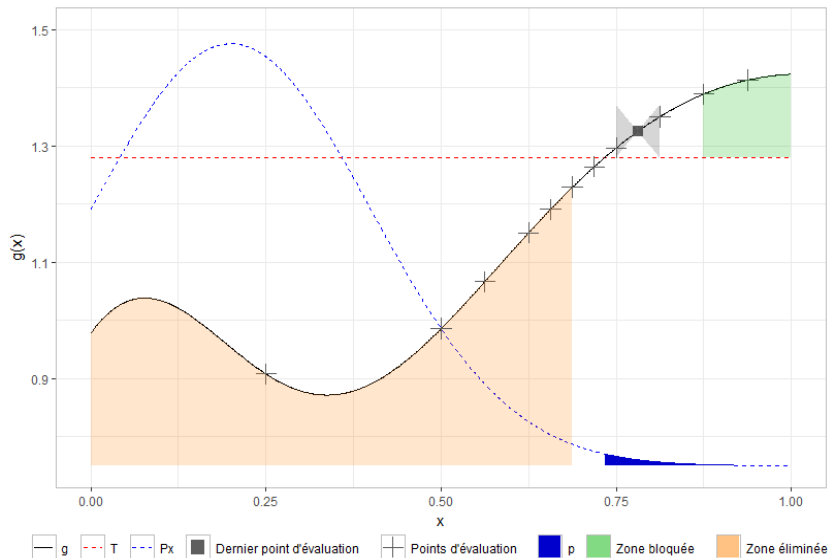
Exemple en dimension 1 : Étape 5



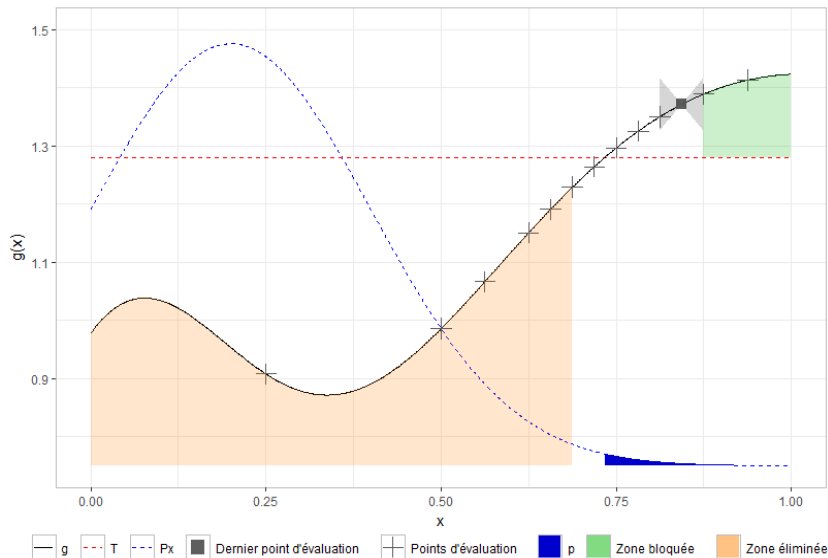
Exemple en dimension 1 : Étape 5



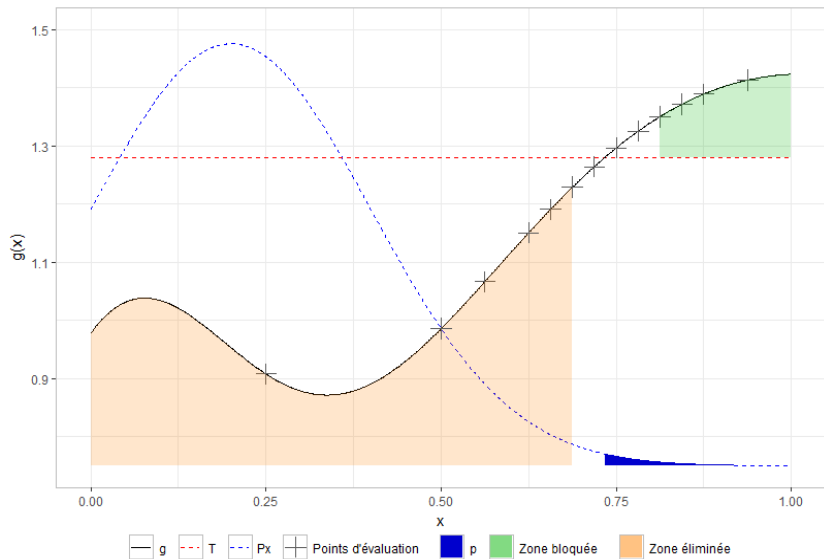
Exemple en dimension 1 : Étape 5



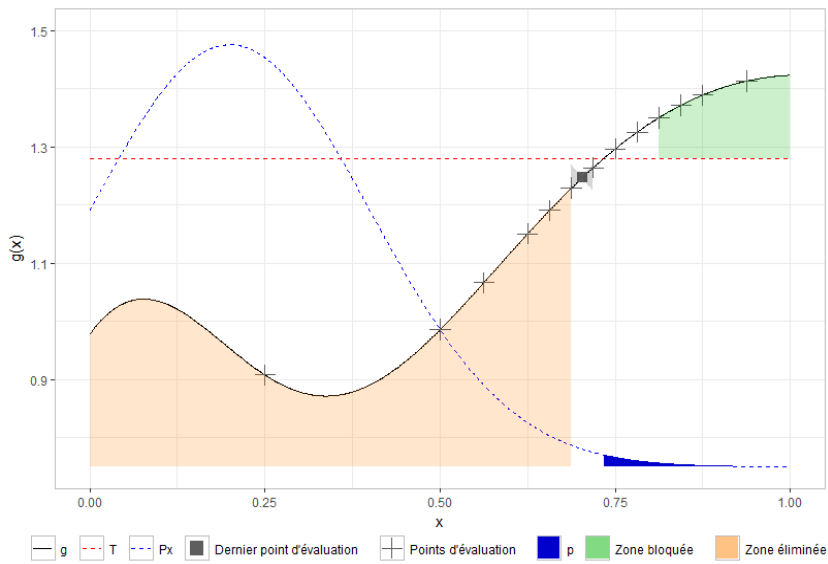
Exemple en dimension 1 : Étape 5



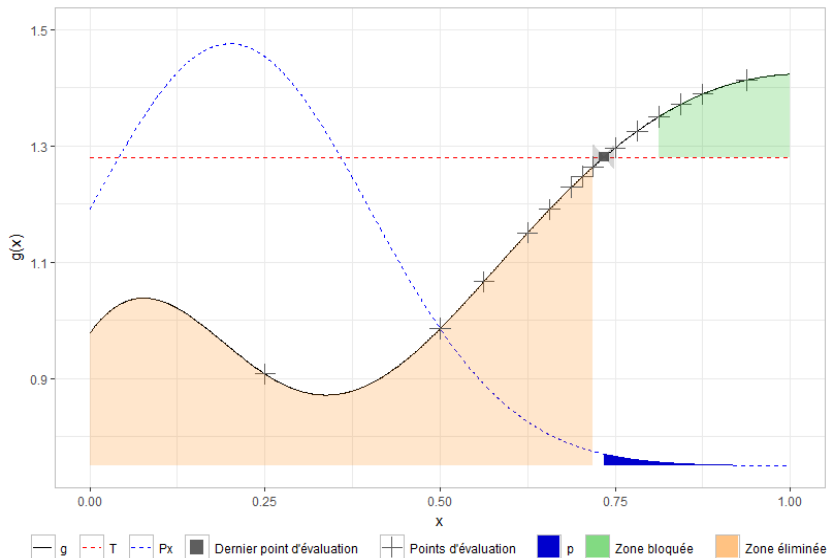
Exemple en dimension 1 : Étape 5



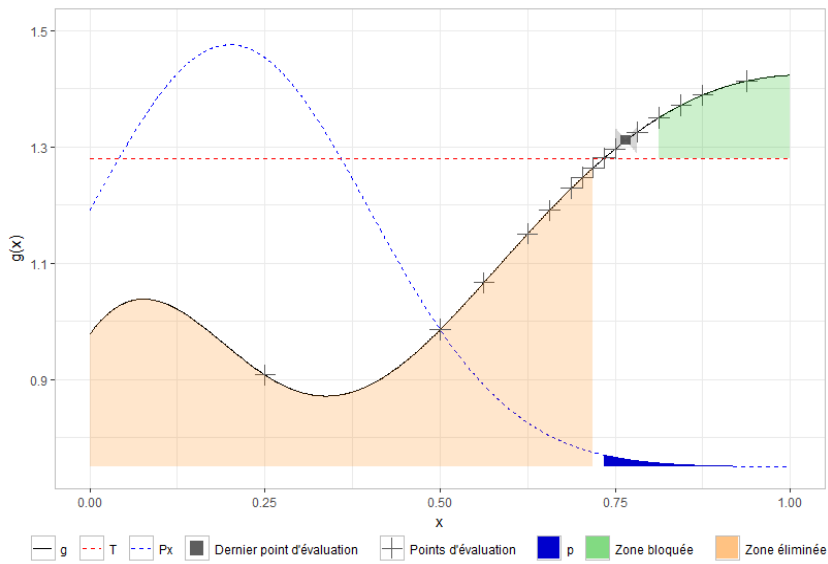
Exemple en dimension 1 : Étape 6



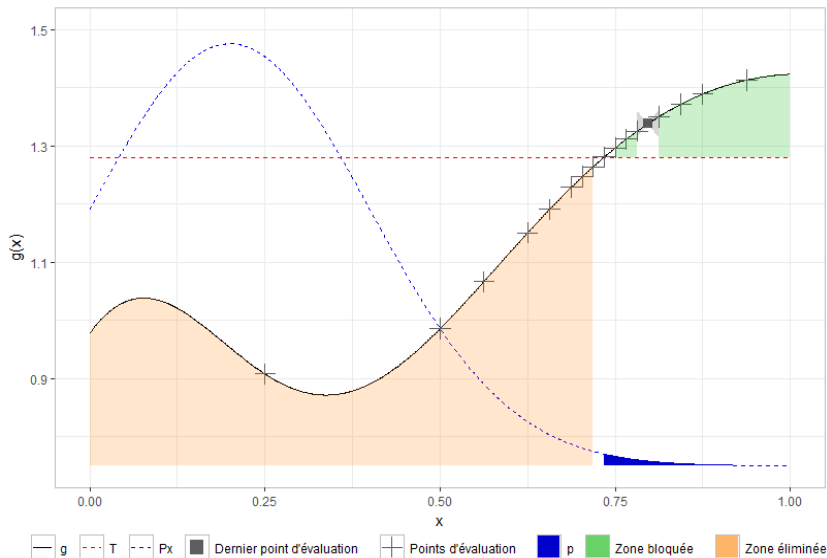
Exemple en dimension 1 : Étape 6



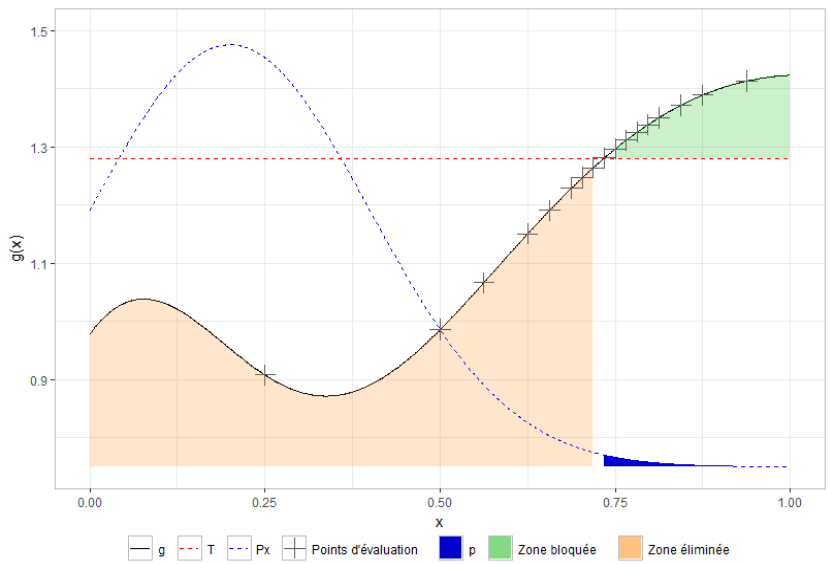
Exemple en dimension 1 : Étape 6



