

Récursivité

Admettons que nous disposons d'une écriture intégrale de la grandeur \mathcal{A} que nous cherchons à estimer, mais que cette intégrale fait intervenir la grandeur \mathcal{A} elle-même : \mathcal{A} est connue comme la solution d'une *équation intégrale* (une équation de Fredholm). Admettons de plus que cette équation est linéaire :

$$\mathcal{A} = \int_{\mathcal{D}} \alpha(x)\mathcal{A} + \beta(x) dx \quad (1)$$

où a et b sont des fonctions connues sur \mathcal{D} .

Nous procédons alors en trois étapes :

1. On probabilise l'intégrale en introduisant une variable d'échantillonnage de \mathcal{D} , comme nous l'avons fait pour la première intégrale dans le TP2.
2. On probabilise la somme $\alpha(x)\mathcal{A} + \beta(x)$ de façon à l'écrire comme une espérance. On dispose alors d'un algorithme de Bernoulli pour l'estimation de cette espérance, exactement comme nous l'avons fait dans l'exemple du TP1 où nous estimions $4 + 1$ par Monte Carlo.
3. On utilise la double randomisation pour combiner les deux, comme nous l'avons fait pour combiner les deux intégrales du TP2.

Probabilisation de l'intégrale On se donne une variable aléatoire X sur $\mathcal{U}_X = \mathcal{D}$ de densité de probabilité p_X . Cette densité de probabilité doit être non nulle en tous les points du domaine. On multiplie et on divise l'intégrande par p_X pour écrire

$$\mathcal{A} = \int_{\mathcal{D}} p_X(x) dx \left(\frac{\alpha(x)}{p_X(x)} \mathcal{A} + \frac{\beta(x)}{p_X(x)} \right) \quad (2)$$

Cette écriture se lit comme un premier algorithme : on crée une réalisation x la variable aléatoire X et la réalisation correspondante du poids de Monte Carlo est $w = \frac{\alpha(x)}{p_X(x)} \mathcal{A} + \frac{\beta(x)}{p_X(x)}$. La moyenne d'un grand nombre de tels poids produira une estimation de \mathcal{A} .

Probabilisation de la somme On se donne une probabilité P et on écrit

$$\frac{\alpha(x)}{p_X(x)} \mathcal{A} + \frac{\beta(x)}{p_X(x)} = P \left[\frac{\alpha(x)}{p_X(x)P} \mathcal{A} \right] + (1 - P) \left[\frac{\beta(x)}{p_X(x)(1 - P)} \right] \quad (3)$$

Cette écriture se lit comme un second algorithme : on crée une réalisation b d'une variable aléatoire de Bernoulli B de probabilité P ; si la réalisation est $b = 1$, on retient le poids de Monte Carlo $w = \frac{\alpha(x)}{p_X(x)P} \mathcal{A}$; si la réalisation est $b = 0$, on retient le poids de Monte Carlo $w = \frac{\beta(x)}{p_X(x)(1 - P)}$. La moyenne d'un grand nombre de tels poids produira une estimation de $\frac{\alpha(x)}{p_X(x)} \mathcal{A} + \frac{\beta(x)}{p_X(x)}$.

Double randomisation On combine les deux écritures précédentes pour obtenir

$$\mathcal{A} = \int_{\mathcal{D}} p_X(x) dx \left(P \left[\frac{\alpha(x)}{p_X(x)P} \mathcal{A} \right] + (1 - P) \left[\frac{\beta(x)}{p_X(x)(1 - P)} \right] \right) \quad (4)$$

qui se lit comme l'algorithme suivant : on crée une réalisation x de la variable aléatoire X ; on crée une réalisation b d'une variable aléatoire de Bernoulli B de probabilité P ; si la réalisation est $b = 1$, on retient le poids de Monte Carlo $w = \frac{\alpha(x)}{p_X(x)P} \mathcal{A}$; si la réalisation est $b = 0$, on retient le poids de Monte Carlo $w = \frac{\beta(x)}{p_X(x)(1 - P)}$. La moyenne d'un grand nombre de tels poids produira une estimation de \mathcal{A} .

