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

50èmes Journées de Statistique, Paris

Lucie Bernard - Institut Denis Poisson - LPSM - STMicroelectronics

Arnaud Guyader - LPSM - Sorbonne Université, Paris

Philippe Leduc - STMicroelectronis, Tours

Florent Malrieu - Institut Denis Poisson - Université Tours

- Soit $\mathbb{X} = [0,1]^d$ et $g : \mathbb{X} \to \mathbb{R}$, une fonction **boîte noire coûteuse à évaluer**.

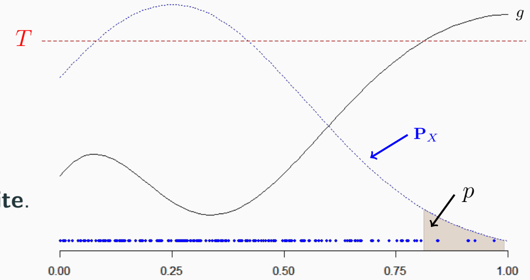
- À partir d'un **petit nombre d'évaluations** de g, on veut estimer la probabilité de défaillance p définie par :

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

où :

- $\mathbf{X}$ est une variable aléatoire dont la loi $\mathbf{P}_\mathbf{X}$ est connue à une constante de normalisation près,

- T est un seuil tel que p est **très petite**.



L'algorithme s'applique sous les conditions que,

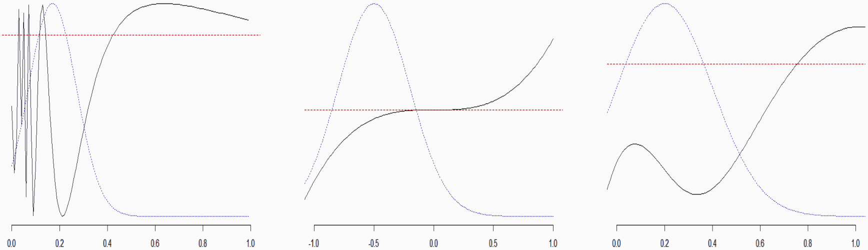
1. g est **Lipschitzienne** de constante M **connue** :

$$|g(\mathbf{x}) - g(\mathbf{x}')| \leq M \left\|\mathbf{x} - \mathbf{x}'\right\|_\infty, \quad \forall\, (\mathbf{x}, \mathbf{x}') \in \mathbb{X} \times \mathbb{X}.$$

2. g vérifie

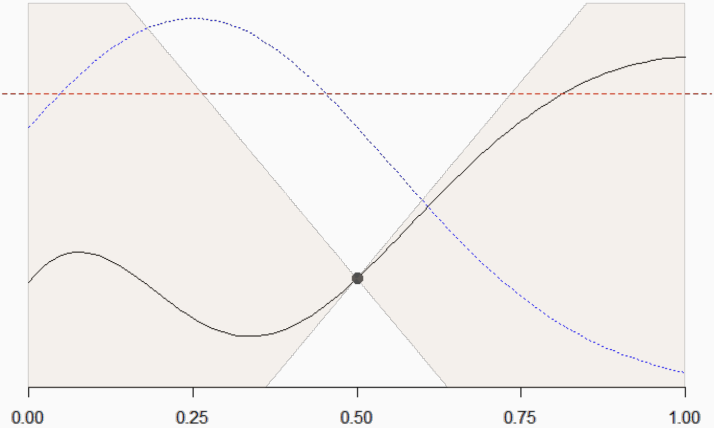
$$\lambda\left(\{\mathbf{x} \in \mathbb{X} : |g(\mathbf{x}) - T| \leq \delta\}\right) \leq L\delta^d, \quad \delta > 0,$$

où $\lambda(E)$ est la mesure de Lebesgue de l'ensemble E.