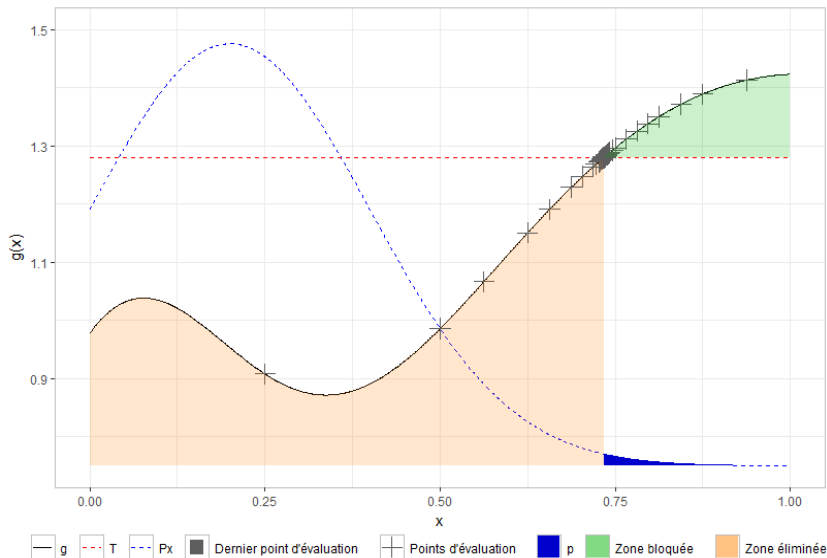
Exemple en dimension 1 : Étape 6



Exemple en dimension 1 : Étape 6

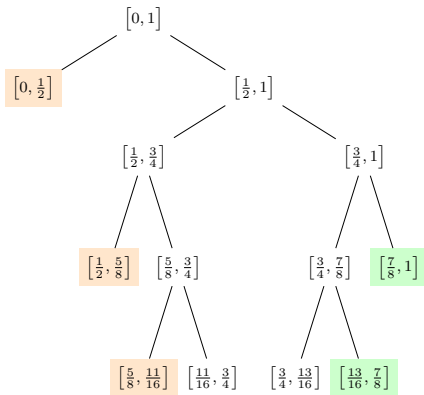


Exemple en dimension 1 : Fin



L'algorithme procède à une **partition dyadique** de $A = [0, 1]^d$ et utilise l'hypothèse (\mathcal{H}_1) pour **définir** :

- Une suite décroissante $(\mathcal{A}_k)_{k \geq 0}$ de sous-domaines de A où g **peut dépasser** T ,
- Une suite **croissante** $(\mathcal{S}_k)_{k \geq 0}$ de sous-domaines de A où g est **toujours au-dessus** de T ,
- Une suite **croissante** $(\mathcal{O}_k)_{k \geq 0}$ de sous-domaines de A où g est **toujours en-dessous** de T .