Récursivité L'algorithme qui précède estime \mathcal{A} mais il n'est praticable que si \mathcal{A} est déjà connu. En effet, lorsque la réalisation de la variable de Bernoulli est $b = 1$ le poids de Monte Carlo $w = \frac{\alpha(x)}{p_X(x)P} \mathcal{A}$ contient \mathcal{A} lui-même. Cependant, ce poids est une fonction linéaire de \mathcal{A} . Dans la littérature sur la méthode de Monte Carlo, quand certaines branches de l'algorithme conduisent à une expression du poids qui est une fonction linéaire de la grandeur à estimer, la traduction algorithmique de la formulation intégrale est à penser de façon récursive : l'apparition de \mathcal{A} signifie que l'algorithme se poursuit, avec une nouvelle réalisation de X et une nouvelle réalisation de B , conduisant à une nouvelle expression d'un poids intermédiaire qui prend la place de \mathcal{A} dans $w = \frac{\alpha(x)}{p_X(x)P} \mathcal{A}$. Bien sûr cette nouvelle expression du poids total peut elle-même impliquer \mathcal{A} , ce qui se traduit par encore une nouvelle réalisation de X et une nouvelle réalisation de B , etc. Cette récursivité demande une gestion du préfacteur $\frac{\alpha(x)}{p_X(x)P}$ qui est illustrée dans l'algorithme 1.

Algorithme 1 : Exemple d'algorithme récursif

```

1 somme = 0;
2 sommeDesCarres = 0;
3 pour chaque  $i \in \langle 1 : n \rangle$  faire
4   |   recursivite = vrai;
5   |   prefacteur = 1;
6   |   tant que recursivite faire
7   |   |   Réalisation  $x$  de  $X$ ;
8   |   |   Réalisation  $b$  d'une loi de Bernoulli  $B$  de probabilité  $P$ ;
9   |   |   si  $b=1$  alors
10  |   |   |   prefacteur = prefacteur *  $\frac{\alpha(x)}{p_X(x)P}$ ;
11  |   |   |   sinon
12  |   |   |   |    $w = \textit{prefacteur} * \frac{\beta(x)}{p_X(x)(1-P)}$ ;
13  |   |   |   |   recursivite = faux;
14  |   |   fin
15  |   fin
16  |   somme = somme +  $w$ ;
17  |   sommeDesCarres = sommeDesCarres +  $w^2$ ;
18 fin
19  $m = \frac{\textit{somme}}{n}$ ;
20  $s = \frac{1}{\sqrt{n-1}} \sqrt{\frac{\textit{sommeDesCarres}}{n} - m^2}$ ;
```

Développement correspondant On justifie formellement cet algorithme en posant les écritures en espérance. L'équation 4 signifie

$$\mathcal{A} = \mathbb{E} \left(\mathcal{H}(B) \left[\frac{\alpha(X)}{p_X(X)P} \mathcal{A} \right] + \mathcal{H}(1-B) \left[\frac{\beta(X)}{p_X(X)(1-P)} \right] \right) \quad (5)$$

où la fonction de Heaviside \mathcal{H} est définie de sorte que $\mathcal{H}(1) = 1$ et $\mathcal{H}(0) = 0$. On peut alors définir deux variables aléatoires X_1 et X_2 indistinctement distribuées et indépendantes, qui suivent la même loi que X , et de même définir deux variables aléatoires B_1 et B_2 indistinctement distribuées et indépendantes, qui suivent la même

loi que B , pour écrire \mathcal{A} de deux façons différentes :

$$\begin{aligned}\mathcal{A} &= \mathbb{E} \left(\mathcal{H}(B_1) \left[\frac{\alpha(X_1)}{p_X(X_1)P} \mathcal{A} \right] + \mathcal{H}(1 - B_1) \left[\frac{\beta(X_1)}{p_X(X_1)(1 - P)} \right] \right) \\ \mathcal{A} &= \mathbb{E} \left(\mathcal{H}(B_2) \left[\frac{\alpha(X_2)}{p_X(X_2)P} \mathcal{A} \right] + \mathcal{H}(1 - B_2) \left[\frac{\beta(X_2)}{p_X(X_2)(1 - P)} \right] \right)\end{aligned}\tag{6}$$