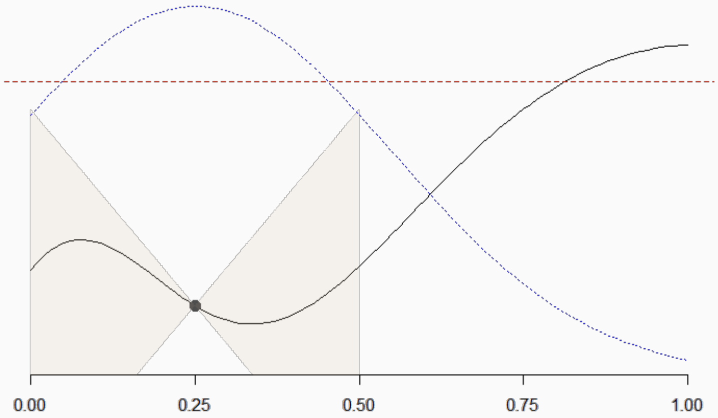


✓ ✓ ✓

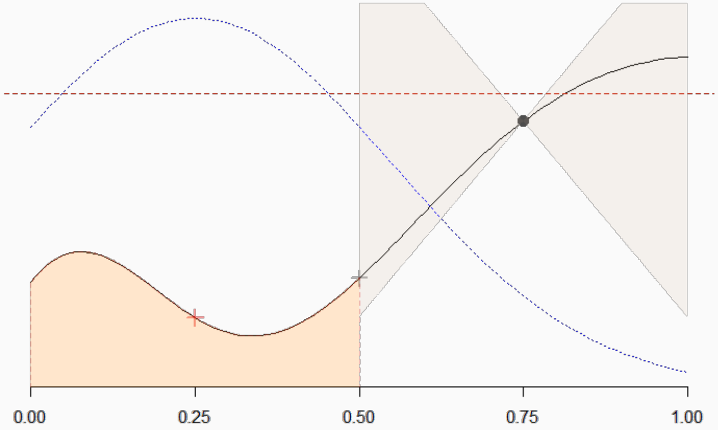
Exemple en dimension 1 : Étape 1



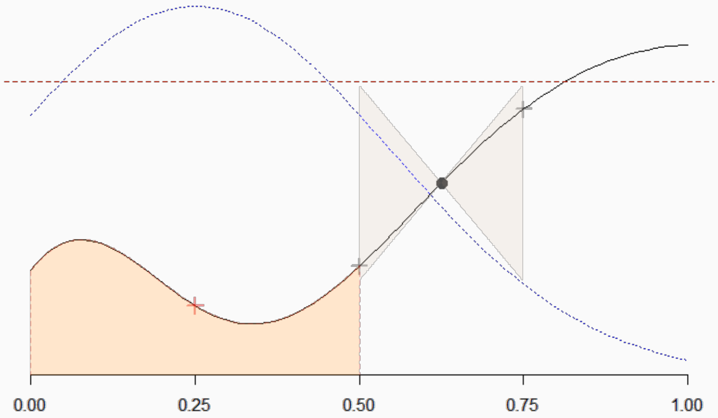
Exemple en dimension 1 : Étape 2



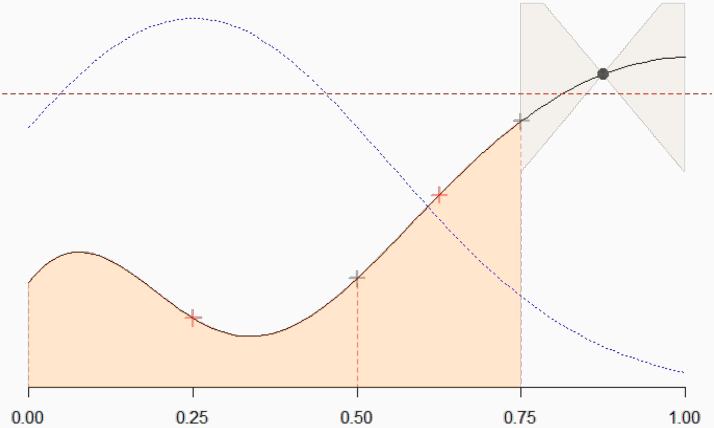
Exemple en dimension 1 : Étape 2



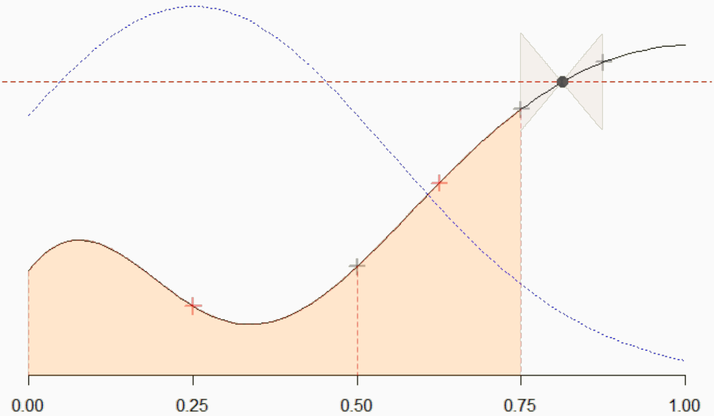
Exemple en dimension 1 : Étape 3



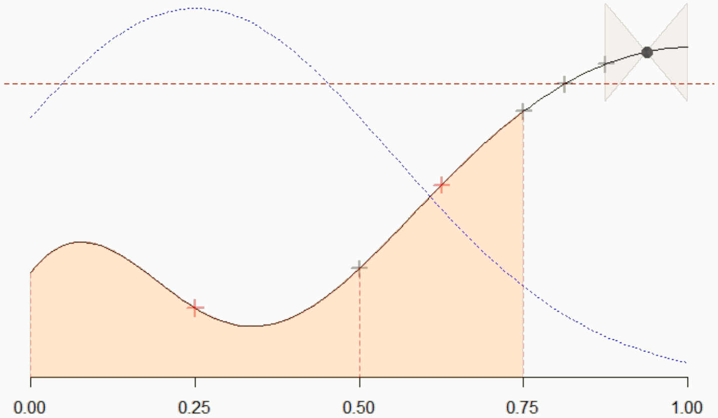
Exemple en dimension 1 : Étape 3



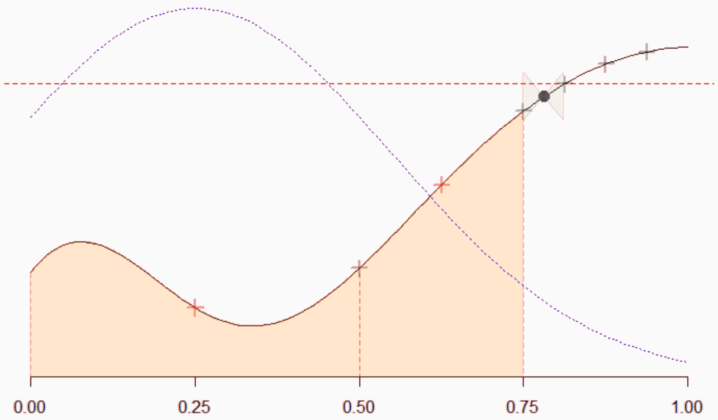
Exemple en dimension 1 : Étape 4



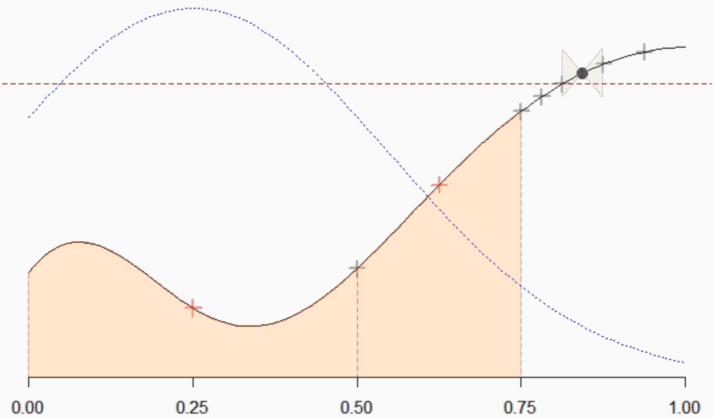
Exemple en dimension 1 : Étape 4



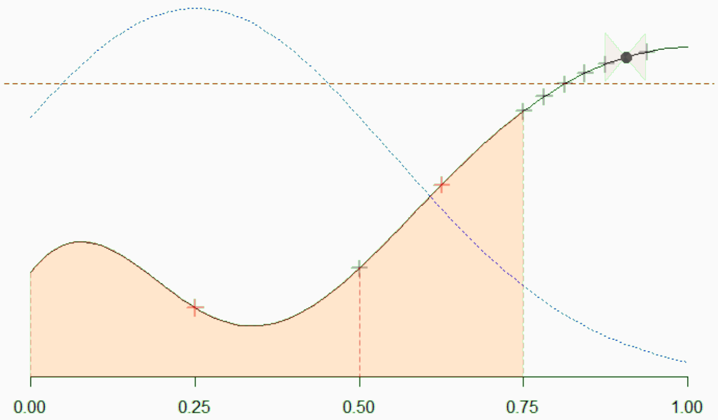
Exemple en dimension 1 : Étape 5



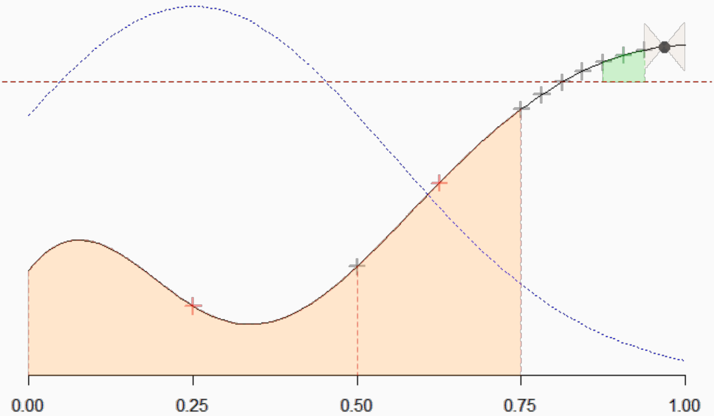
Exemple en dimension 1 : Étape 5



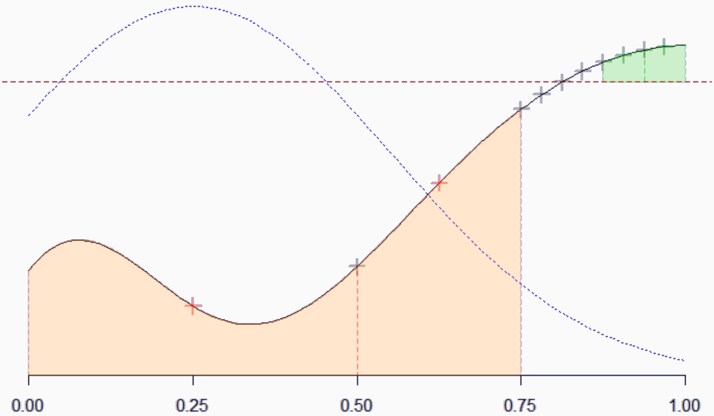
Exemple en dimension 1 : Étape 5



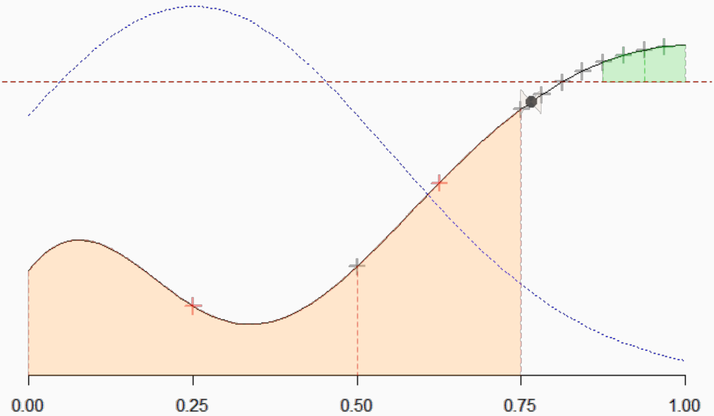
Exemple en dimension 1 : Étape 5



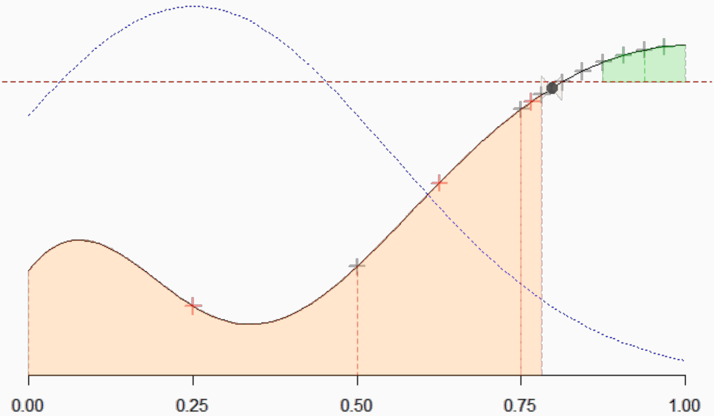
Exemple en dimension 1 : Étape 5



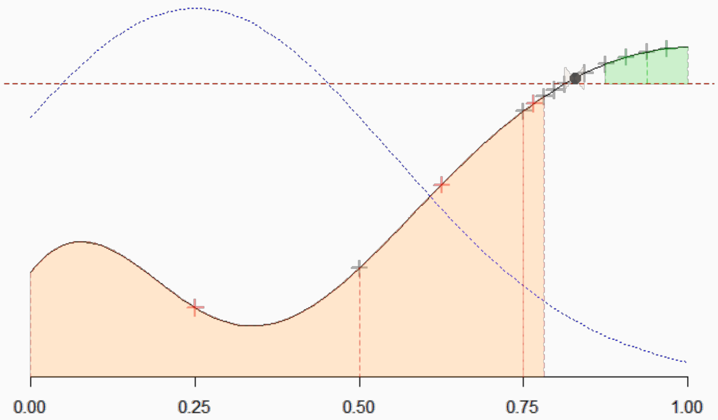
Exemple en dimension 1 : Étape 6



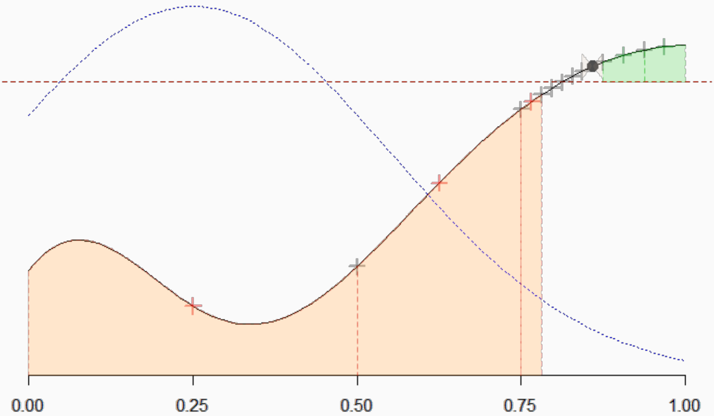
Exemple en dimension 1 : Étape 6



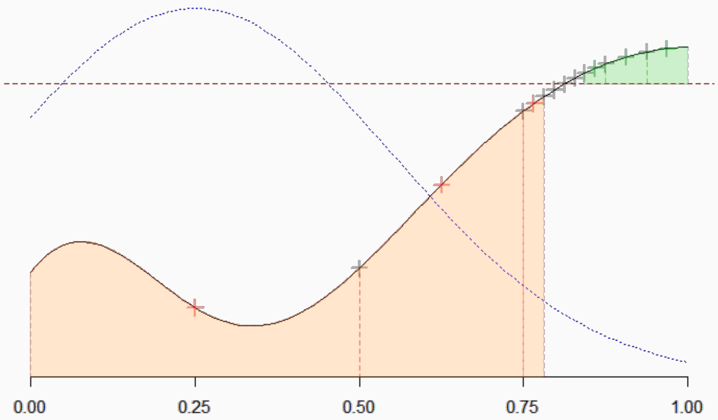