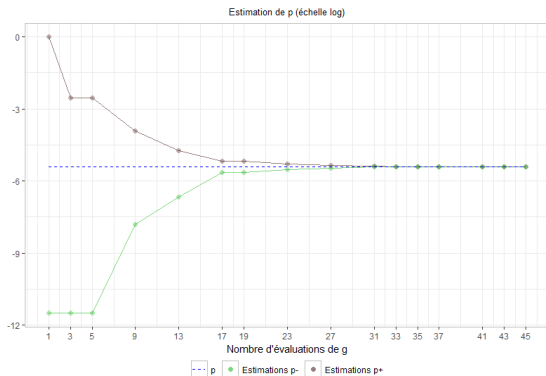


À chaque étape $k = 0, 1, \dots$, des approximations p_k^- et p_k^+ de p sont données par :

$$p_k^- = \mathbf{P}_X(\mathcal{S}_k) \text{ et } p_k^+ = \mathbf{P}_X(\mathcal{S}_k) + \mathbf{P}_X(\mathcal{A}_k),$$

et vérifient :

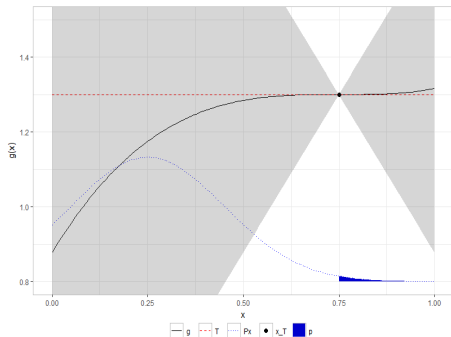
$$p_k^- \leq p \leq p_k^+.$$



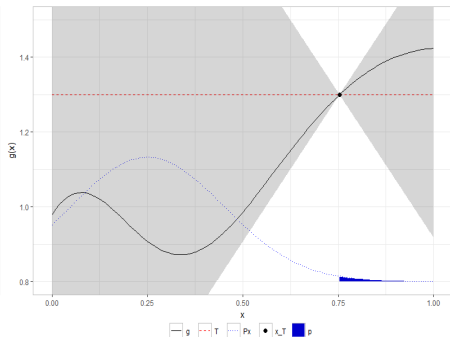
Analyse et propriétés de l'algorithme

Soit C une constante qui vérifie :

$$C \geq \left(\frac{M}{\|\nabla g(\mathbf{x}_T)\|_\infty} \right)^d \geq 1.$$



C grand.



