

SIO - Simulation et optimisation
Simulation et générateurs

15 septembre 2025

Table des matières

1	Simulation	1
1.1	Pourquoi simuler un système?	2
1.2	Génération de nombres aléatoires	2
2	Génération de variables aléatoires	2
2.1	Méthode des fonctions inverses	3
2.1.1	Utilisation variables uniformes	3
2.1.2	Utilisation variables exponentielles	3
2.2	Variables aléatoires discrètes	4
2.2.1	Utilisation	4
2.2.2	Variable de Bernoulli $X \sim B(p)$	4
2.2.3	Variable binomiale $X \sim B(n, p)$	4
2.2.4	Variable de Poisson $X \sim P(\lambda)$	4
2.3	Méthodes des mélanges	5
2.4	Méthode des convolutions	5
2.5	Méthode des rejets	5
3	Méthode de Monte-Carlo	6
3.1	Estimateur classique	7
3.2	Estimateur de la variance	7
3.3	Précision de l'estimateur	7
3.4	Intervalle de confiance	7
3.5	Méthode d'acceptation-rejet	8
3.6	Echantillonnage uniforme	9
3.7	Echantillonnage préférentiel	10

1 Simulation

1.1 Pourquoi simuler un système?

La simulation permet de modéliser le comportement d'un système réel, souvent complexe. Cependant, elle présente plusieurs difficultés:

- le choix, la conception et la validation d'un modèle sont autant d'étapes délicates
- la collecte et l'analyse des données est une phase cruciale pour obtenir des résultats utilisables
- la mise en œuvre informatique peu demander passablement de temps
- le risque de bugs est toujours présent
- l'analyse statistique des résultats doit être faite avec rigueur
- la convergence est parfois (voire souvent) lente

i Info

Pour obtenir des résultats corrects et suffisamment précis, la simulation est **gourmande en temps** de conception, de développement, de validation et de calcul.

1.2 Génération de nombres aléatoires

Un ordinateur étant déterministe, il ne peut pas générer de nombres aléatoires. On utilise donc des **pseudo-nombres aléatoires** (PRNG: Pseudo-Random Number Generator) qui sont générés par des algorithmes à partir d'une **graine** (seed). Si on utilise la même graine, on obtient la même séquence de nombres pseudo-aléatoires.

2 Génération de variables aléatoires

La génération de variables aléatoires suivant une distribution donnée à pour objectif de représenter des phénomènes aléatoires réels dans une simulation. Plusieurs méthodes existent pour générer des variables aléatoires suivant une distribution spécifique.

i Exemple

Imaginons que nous souhaitons simuler la taille des individus dans une population. Si nous savons que la taille suit une distribution normale avec une moyenne de 170 cm et un écart-type de 10 cm, nous souhaiterons donc générer des variables aléatoires suivant cette distribution normale pour représenter les tailles des individus dans notre simulation.

2.1 Méthode des fonctions inverses

La méthode des fonctions inverses consiste à utiliser la fonction de répartition inverse (quantile) pour générer des variables aléatoires suivant une distribution donnée. On génère d'abord un nombre aléatoire uniforme U dans l'intervalle $[0, 1]$, puis on applique la fonction de répartition inverse $F^{-1}(-1)$ pour obtenir la variable aléatoire $X = F^{-1}(U)$.

Cette approche est particulièrement efficace lorsque la fonction de répartition inverse est facile à calculer.

2.1.1 Utilisation variables uniformes

On génère un nombre aléatoire uniforme u dans $[0, 1]$.

On vient trouver la valeur:

$$x = F^{-1}(u) = a + u(b - a)$$

où F est la fonction de répartition de la distribution cible, et a et b sont les bornes de l'intervalle.

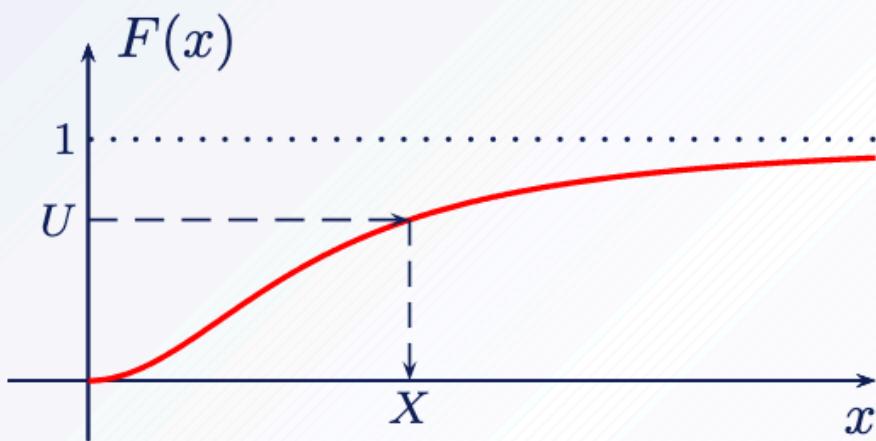


Fig. 1. – Capture des slides du cours – Méthode des fonctions inverses

2.1.2 Utilisation variables exponentielles

On génère un nombre aléatoire uniforme u dans $[0, 1]$.