Exemple en dimension 1 : Étape 6



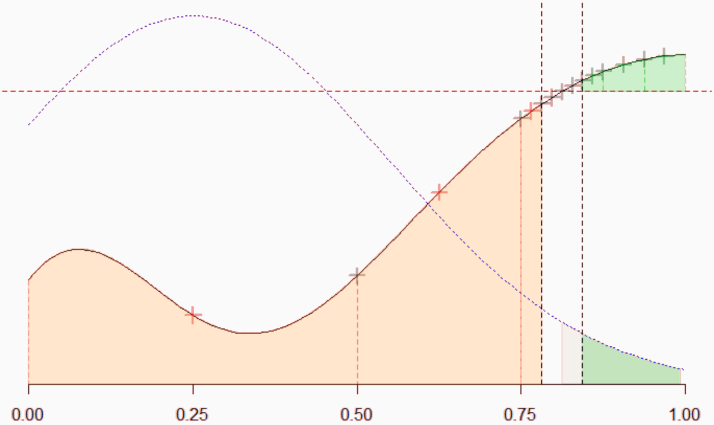
Exemple en dimension 1 : Étape 6



Exemple en dimension 1 : Étape 6



Exemple en dimension 1 : Étape 6



Formalisme

L'algorithme procède à une **partition partiellement dyadique** de $\mathbb{X} = [0,1]^d$ et utilise l'hypothèse selon laquelle g est Lipschitzienne pour **définir :**

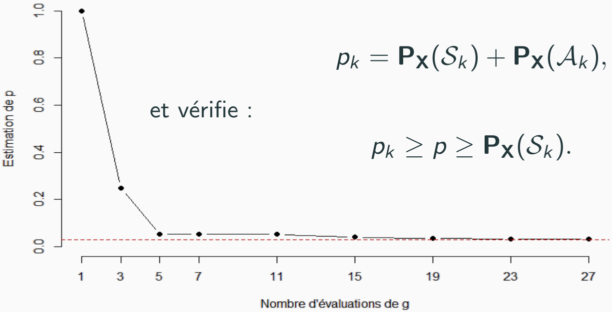
- Une suite décroissante $(\mathcal{A}_k)_{k\geq 0}$ de sous-domaines de $\mathbb{X}$ où g **peut dépasser** T,
- Une suite croissante $(\mathcal{S}_k)_{k\geq 0}$ de sous-domaines de $\mathbb{X}$ où g est **toujours au-dessus de** T,
- Une suite croissante $(\mathcal{O}_k)_{k\geq 0}$ de sous-domaines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

À chaque étape $k = 0, 1, \ldots$, une approximation p_k de p alors donnée par :

$$p_k = \mathbf{P}_{\mathbf{X}}(\mathcal{S}_k) + \mathbf{P}_{\mathbf{X}}(\mathcal{A}_k),$$

et vérifie :

$$p_k \geq p \geq \mathbf{P}_{\mathbf{X}}(\mathcal{S}_k).$$



Estimation de p

À chaque étape $k = 1, 2, \ldots$, on **estime** l'approximation p_k avec une méthode dite de multi-niveaux (*Multilevel Splitting*, aussi appelée *Subset Simulation*).