En reportant l'une dans l'autre on obtient :

$$\mathcal{A} = \mathbb{E} \left(\mathcal{H}(B_1) \left[\frac{\alpha(X_1)}{p_X(X_1)P} \mathbb{E} \left(\mathcal{H}(B_2) \left[\frac{\alpha(X_2)}{p_X(X_2)P} \mathcal{A} \right] + \mathcal{H}(1 - B_2) \left[\frac{\beta(X_2)}{p_X(X_2)(1 - P)} \right] \right) \right] + \mathcal{H}(1 - B_1) \left[\frac{\beta(X_1)}{p_X(X_1)(1 - P)} \right] \right)\tag{7}$$

Avec toujours la même idée essentielle de la double randomisation (l'espérance d'une fonction linéaire d'une espérance est une espérance), on obtient

$$\mathcal{A} = \mathbb{E} \left(\mathcal{H}(B_1) \left[\frac{\alpha(X_1)}{p_X(X_1)P} \mathcal{H}(B_2) \left[\frac{\alpha(X_2)}{p_X(X_2)P} \mathcal{A} \right] + \mathcal{H}(1 - B_2) \left[\frac{\beta(X_2)}{p_X(X_2)(1 - P)} \right] \right] + \mathcal{H}(1 - B_1) \left[\frac{\beta(X_1)}{p_X(X_1)(1 - P)} \right] \right)\tag{8}$$

soit en développant,

$$\begin{aligned}\mathcal{A} &= \mathbb{E} \left(\mathcal{H}(B_1) \frac{\alpha(X_1)}{p_X(X_1)P} \mathcal{H}(B_2) \frac{\alpha(X_2)}{p_X(X_2)P} \mathcal{A} \right. \\ &\quad \left. + \mathcal{H}(B_1) \frac{\alpha(X_1)}{p_X(X_1)P} \mathcal{H}(1 - B_2) \frac{\beta(X_2)}{p_X(X_2)(1 - P)} \right. \\ &\quad \left. + \mathcal{H}(1 - B_1) \frac{\beta(X_1)}{p_X(X_1)(1 - P)} \right)\end{aligned}\tag{9}$$

Ce développement peut être étendu à l'infini (introduire $X_3, B_3, X_4, B_4 \dots$ pour remplacer \mathcal{A} à chaque étape) pour aboutir à

$$\mathcal{A} = \mathbb{E} \left(\sum_{i=1}^{+\infty} \left[\prod_{j=1}^{i-1} \mathcal{H}(B_j) \frac{\alpha(X_j)}{p_X(X_j)P} \right] \mathcal{H}(1 - B_i) \frac{\beta(X_i)}{p_X(X_i)(1 - P)} \right)\tag{10}$$

ce qui se traduit bien en l'algorithme 1.

Symbolique

Reprenons de l'exemple que nous avons utilisé pour le calcul de sensibilité dans la fiche d'introduction. On suppose que l'on dispose d'un algorithme de Monte Carlo estimant $\mathcal{A}(\alpha)$ de la façon suivante (voir la section sur la formulation intégrale) :

$$\mathcal{A}(\alpha) = \int_{\mathcal{U}_X} p_X(x; \alpha) dx \tilde{w}(x; \alpha)\tag{11}$$

Dans le calcul de sensibilité, l'objectif était d'estimer la dérivée $\frac{\partial \mathcal{A}}{\partial \alpha}$, c'est à dire la dérivée de la grandeur étudiée par rapport à un paramètre du problème. Le paramètre impacte à la fois la loi d'échantillonnage p_X et la fonction poids \tilde{w} . Il n'affecte pas l'univers de X car les questions de sensibilité qui passent par l'univers sont beaucoup plus délicates : elles sont l'objet d'une activité de recherche très intense et nous pouvons vous guider dans la littérature correspondante, certaines avancées ayant déjà un fort niveau d'opérationnalité. Mais pour une entrée dans les questions de sensibilités nous laissons de côté les situations où le paramètre modifie le domaine d'intégration.