C petit.

▷ L'algorithme est d'autant plus performant que C est proche de 1.

Proposition

1. Soit n_k le nombre total d'évaluations de la fonction g à la fin de l'étape $k \geq 1$. Alors n_k **croît linéairement** avec k :

$$n_k \leq 1 + C4^d(k - 1).$$

2. Soit $m = \sup_{\mathbf{x} \in A} f_{\mathbf{X}}(\mathbf{x})$, où $f_{\mathbf{X}}$ est la densité de \mathbf{X} . Pour tout $k \geq 1$, l'approximation p_k^+ de p vérifie :

$$p \leq p_k^+ \leq p + mC2^{d(2-k)}.$$

Corollaire

Soit $n \geq 2$ le **nombre maximal** d'appels à la fonction g . L'**erreur d'approximation**

$$E_n = p_n^+ - p,$$

vérifie :

$$E_n \leq mC4^d 2^{\frac{-nd}{C4^d}}.$$

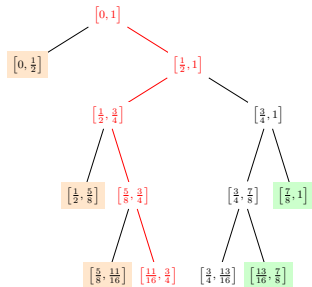
- À chaque étape $k = 1, 2, \dots$, les approximations p_k^- et p_k^+ sont **estimées** via une méthode **multi-niveaux** (*Multilevel Splitting* [1], [2]). Par exemple, on a :

$$p_k^+ = \mathbf{P}_X(\mathcal{A}_k) = \sum_{Q_k \in \mathcal{A}_k} \mathbf{P}_X(Q_k),$$

où Q_k est un cube dyadique de côté $\frac{1}{2^k}$.

- On **décompose** $\mathbf{P}_X(Q_k)$ en produit de probabilités conditionnelles :

$$\mathbf{P}_X(Q_k) = \mathbf{P}_X(Q_k | Q_{k-1})\mathbf{P}_X(Q_{k-1} | Q_{k-2}) \dots \mathbf{P}_X(Q_1) = p_k p_{k-1} \dots p_1.$$



Exemple :

$$\begin{aligned} \mathbf{P}_X\left(\left[\frac{11}{16}, \frac{3}{4}\right]\right) &= \mathbf{P}_X\left(\left[\frac{11}{16}, \frac{3}{4}\right] \mid \left[\frac{5}{8}, \frac{3}{4}\right]\right) \\ &\times \mathbf{P}_X\left(\left[\frac{5}{8}, \frac{3}{4}\right] \mid \left[\frac{1}{2}, \frac{3}{4}\right]\right) \\ &\times \mathbf{P}_X\left(\left[\frac{1}{2}, \frac{3}{4}\right] \mid \left[\frac{1}{2}, 1\right]\right) \\ &\times \mathbf{P}_X\left(\left[\frac{1}{2}, 1\right]\right). \end{aligned}$$

- Pour tout $j = 1, \dots, k$, les probabilités conditionnelles $p_j = \mathbf{P}_{\mathbf{X}}(Q_j \mid Q_{j-1})$ sont estimées par **Monte-Carlo naïf** :

$$\hat{p}_j = \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{\mathbf{X}_i \in Q_j},$$

où $\mathbf{X}_1, \dots, \mathbf{X}_N$ est un échantillon distribué suivant la **loi conditionnelle** $\mathcal{L}(\mathbf{X} \mid \mathbf{X} \in Q_{j-1})$. Il est généré à l'aide d'un **algorithme de Metropolis-Hasting**.

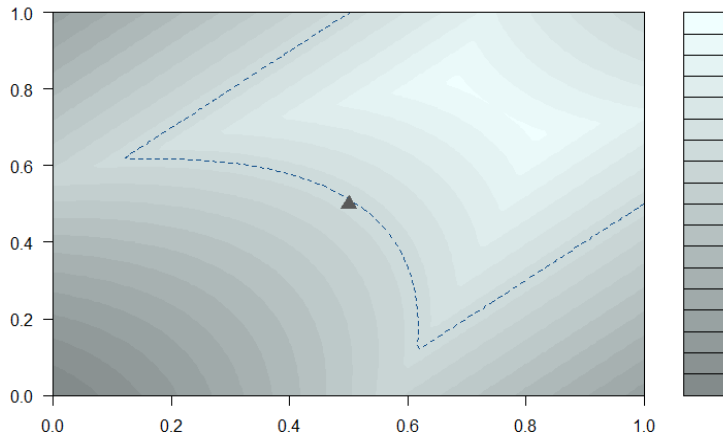
- L'estimateur de $\mathbf{P}_{\mathbf{X}}(Q_k)$ est :

$$\hat{\mathbf{P}}_{\mathbf{X}}(Q_k) = \hat{p}_k \hat{p}_{k-1} \dots \hat{p}_1,$$