On vient trouver la valeur:

$$x = -\frac{1}{\lambda} \ln(1 - u)$$

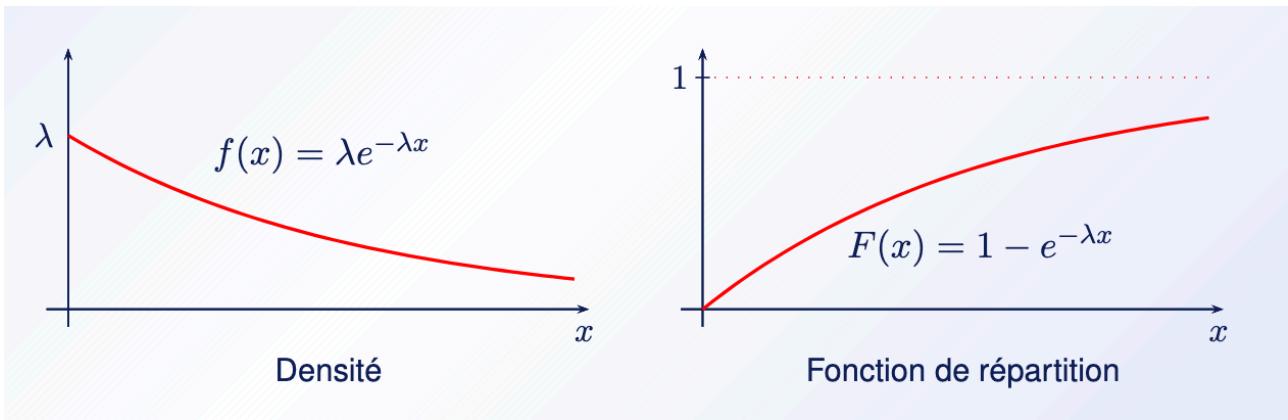


Fig. 2. – Capture des slides du cours – Méthode des fonctions inverses pour une variable exponentielle

2.2 Variables aléatoires discrètes

Une variable aléatoire discrète prend un nombre fini ou dénombrable de valeurs. Pour générer une variable aléatoire discrète suivant une distribution donnée, on peut utiliser la méthode des fonctions inverses adaptée aux variables généralisées en définissant

$$F^{-1}(u) = \min\{x \mid F(x) \geq u\}$$

2.2.1 Utilisation

On génère une réalisation u d'une variable uniforme $U(0, 1)$.

Déterminer k tel que

$$\sum_{i=1}^{k-1} p_i < u \leq \sum_{i=1}^k p_i$$

et retourner $x = x_k$.

2.2.2 Variable de Bernoulli $X \sim B(p)$

Une variable de Bernoulli peut prendre deux valeurs possibles

$$P(\text{échec}) = P(X = 0) = 1 - p$$

$$P(\text{succès}) = P(X = 1) = p$$

Algorithme:

On génère une réalisation u d'une variable uniforme $U(0, 1)$.

- Si $u < p$, retourner $X = 1$ (succès)
- Sinon, retourner $X = 0$ (échec)

2.2.3 Variable binomiale $X \sim B(n, p)$

Une variable binomiale représente le nombre de succès dans n essais indépendants de Bernoulli.

Algorithme:

Initialiser $X = 0$.

Répéter n fois:

- Générer une réalisation u d'une variable uniforme $U(0, 1)$.
- Si $u < p$, incrémenter X de 1.

Retourner X .

2.2.4 Variable de Poisson $X \sim P(\lambda)$

Une variable de Poisson modélise le nombre d'événements se produisant dans un intervalle de temps donné, avec un taux moyen λ .

Algorithme:

Générer des réalisations indépendante (y_0, y_1, \dots) d'une loi $E(1)$ jusqu'à la première valeur k telle que

$$\sum_{i=0}^k y_i > \lambda$$

Retourner $X = k$.