Sans le dire, les sensibilités que nous regardions à travers la dérivée $\frac{\partial \mathcal{A}}{\partial \alpha}$ étaient des sensibilités au sens linéaire. La dérivée nous donne la réponse de \mathcal{A} à une petite perturbation du paramètre, petite (infinitésimale) au sens où la réponse de \mathcal{A} est linéaire, la dérivée $\frac{\partial \mathcal{A}}{\partial \alpha}$ portant alors toute l'information sur cette réponse.

Les Monte Carlo dits *symboliques* s'intéressent au contraire à des perturbations telles que la réponse de \mathcal{A} à une variation de α est non-linéaire. On cherche donc à caractériser la fonction $\mathcal{A}(\alpha)$ elle-même, toujours avec l'idée que α prend des valeurs dans le voisinage d'une valeur de référence α_{ref} mais en acceptant des écarts $\alpha - \alpha_{ref}$ non-infinitésimaux. L'adjectif *symbolique* est utilisé pour dire qu'une forme symbolique sera proposée pour $\mathcal{A}(\alpha)$ et que l'algorithme de Monte Carlo produira non pas une estimation de \mathcal{A} pour une valeur de α donnée, mais une estimation des paramètres de cette forme symbolique. Il y a plusieurs approches possibles et nous ne décrivons ici que celle qui a été à l'origine de cette approche, fondée sur l'échantillonnage par importance.

Quand nous avons introduit l'échantillonnage par importance c'était avec l'idée que l'on pouvait choisir les lois d'échantillonnage dans le but de réduire la variance de l'estimateur. Partant de l'équation 11, on pouvait changer $p_X(x; \alpha)$ en prenant une autre densité de probabilité $p_{X,opt}(x)$ et corriger le poids en le multipliant par le préfacteur $\frac{p_X(x; \alpha)}{p_{X,opt}(x)}$. Utilisons ici cette souplesse de la méthode, non pas dans le but de réduire la variance, mais dans le but de toujours utiliser α_{ref} pour l'échantillonnage et donc reporter la dépendance à α dans le poids seulement. On choisit donc $p_{X, "opt"}(x) = p_X(x; \alpha_{ref})$:

$$\mathcal{A}(\alpha) = \int_{\mathcal{U}_X} p_X(x; \alpha_{ref}) dx \frac{p_X(x; \alpha)}{p_X(x; \alpha_{ref})} \tilde{w}(x; \alpha) \quad (12)$$

Dans l'algorithme de Monte Carlo correspondra, on produira n réalisations x_i . L'estimation de $\mathcal{A}(\alpha)$ notée $m(\alpha)$ et l'estimation de la variance de l'estimateur de $\mathcal{A}(\alpha)$, notée $s(\alpha)^2$, seront alors

$$\begin{aligned} m(\alpha) &= \frac{1}{n} \sum_{i=1}^n \frac{p_X(x_i; \alpha)}{p_X(x_i; \alpha_{ref})} \tilde{w}(x_i; \alpha) \\ s(\alpha)^2 &= \frac{1}{n-1} \left(\left(\frac{1}{n} \sum_{i=1}^n \left[\frac{p_X(x_i; \alpha)}{p_X(x_i; \alpha_{ref})} \tilde{w}(x_i; \alpha) \right]^2 \right) - m(\alpha)^2 \right) \end{aligned} \quad (13)$$

Ces deux estimations sont bien des formes symboliques de fonctions de α . Il s'agit du premier niveau possible d'une approche symbolique : grâce à la technique de l'échantillonnage par "importance" on écrit un algorithme dans lequel les échantillonnages ne dépendent plus de α et on stocke toutes les réalisations x_i afin de pouvoir rejouer l'algorithme d'estimation pour toute nouvelle valeur de α sans avoir besoin de reconstruire ces réalisations. Cette façon de procéder est particulièrement intéressante quand c'est l'échantillonnage qui demande le plus de puissance de calcul. C'est souvent le cas. Dans les applications à l'ingénierie auxquelles nous pensons, X est souvent remplacé par un "chemin" visitant plusieurs physiques couplés impliquant des bases de données de grande dimension, le plus souvent dans des CAO de grande taille, est la puissance de calcul est avant tout demandée pour l'échantillonnage de ces chemins. Dans ce cas, pouvoir utiliser les mêmes chemins pour différentes valeurs de α (utiliser les formes symboliques de l'équation 13) est déjà utile sans aucun travail supplémentaire.