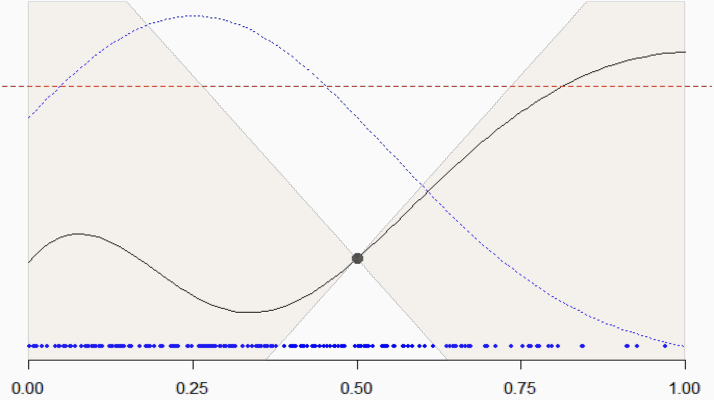
- On initialise $\mathbf{P}_{\mathbf{X}}(\mathcal{A}_0) = 1$.
- Pour tout $k = 1, 2, \ldots,$ on fait la décomposition suivante :

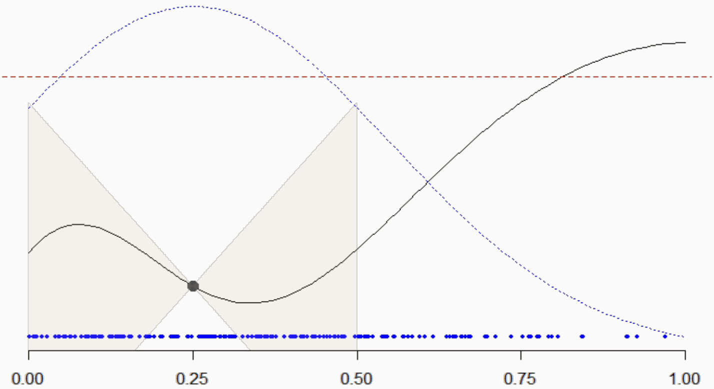
$$\mathbf{P}_{\mathbf{X}}(\mathcal{A}_k) = \sum_{i=1}^{\#(\mathcal{A}_k)} \mathbb{P}(\mathbf{X} \in Q_i) = \sum_{i=1}^{\#(\mathcal{A}_k)} \mathbb{P}(\mathbf{X} \in Q_i | \mathbf{X} \in \mathcal{A}_{k-1}) \mathbf{P}_{\mathbf{X}}(\mathcal{A}_{k-1}).$$

- On simule un échantillon $\mathbf{X}_1, \ldots, \mathbf{X}_N \sim \mathcal{L}(\mathbf{X} | \mathbf{X} \in \mathcal{A}_{k-1})$ avec un algorithme de Metropolis-Hasting et on fait l'approximation suivante :

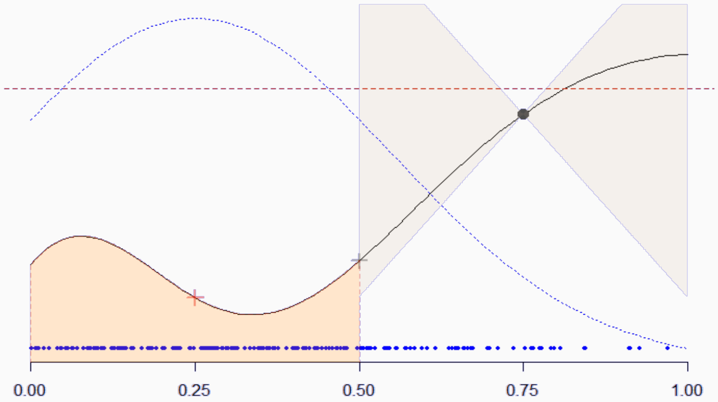
$$\mathbb{P}(\mathbf{X} \in Q_i | \mathbf{X} \in \mathcal{A}_{k-1}) \backsimeq \frac{1}{N} \sum_{j=1}^{N} \mathbb{1}_{\mathbf{X}_j \in Q_i}.$$

Estimation de p : Exemple en dimension 1.

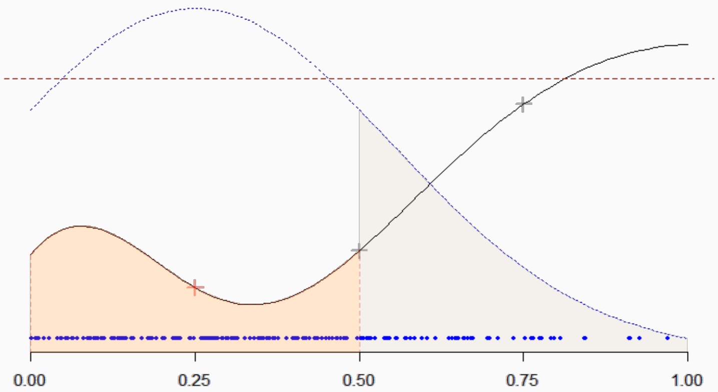



0.00
0.25
0.50
0.75
1.00

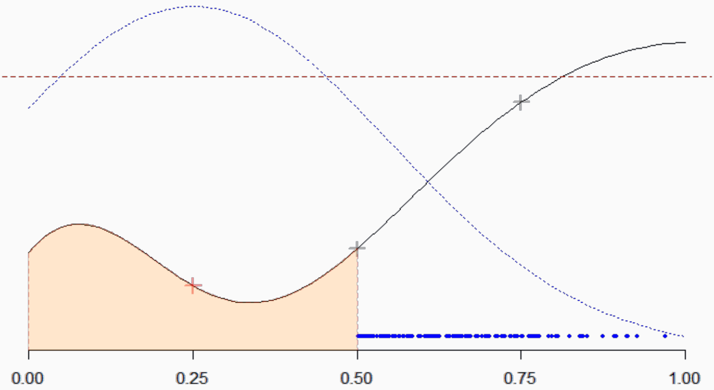
Estimation de p : Exemple en dimension 1.



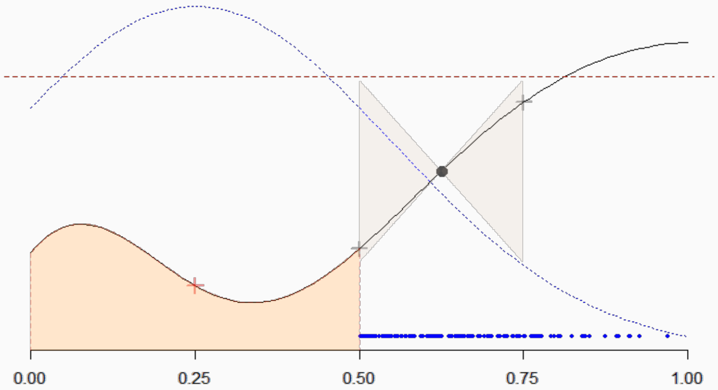
Estimation de p : Exemple en dimension 1.