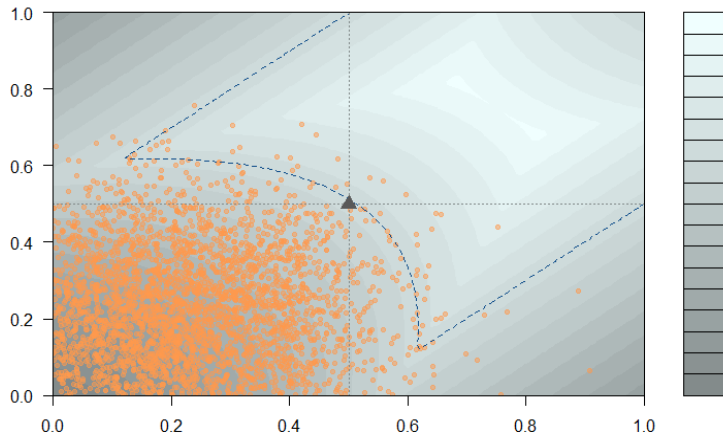
et vérifie :

$$\sqrt{N} \left(\frac{\hat{\mathbf{P}}_{\mathbf{X}}(Q_k) - \mathbf{P}_{\mathbf{X}}(Q_k)}{\mathbf{P}_{\mathbf{X}}(Q_k)} \right) \xrightarrow[N \rightarrow \infty]{\mathcal{L}} \mathcal{N} \left(0, \sum_{j=1}^k \frac{1 - p_j}{p_j} \right).$$