Mais le plus souvent, cette forme symbolique peut se simplifier en regroupant des termes. Admettons par exemple que \tilde{w} ne dépend pas de α , soit $\tilde{w} \equiv \tilde{w}(x)$, et que $p_X(x; \alpha)$ est de la forme suivante,

$$p_X(x; \alpha) = f(x) (\mathcal{H}(x)g_1(\alpha) + (1 - \mathcal{H}(x))g_2(\alpha)) \quad (14)$$

où la fonction Heaviside est telle que $\mathcal{H}(x) = 1$ si $x > 0$ et $\mathcal{H}(x) = 0$ sinon, les fonctions f , g_1 et g_2 étant connues. Dans ce cas, il est facile de regrouper les termes dans

l'équation 13 pour obtenir

$$m(\alpha) = m_1 \frac{g_1(\alpha)}{g_1(\alpha_{ref})} + m_2 \frac{g_2(\alpha)}{g_2(\alpha_{ref})}$$

$$s(\alpha)^2 = \frac{1}{n-1} \left(s_1 \frac{g_1(\alpha)^2}{g_1(\alpha_{ref})^2} + s_2 \frac{g_2(\alpha)^2}{g_2(\alpha_{ref})^2} - m(\alpha)^2 \right) ?? \quad (15)$$

avec

$$m_1 = \frac{1}{n} \sum_{i=1}^n \mathcal{H}(x_i) f(x_i) \tilde{w}(x_i)$$

$$m_2 = \frac{1}{n} \sum_{i=1}^n (1 - \mathcal{H}(x_i)) f(x_i) \tilde{w}(x_i)$$

$$s_1 = \frac{1}{n} \sum_{i=1}^n \mathcal{H}(x_i) f(x_i)^2 \tilde{w}(x_i)^2$$

$$s_2 = \frac{1}{n} \sum_{i=1}^n (1 - \mathcal{H}(x_i)) f(x_i)^2 \tilde{w}(x_i)^2 \quad (16)$$

Dans un tel cas, la sortie de l'algorithme de Monte Carlo est cet ensemble de quatre valeurs, m_1 , m_2 , s_1 et s_2 qui sont les paramètres de la forme symbolique de l'estimateur de $\mathcal{A}(\alpha)$, soit $m(\alpha)$ dans l'équation ?? et les paramètres de la forme symbolique de la variance de cet estimateur, soit $s(\alpha)^2$ dans l'équation ??.

Il est important de retenir que non seulement on dispose d'un estimateur de $\mathcal{A}(\alpha)$ qui nous donne la fonction de α elle-même (analyser sa forme étant souvent riche en termes d'analyse physique), mais aussi le niveau d'incertitude, qui est lui aussi une fonction de α . Souvent l'incertitude s'avère plus forte quand α s'éloigne de α_{ref} et il faut parfois augmenter la taille de l'échantillon pour saisir la non-linéarité de la réponse sur une plage de variation élevée, auquel cas plusieurs calculs simulations de Monte Carlo peuvent être nécessaires, pour différentes valeurs de α_{ref} , par exemple équi-réparties dans la plage d'intérêt de α .