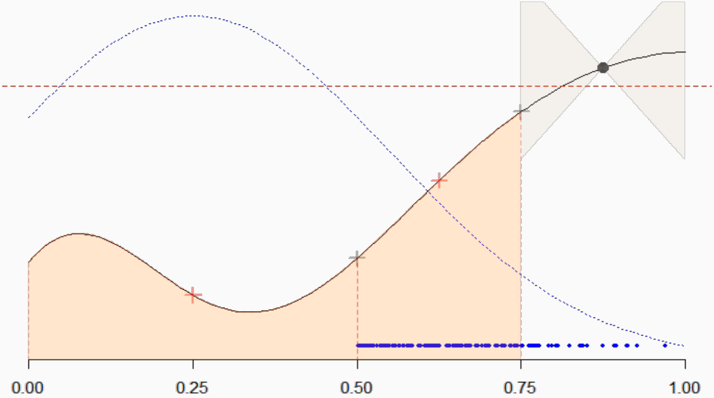
Estimation de p : Exemple en dimension 1.



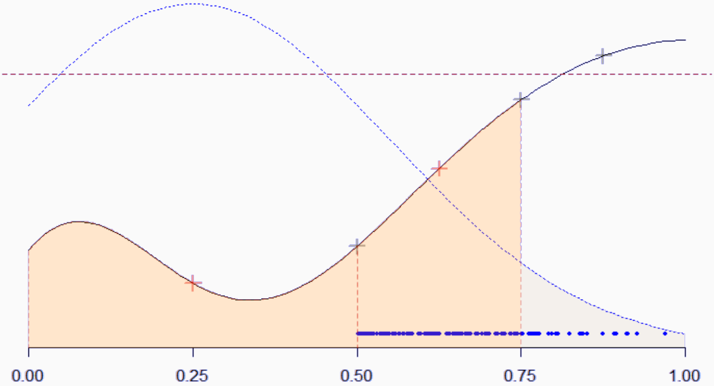
Estimation de p : Exemple en dimension 1.



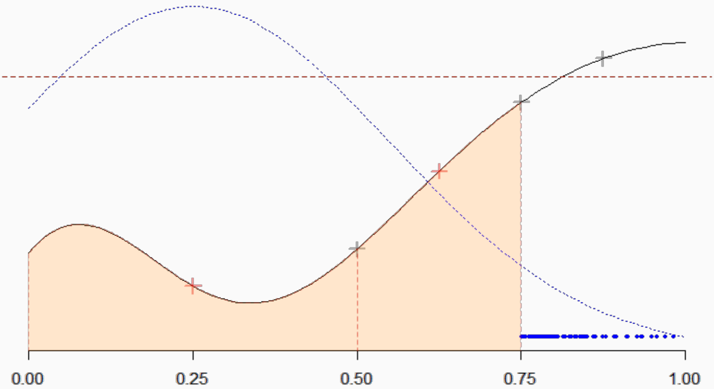
Estimation de p : Exemple en dimension 1.



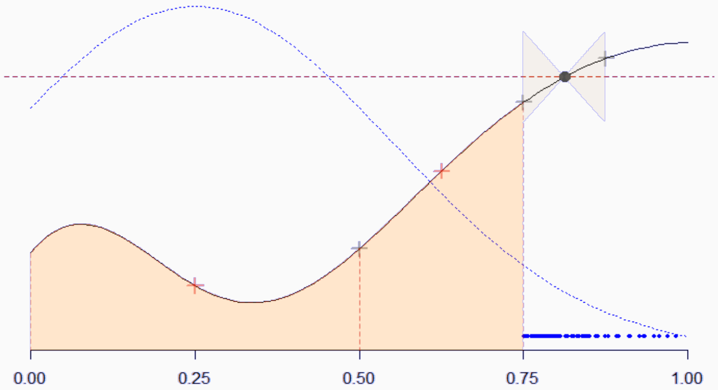
Estimation de p : Exemple en dimension 1.



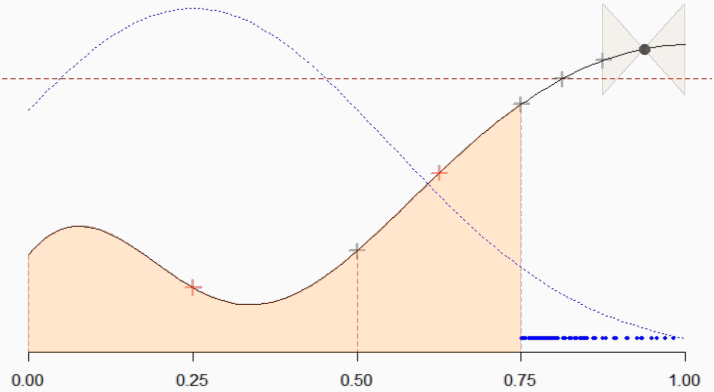
Estimation de p : Exemple en dimension 1.



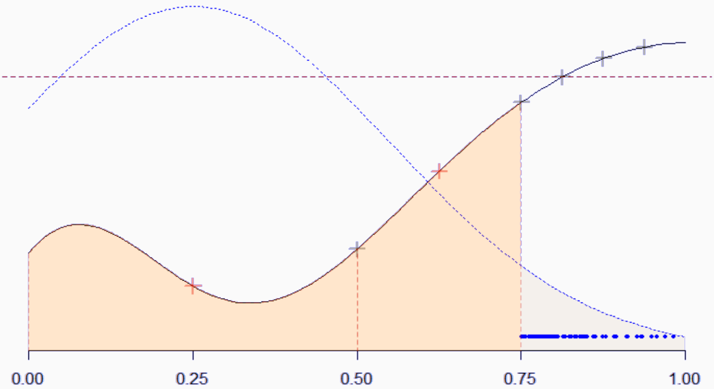
Estimation de p : Exemple en dimension 1.



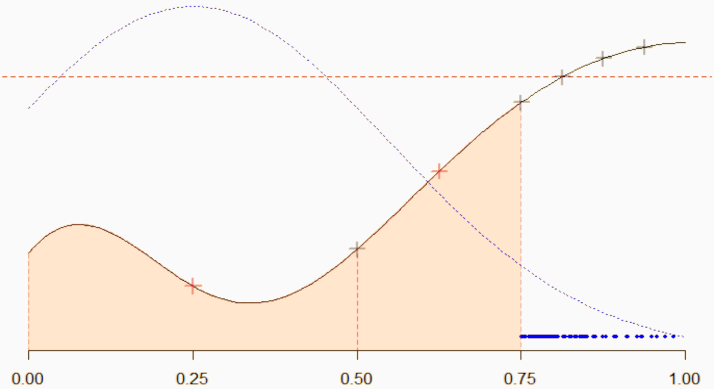
Estimation de p : Exemple en dimension 1.



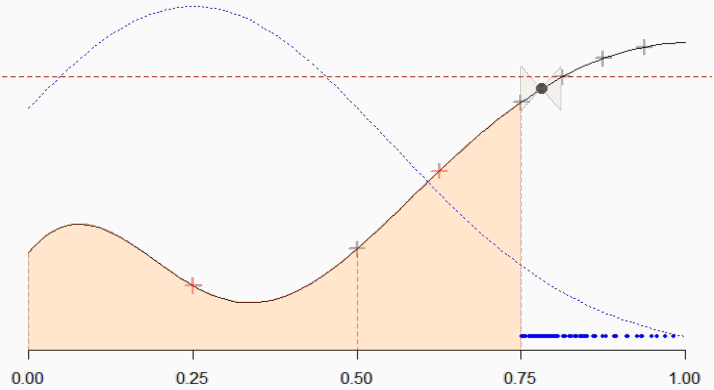
Estimation de p : Exemple en dimension 1.