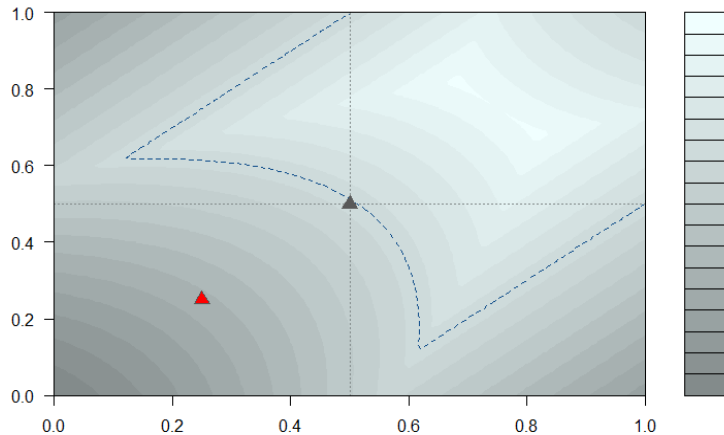
Exemple en dimension 2 : Étape 1



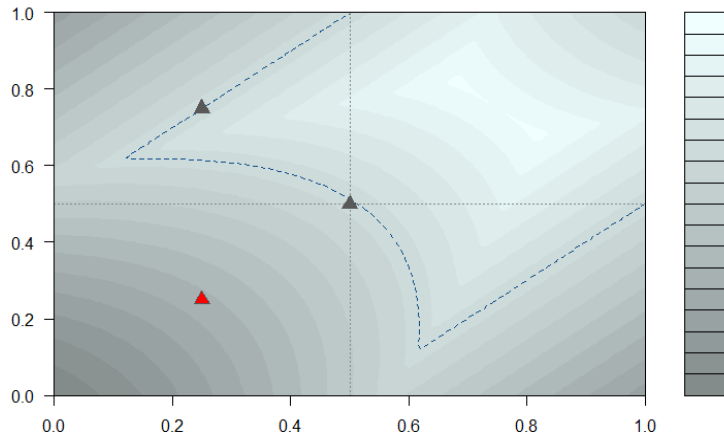
Exemple en dimension 2 : Étape 1



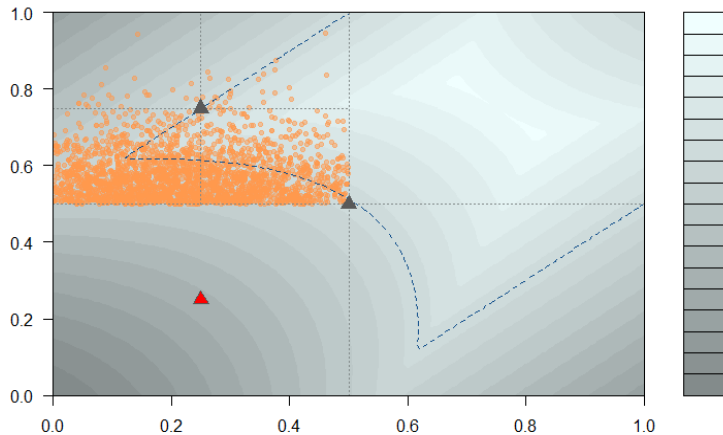
Exemple en dimension 2 : Étape 2



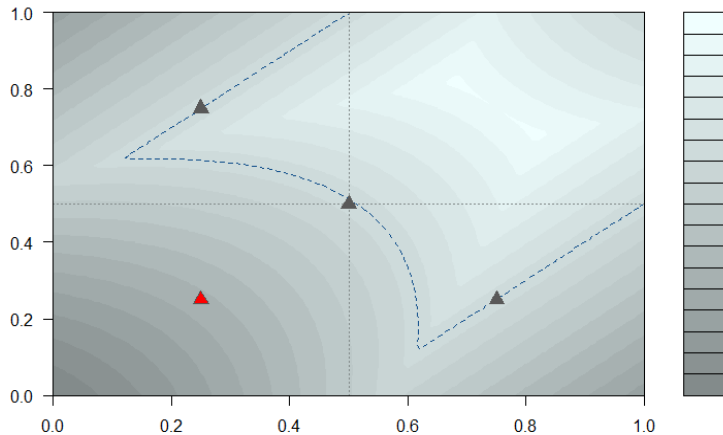
Exemple en dimension 2 : Étape 2



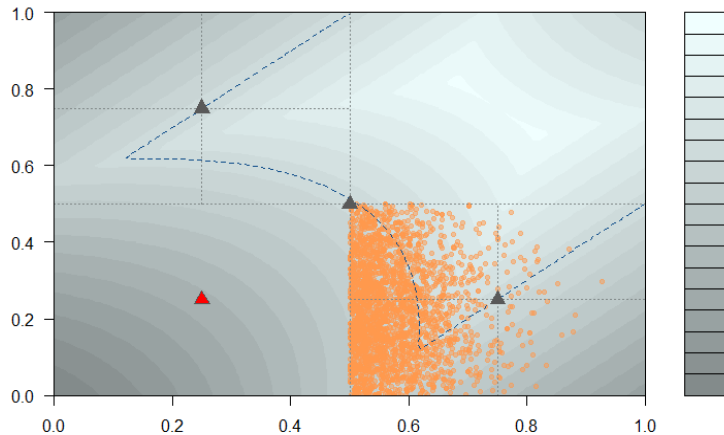
Exemple en dimension 2 : Étape 2



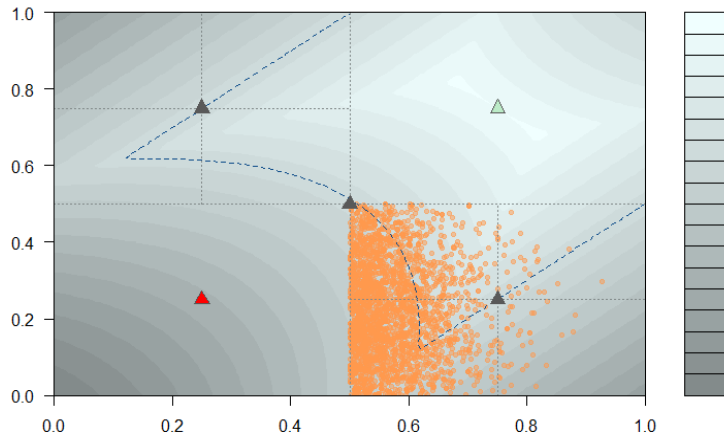
Exemple en dimension 2 : Étape 2



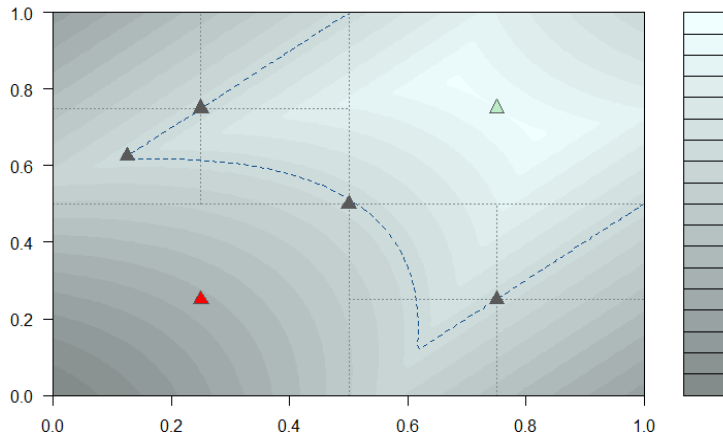
Exemple en dimension 2 : Étape 2



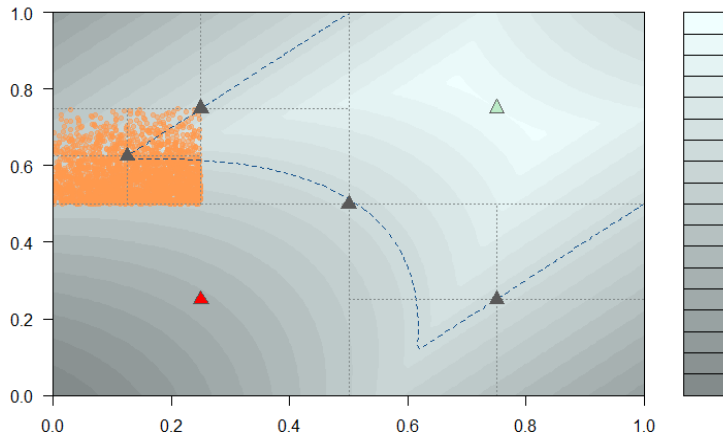
Exemple en dimension 2 : Étape 3



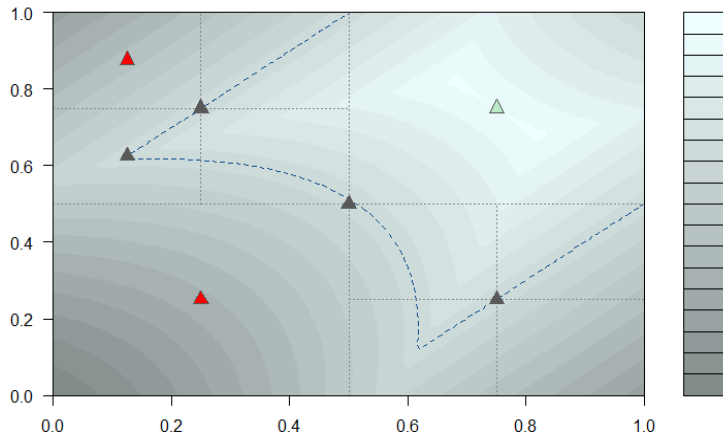
Exemple en dimension 2 : Étape 3