Non linéaire

Lorsque nous avons parlé de double randomisation, comme un moyen de combiner les espérances de fonctions aléatoires imbriquées (jusqu'à la dimension infinie avec la récursivité), il s'agissait toujours de fonction aléatoire dans lesquelles une espérance apparaissaient de façon linéaire. Et effectivement, la double randomisation est une idée profondément associée au linéaire, au point que la plupart des écrits théoriques sur la méthode de Monte Carlo affirme l'impossibilité d'une extension au monde du non-linéaire, d'où les idées de Monte Carlo en essaim où on accepte le sacrifice de certaines des forces essentielles de la méthode dans la visée du non-linéaire. Cependant, d'autres idées sont apparues qui préservent l'intégralité du cadre théorique. Nous allons ici décrire l'idée principale, le reste étant soit resté à des modèles physiques très spécifiques, par exemple la physique cinétique, soit trop embryonnaire, par exemple des résonances récentes avec la transformée de Laplace.

L'idée est présente à la fois dans la littérature de mathématique appliquée et dans celle de l'ingénierie du génie des procédés. En partant toujours du même type de formulation intégrale correspondant à un algorithme de Monte Carlo, on considère une fonction poids $\tilde{w}_{\mathcal{A}}$ qui dépend d'un paramètre \mathcal{C} , soit

$$\mathcal{A} = \int_{\mathcal{U}_X} p_X(x) dx \tilde{w}_{\mathcal{A}}(x; \mathcal{C}) \quad (17)$$

le paramètre \mathcal{C} n'étant connu que sous la forme d'une écriture intégrale, correspondant elle-même à un algorithme de Monte Carlo, soit

$$\mathcal{C} = \int_{\mathcal{U}_Y} p_Y(y) dy \tilde{w}_{\mathcal{C}}(y) \quad (18)$$

En termes d'espérance, ces deux écritures signifient

$$\begin{aligned} \mathcal{A} &= \mathbb{E}(\tilde{w}_{\mathcal{A}}(X; \mathcal{C})) \\ \mathcal{C} &= \mathbb{E}(\tilde{w}_{\mathcal{C}}(Y)) \end{aligned} \quad (19)$$

et si la dépendance de $\tilde{w}_{\mathcal{A}}$ à \mathcal{C} on peut regrouper les deux espérances en se servant du fait que l'espérance d'une fonction linéaire d'une espérance est une espérance :

$$[\text{cas linéaire}] \quad \mathcal{A} = \mathbb{E}(\tilde{w}_{\mathcal{A}}(X; \tilde{w}_{\mathcal{C}}(Y))) \quad (20)$$

Ainsi il n'est plus nécessaire de commencer par estimer \mathcal{C} à l'aide de son algorithme de Monte Carlo avant d'utiliser l'algorithme de \mathcal{A} , en remplaçant alors \mathcal{C} par son estimation (ce qui de toute façon introduirait un biais). On peut directement utiliser un nouvel algorithme qui alie les deux précédents, où chaque poids w est construit à l'aide d'une réalisation x de X et d'une réalisation y de Y en combinant les deux fonctions poids : $w = \tilde{w}_{\mathcal{A}}(x; \tilde{w}_{\mathcal{C}}(y))$.

Mais ceci ne vaut que sous hypothèse de linéarité. Si $\tilde{w}_{\mathcal{A}}$ dépend non-linéairement de \mathcal{C} ce n'est pas possible. Par contre, nous pouvons considérer le cas particulier où la non-linéarité peut être développée en série entière. On admet donc maintenant que $\tilde{w}_{\mathcal{A}}(x; \mathcal{C})$ peut s'écrire

$$\tilde{w}_{\mathcal{A}}(x; \mathcal{C}) = \sum_{i=0}^{+\infty} a_i(x) (\mathcal{C} - \mathcal{C}_{ref})^i \quad (21)$$

où \mathcal{C}_{ref} est une grandeur connue et les a_i sont des fonctions de x connues. Une stratégie possible est alors la suivante :

- probabiliser les sommes successive de la série ;
- aborder la puissance $(\mathcal{C} - \mathcal{C}_{ref})^i$ comme un produit à i termes et dans chaque terme j remplacer \mathcal{C} par l'espérance $\mathbb{E}(\tilde{w}_{\mathcal{C}}(Y_j))$ où les Y_j sont des variables aléatoires IID de même loi que Y .