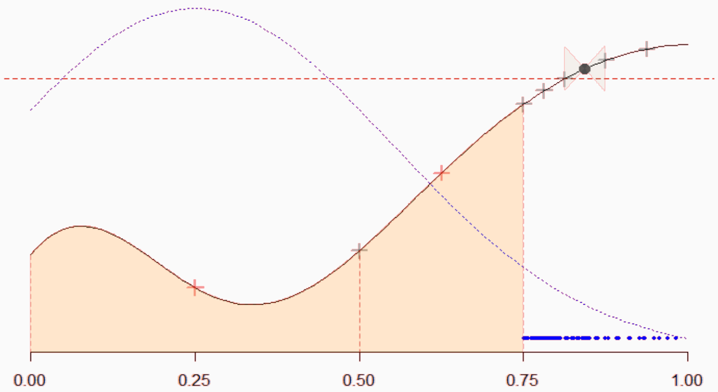
Estimation de p : Exemple en dimension 1.



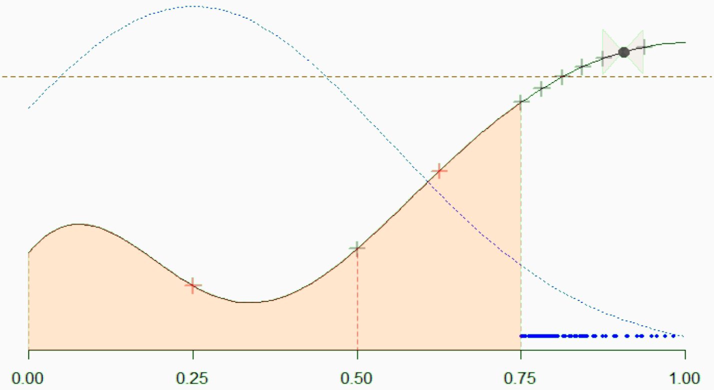
Estimation de p : Exemple en dimension 1.



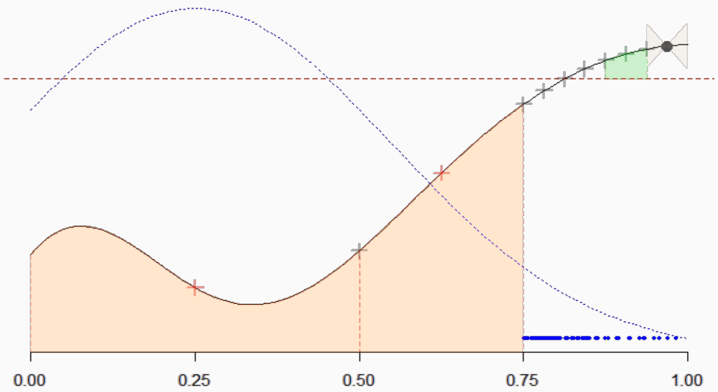
Estimation de p : Exemple en dimension 1.



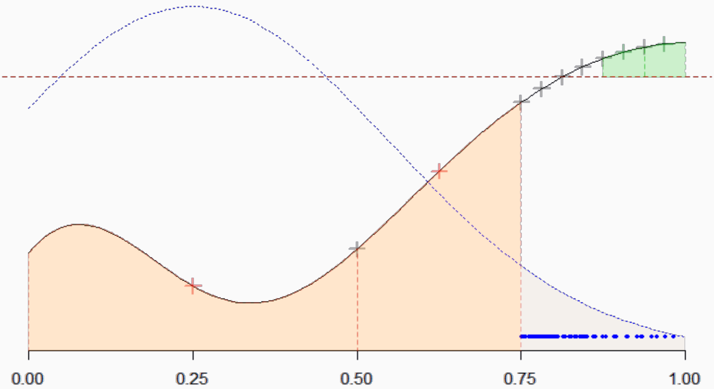
Estimation de p : Exemple en dimension 1.



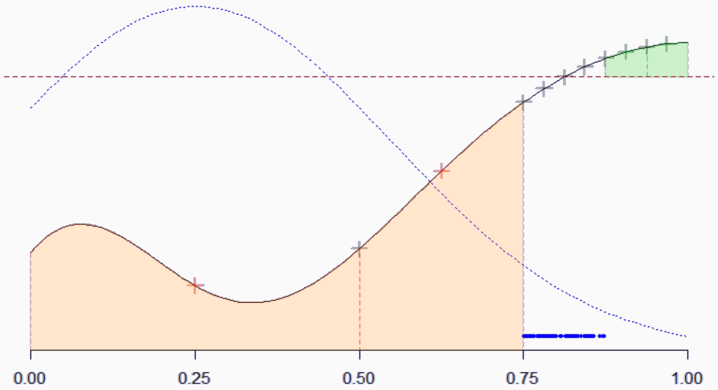
Estimation de p : Exemple en dimension 1.



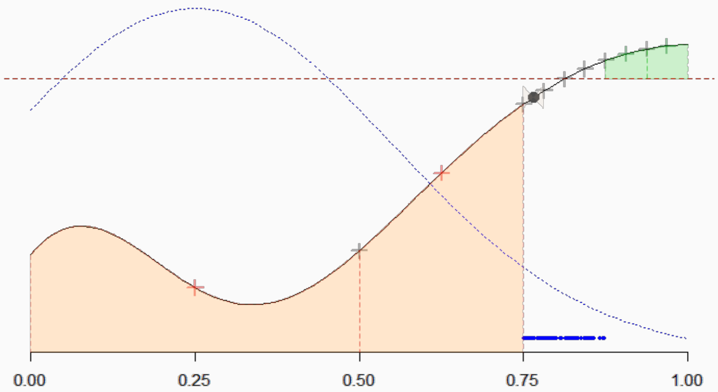
Estimation de p : Exemple en dimension 1.



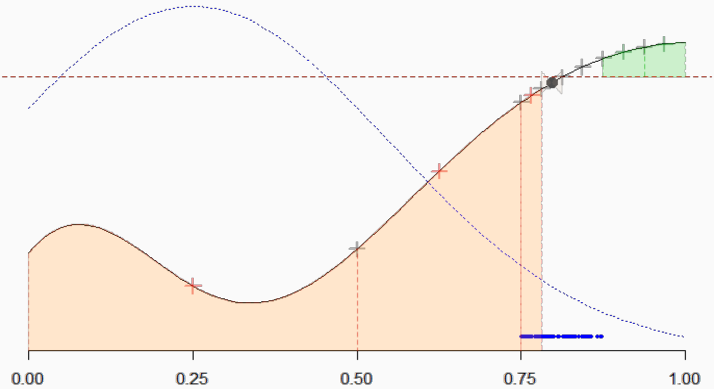
Estimation de p : Exemple en dimension 1.



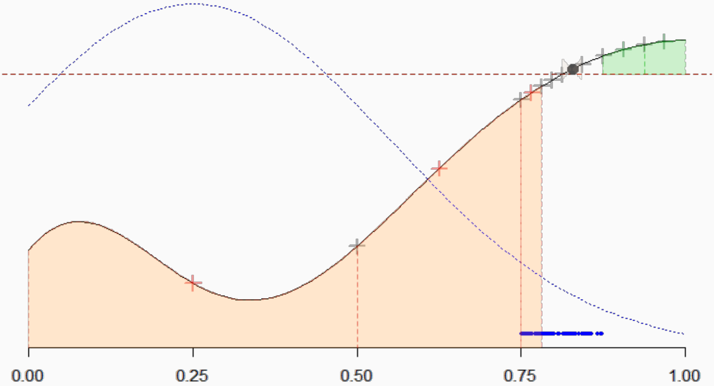
Estimation de p : Exemple en dimension 1.