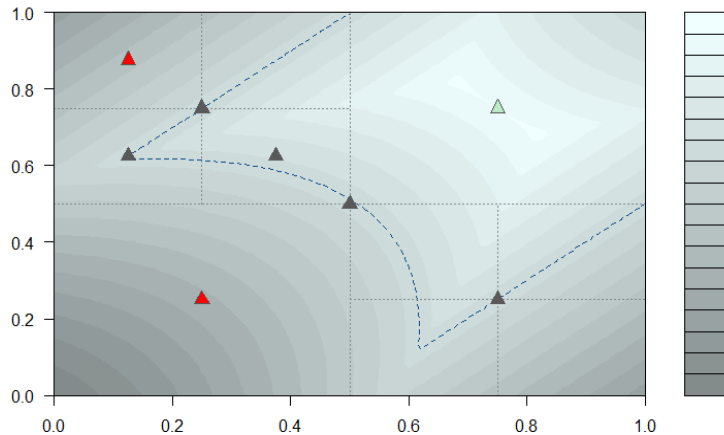
Exemple en dimension 2 : Étape 3



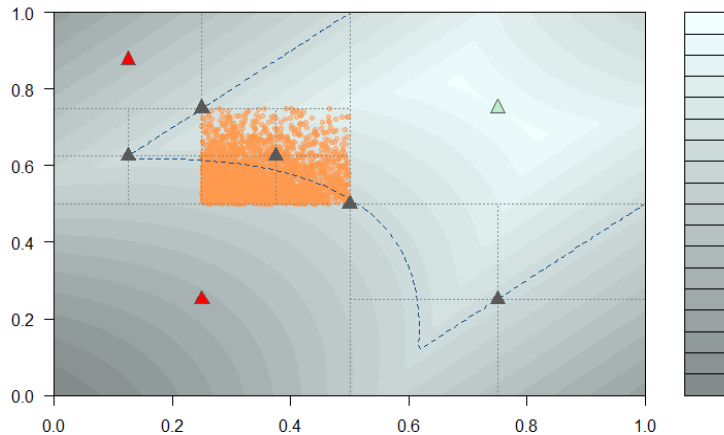
Exemple en dimension 2 : Étape 3



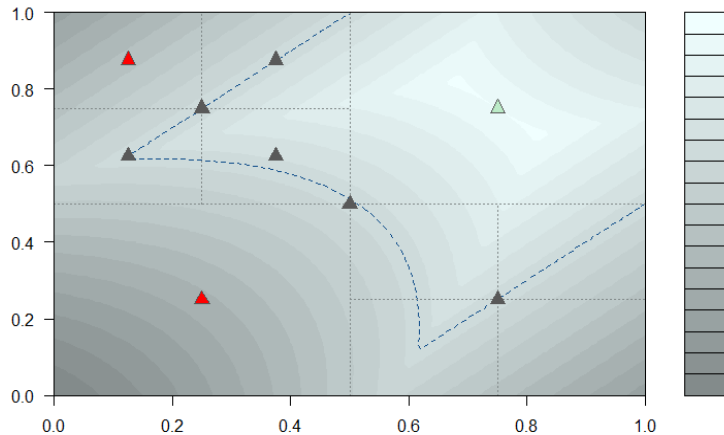
Exemple en dimension 2 : Étape 3



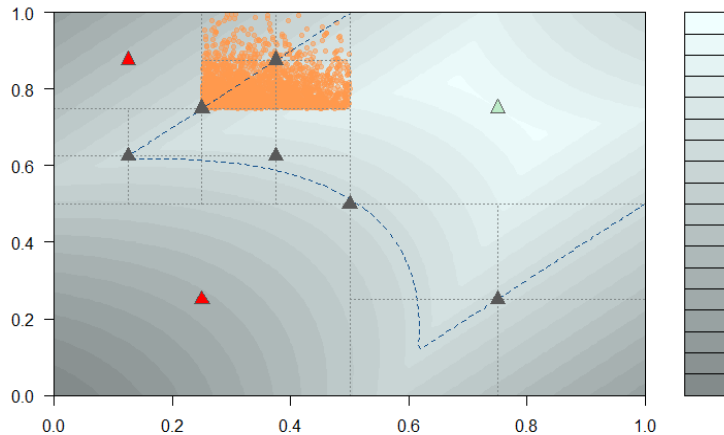
Exemple en dimension 2 : Étape 3



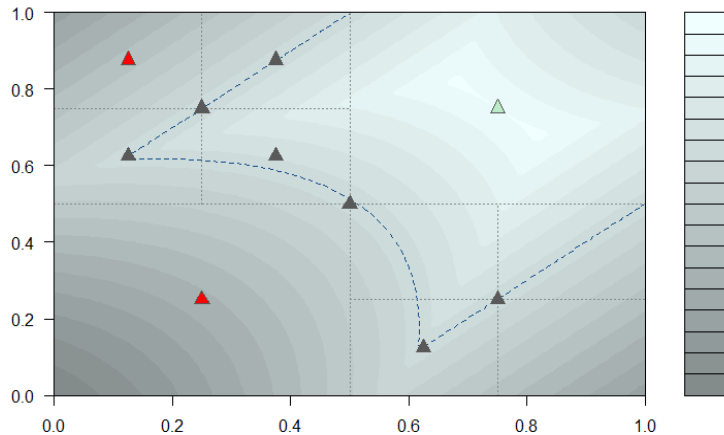
Exemple en dimension 2 : Étape 3



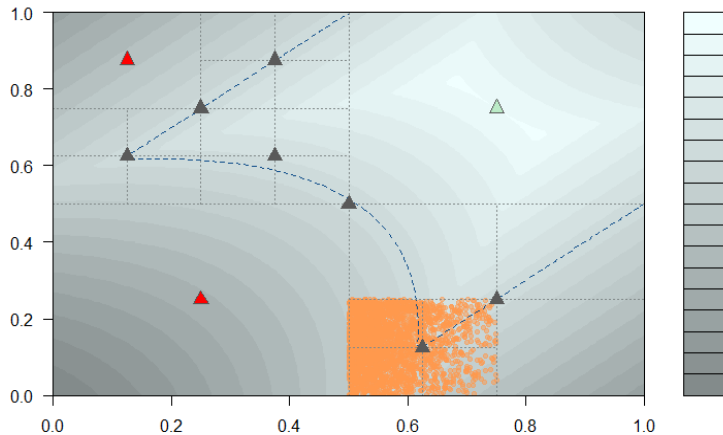
Exemple en dimension 2 : Étape 3



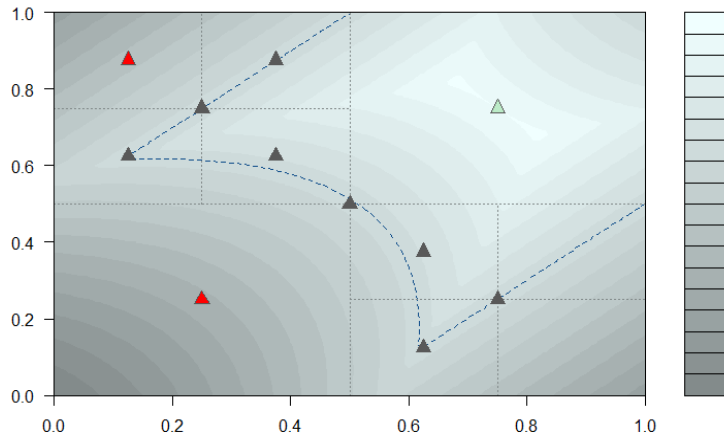
Exemple en dimension 2 : Étape 3



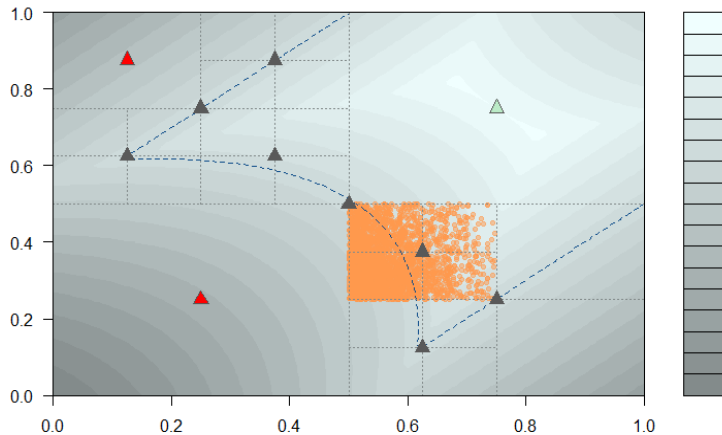
Exemple en dimension 2 : Étape 3



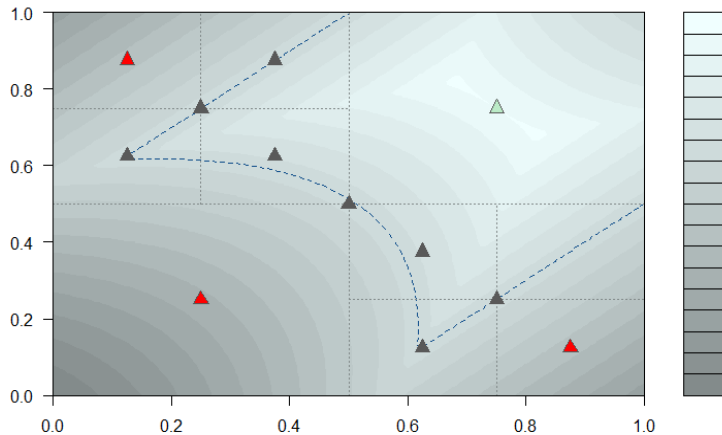
Exemple en dimension 2 : Étape 3