Probabiliser la série On introduit un jeu de probabilités $P_0, P_1, P_2 \dots$ tel que $\sum_{i=0}^{+\infty} P_i = 1$. Cela définit une variable aléatoire I telle que $I = 0$ avec la probabilité P_0 , $I = 1$ avec la probabilité P_1 , etc. On peut alors écrire

$$\tilde{w}_{\mathcal{A}}(x; \mathcal{C}) = \sum_{i=0}^{+\infty} P_i \left[\frac{a_i(x)}{P_i} (\mathcal{C} - \mathcal{C}_{ref})^i \right] \quad (22)$$

ce qui s'interprète comme un algorithme de Monte Carlo où on crée une réalisation i de I et le poids de Monte Carlo est $w = \frac{a_i(x)}{P_i} (\mathcal{C} - \mathcal{C}_{ref})^i$. En termes d'espérance,

$$\tilde{w}_{\mathcal{A}}(x; \mathcal{C}) = \mathbb{E} \left(\frac{a_I(x)}{P_I} (\mathcal{C} - \mathcal{C}_{ref})^I \right) \quad (23)$$

Déployer la puissance Le développement en série entière résume toute la non-linéarité en une puissance $(\mathcal{C} - \mathcal{C}_{ref})^i$. L'équation 24 nous permet d'écrire \mathcal{C} sous forme intégrale et comme p_Y est une densité de probabilité sur \mathcal{U}_Y , nous savons aussi que $\int_{\mathcal{U}_Y} p_Y(y) dy = 1$, donc

$$\mathcal{C}_{ref} = \int_{\mathcal{U}_Y} p_Y(y) dy \mathcal{C}_{ref} \quad (24)$$

Nous pouvons alors écrire

$$\mathcal{C} - \mathcal{C}_{ref} = \int_{\mathcal{U}_Y} p_Y(y) dy (\tilde{w}_{\mathcal{C}}(y) - \mathcal{C}_{ref}) \quad (25)$$

soit

$$\mathcal{C} - \mathcal{C}_{ref} = \mathbb{E}(\tilde{w}_{\mathcal{C}}(Y) - \mathcal{C}_{ref}) \quad (26)$$

L'étape importante à ce stade consiste à définir une suite Y_1, Y_2, Y_3, \dots de variables aléatoires IID de même loi que Y . Pour tout i on peut alors écrire

$$\begin{aligned} (\mathcal{C} - \mathcal{C}_{ref})^i &= \left(\int_{\mathcal{U}_Y} p_Y(y_1) dy_1 (\tilde{w}_{\mathcal{C}}(y_1) - \mathcal{C}_{ref}) \right) \left(\int_{\mathcal{U}_Y} p_Y(y_2) dy_2 (\tilde{w}_{\mathcal{C}}(y_2) - \mathcal{C}_{ref}) \right) \dots \left(\int_{\mathcal{U}_Y} p_Y(y_i) dy_i (\tilde{w}_{\mathcal{C}}(y_i) - \mathcal{C}_{ref}) \right) \\ &= \int_{\mathcal{U}_Y} p_Y(y_1) dy_1 \int_{\mathcal{U}_Y} p_Y(y_2) dy_2 \dots \int_{\mathcal{U}_Y} p_Y(y_i) dy_i (\tilde{w}_{\mathcal{C}}(y_1) - \mathcal{C}_{ref}) (\tilde{w}_{\mathcal{C}}(y_2) - \mathcal{C}_{ref}) \dots (\tilde{w}_{\mathcal{C}}(y_i) - \mathcal{C}_{ref}) \end{aligned} \quad (27)$$

soit

$$(\mathcal{C} - \mathcal{C}_{ref})^i = \mathbb{E} \left(\prod_{j=1}^i (\tilde{w}_{\mathcal{C}}(Y_j) - \mathcal{C}_{ref}) \right) \quad (28)$$

La non-linéarité est ainsi ramenée à une unique espérance.