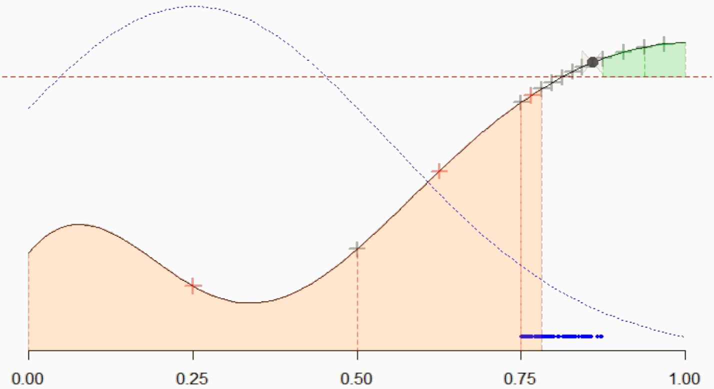
Estimation de p : Exemple en dimension 1.



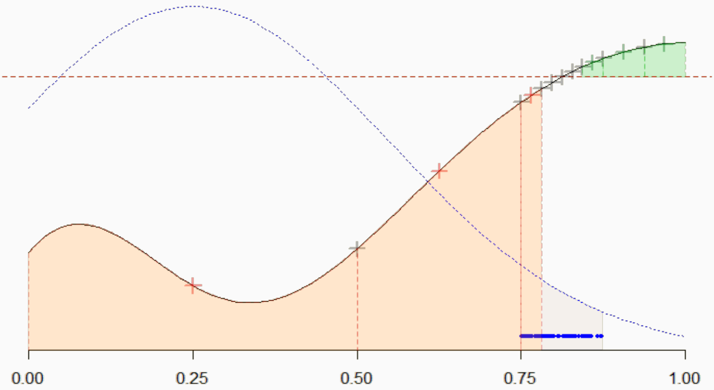
Estimation de p : Exemple en dimension 1.



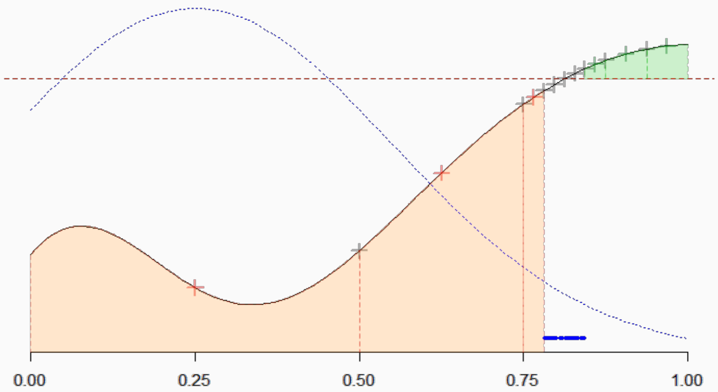
Estimation de p : Exemple en dimension 1.



Estimation de p : Exemple en dimension 1.



Estimation de p : Exemple en dimension 1.



Proposition

1. *Soit* $m = \sup_{\mathbf{x} \in A} f_X(\mathbf{x})$, *où* f_X *est la densité de* $\mathbf{X}$. *Pour tout* $k \geq 1$ *et* $d \geq 1$, *on a :*

$$p \leq p_k \leq p + LM^d \frac{m}{2^{d(k-1)}}.$$

2. *Soit* n_k *le* **nombre total d'évaluations de la fonction** g *à la fin de l'étape* k. *Pour tout* $k \geq 2$ *et* $d \geq 1$, *on a :*

$$n_k \leq 1 + LM^d 2^d (k-1).$$

Corollaire

*Soit p_n, $n \geq 2$, l'***approximation** *de p associée au nombre n d'évaluations de g. Pour tout $d \geq 1$, l'erreur d'approximation*

$$E_n = p_n - p,$$

vérifie :

$$0 \leq E_n \leq 2LM^d m \times 2^{-nd/(2^d LM^d)}.$$

▷ Par conséquent, en notant $\widehat{p}_n^N$ l'estimation de p_n obtenue par la méthode multi-niveaux, on a :

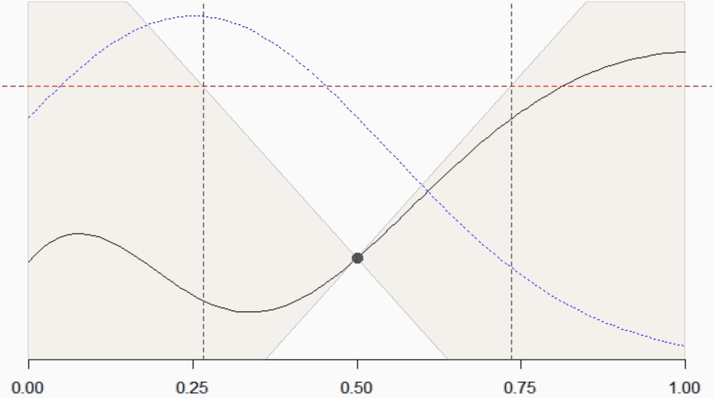
$$|\widehat{p}_n^N - p| = 2LM^d m \times 2^{-nd/(2^d LM^d)} + O_{\mathbb{P}}\left(\frac{1}{\sqrt{N}}\right).$$

F. Cérou, P. Del Moral, T. Furon, and A. Guyader.
Sequential Monte Carlo for rare event estimation.
Stat. Comput., 22(3) :795–808, 2012.

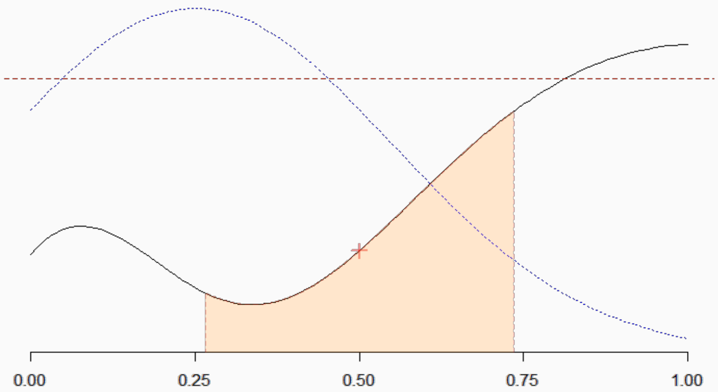
A. Cohen, R. Devore, G. Petrova, and P. Wojtaszczyk.
Finding the minimum of a function.
Methods Appl. Anal., 20(4) :365–381, 2013.

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.