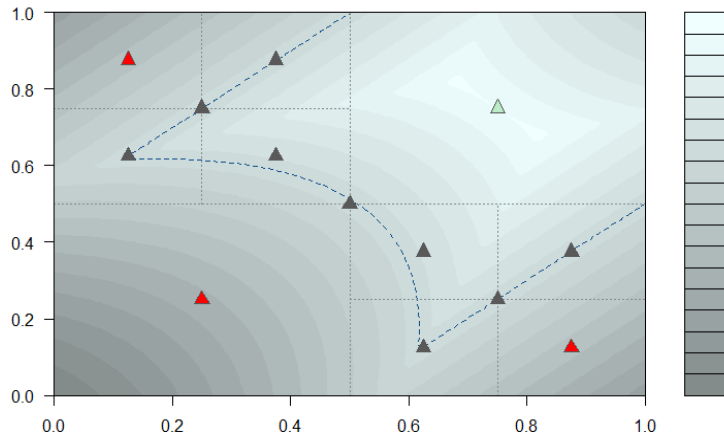
Exemple en dimension 2 : Étape 3



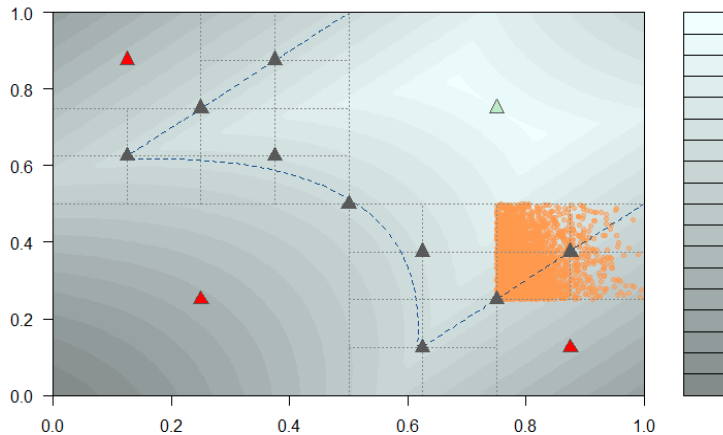
Exemple en dimension 2 : Étape 3



Exemple en dimension 2 : Étape 3



Exemple en dimension 2 : Étape 3





S.K. Au and J.L. Beck.

Estimation of small failure probabilities in high dimensions by subset simulation.

Journal of Probabilistic Engineering Mechanics, Vol. 16 :263–277, 2001.



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

Sequential Monte Carlo for rare event estimation.

Stat. Comput., 22(3) :795–808, 2012.



A. Cohen, R. Devore, G. Petrova, and P. Wojtaszczyk.

Finding the minimum of a function.

Methods Appl. Anal., 20(4) :365–381, 2013.