Reporter la puissance déployée dans la série probabilisée On peut alors reporter cette unique espérance dans l'équation 23 et reporter le tout dans l'expression de \mathcal{A} dans l'équation 19, en utilisant la double randomisation à chacune de ces deux étapes, pour obtenir une écriture qui est équivalente à celle de l'équation 20, mais qui vaut maintenant en non-linéaire, soit

$$\begin{aligned} \tilde{w}_{\mathcal{A}}(x; \mathcal{C}) &= \mathbb{E} \left(\frac{a_I(x)}{P_I} \mathbb{E} \left(\prod_{j=1}^I (\tilde{w}_{\mathcal{C}}(Y_j) - \mathcal{C}_{ref}) \right) \right) \\ &= \mathbb{E} \left(\frac{a_I(x)}{P_I} \prod_{j=1}^I (\tilde{w}_{\mathcal{C}}(Y_j) - \mathcal{C}_{ref}) \right) \end{aligned} \quad (29)$$

et

$$\mathcal{A} = \mathbb{E} \left(\mathbb{E} \left(\frac{a_I(X)}{P_I} \prod_{j=1}^I (\tilde{w}_{\mathcal{C}}(Y_j) - \mathcal{C}_{ref}) \right) \right) \quad (30)$$

donc

$$[\text{cas non-linéaire}] \quad \mathcal{A} = \mathbb{E} \left(\frac{a_I(X)}{P_I} \prod_{j=1}^I (\tilde{w}_{\mathcal{C}}(Y_j) - \mathcal{C}_{ref}) \right) \quad (31)$$

L'algorithme correspondant consiste à créer une réalisation x de X , puis créer une réalisation i de I (l'ordre de la série), et ensuite créer i réalisations $y_1, y_2 \dots y_i$ des variables aléatoires parentes $Y_1, Y_2 \dots Y_i$, le poids de Monte Carlo étant $w = \frac{a_i(x)}{P_i} \prod_{j=1}^i (\tilde{w}_{\mathcal{C}}(Y_j) - \mathcal{C}_{ref})$.

TP

TP7

Estimer l'intégrale suivante :

$$\mathcal{A}(p, q, \gamma) = \int_0^1 dx \left(\gamma x \mathcal{A}(p, q, \gamma) + (1 - \gamma x) \int_0^x dy x^p y^q \right) \quad (32)$$

où $\gamma \in [0, 1]$ ce qui permet d'interpréter le produit γx comme une probabilité et avancer vers un algorithme qui implique une variable aléatoire de Bernoulli B de probabilité $P = \gamma x$. Le succès ou l'échec du test de Bernoulli conduit soit à une récursivité (on doit à nouveau estimer \mathcal{A}), soit à interruption de la récursion (avec l'échantillonnage d'un y sur $[0, x]$ selon une loi à choisir).

Solution du TP7 Un code correspondant au TP7 est proposé dans le répertoire `/home/lafrier/livre_informatique_graphique/Codes/somme_xp_yq_recurusif`.

TP8

Modifier l'algorithme du TP7 de façon à estimer simultanément la sensibilité de \mathcal{A} à γ .

Solution du TP8 Un code correspondant au TP8 est proposé dans le répertoire `/home/lafrier/livre_informatique_graphique/Codes/sensib_somme_xp_yq_recurusif`.

TP9

Concevoir un algorithme de Monte Carlo symbolique pour la dépendance non-linéaire de \mathcal{A} à γ .

TP10

Reprendre le TP7 en modifiant la définition de γ qui devient maintenant une fonction de x :

$$\gamma(x) = \int_0^x z^k dz \quad (33)$$

On fera le choix de ne pas précalculer cette intégrale. Au contraire, on commencera par concevoir un algorithme de Monte Carlo estimant $\gamma(x)$ pour une valeur de x donnée. L'objectif sera alors de combiner cet algorithme avec l'algorithme du TP7, sachant que la combinaison des deux espérances correspondantes est non-linéaire.

Solution du TP10 Un code correspondant au TP10 est proposé dans le répertoire `/home/lafrier/livre_informatique_graphique/Codes/somme_xp_yq_recurusif_non_lineair`.

Index