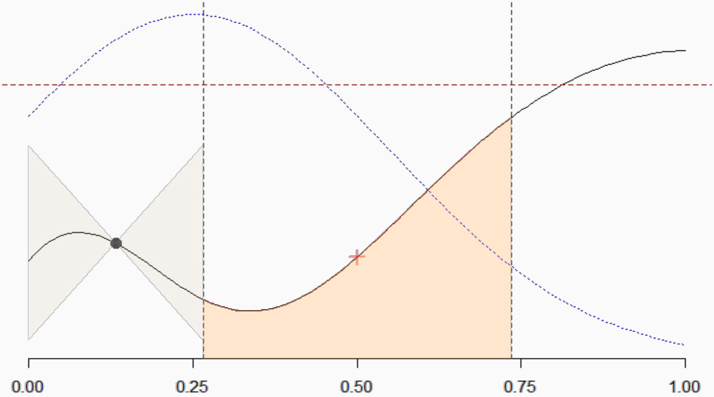
Amélioration de la performance en pratique



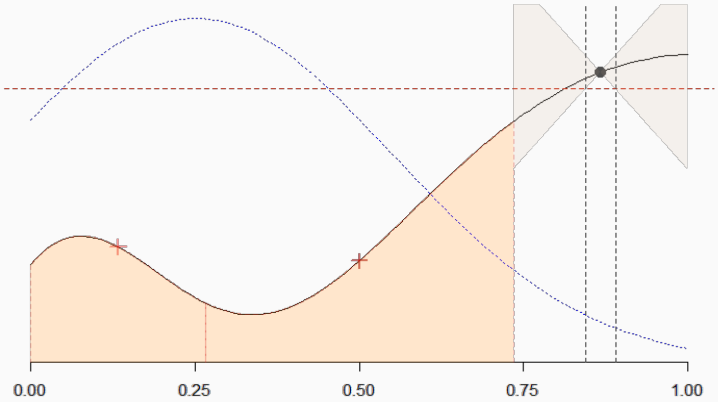
Amélioration de la performance en pratique



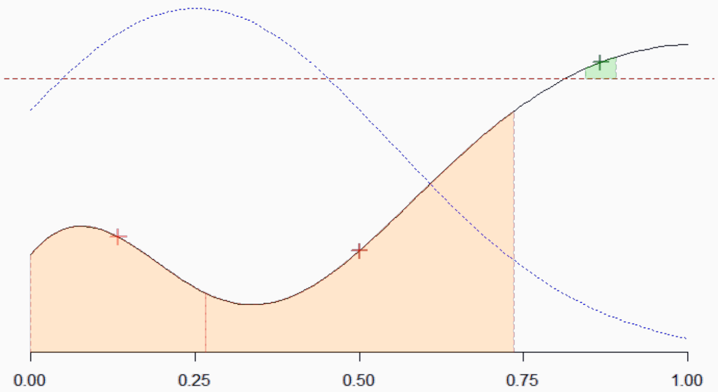
Amélioration de la performance en pratique



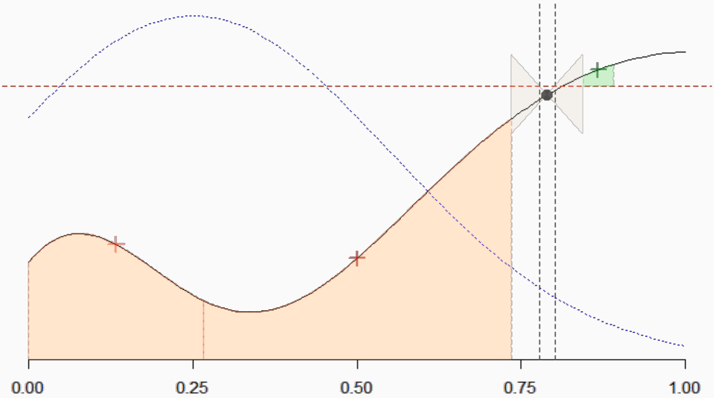
Amélioration de la performance en pratique



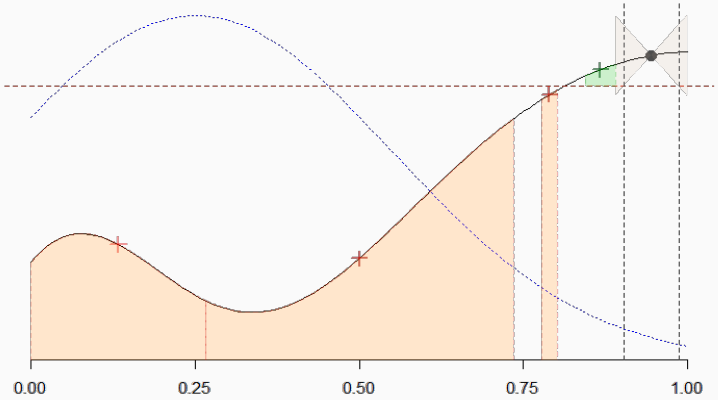
Amélioration de la performance en pratique



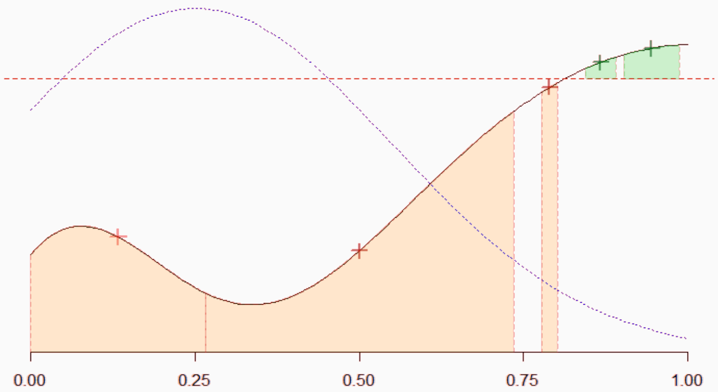
Amélioration de la performance en pratique



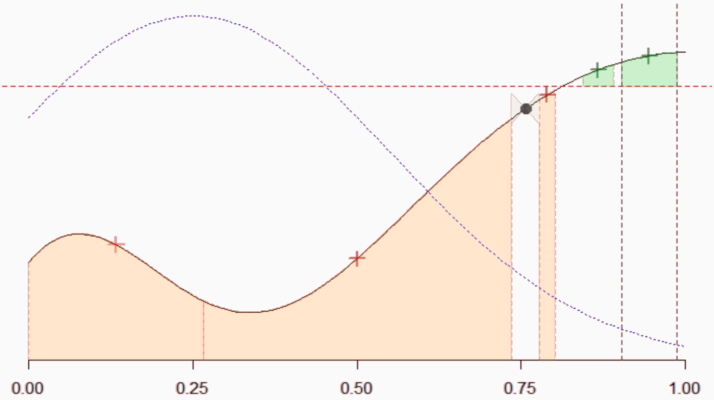
Amélioration de la performance en pratique



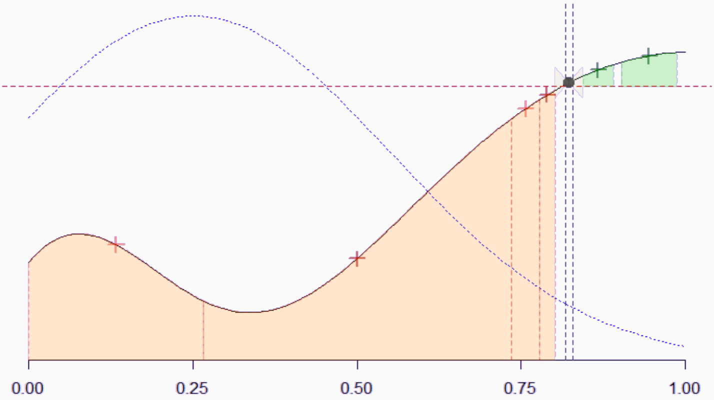
Amélioration de la performance en pratique



Amélioration de la performance en pratique



Amélioration de la performance en pratique



Amélioration de la performance en pratique