2.3 Méthodes des mélanges

La méthode des mélanges s'applique quand une distribution compliquée est en réalité un mélange de plusieurs distributions plus simples. Mathématiquement, la fonction de répartition $F(x)$ peut s'écrire comme une combinaison pondérée de n fonctions de répartition $F_1(x), F_2(x), \dots, F_{n(x)}$:

$$F(x) = \sum_{i=1}^n p_i F_{i(x)} \quad \text{avec} \quad \sum_{i=1}^n p_i = 1 \text{ et } p_i \geq 0 \forall i$$

où les p_i sont les poids (probabilités) de chaque distribution simple, et F_i est la fonction de répartition de la i -ème distribution simple.

Lorsqu'il est facile de générer des réalisations selon chacune des distributions simples F_i , on peut générer des réalisations selon la distribution compliquée en deux étapes :

1. Générer un indice k selon la loi discrète $P(K = i) = p_i$ (électionner laquelle des n distributions utiliser)
2. Générer une réalisation x selon la fonction de répartition F_k (générer selon la distribution choisie)

Retourner x . En répétant ce processus, les valeurs générées suivront naturellement le mélange pondéré, donc la distribution cible.

2.4 Méthode des convolutions

La méthode des convolutions s'applique quand une variable aléatoire X peut être exprimée comme la somme de plusieurs variables aléatoires indépendantes plus simples. Si $X = X_1 + X_2 + \dots + X_n$ et qu'il est facile de générer des réalisations selon chacune des variables X_i , alors générer selon la loi de X devient facile.

L'intuition est simple : au lieu de chercher une formule complexe pour X , on génère chacune de ses composantes et on les additionne.

Algorithme:

1. Pour i de 1 à n , générer une réalisation x_i selon la loi de X_i
2. Retourner la valeur $x = x_1 + x_2 + \dots + x_n$

i Exemple

Une variable triangulaire symétrique $X \sim \mathcal{T}(0, 1, 2)$ peut être exprimée comme la somme de deux variables uniformes indépendantes : $X = U_1 + U_2$ où $U_1, U_2 \sim \mathcal{U}(0, 1)$. Pour générer une réalisation, il suffit de générer deux uniformes et les additionner.

2.5 Méthode des rejets

La méthode des rejets s'applique quand on désire générer des réalisations d'une variable aléatoire X de densité $f(x)$ complexe sans connaître de méthode directe. L'idée consiste à utiliser une distribution auxiliaire plus simple $g(x)$ dont on sait générer des réalisations, ainsi qu'une constante $M \geq 1$ telle que $f(x) \leq M \cdot g(x)$ pour tout x .

Graphiquement, la courbe $M \cdot g(x)$ forme une enveloppe qui recouvre complètement la courbe $f(x)$.

Algorithme:

1. Générer une réalisation z selon la densité $g(x)$ (position sur l'axe x)
2. Générer une réalisation u selon la loi $U(0, 1)$ (hauteur sur l'axe y)
3. Si $u \leq \frac{f(z)}{M \cdot g(z)}$, accepter la valeur z et retourner $x = z$
4. Sinon, retourner à l'étape 1.

Les valeurs acceptées correspondent à des points situés sous la courbe $f(x)$. En répétant ce processus, les points acceptés suivent naturellement la distribution de densité $f(x)$.

i Performance

La probabilité d'accepter une réalisation est égale à $\frac{1}{M}$. En moyenne, il faut donc M itérations pour générer une seule valeur. Les performances s'améliorent en choisissant une densité $g(x)$ qui épouse au mieux $f(x)$, réduisant ainsi la valeur de M .

3 Méthode de Monte-Carlo

La méthode de Monte-Carlo consiste à générer un grand nombre de réalisations indépendantes d'une variable aléatoire pour estimer des caractéristiques qu'on ne sait pas calculer directement. En répétant l'expérience avec suffisamment de points, les estimations empiriques convergent vers les vraies valeurs.

3.1 Estimateur classique

Pour estimer l'espérance $\mu = E(X)$ d'une variable aléatoire X , on génère n réalisations indépendantes (x_1, x_2, \dots, x_n) et on calcule l'estimateur empirique :

$$\hat{\mu} = \bar{x} = \frac{1}{n} \sum_{k=1}^n x_k$$

Par le théorème central limite, cet estimateur converge vers la vraie espérance μ quand n devient grand.

3.2 Estimateur de la variance

De la même manière, on estime la variance $\sigma^2 = \text{Var}(X)$ par :

$$\hat{\sigma}^2 = s^2 = \frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})^2$$

où \bar{x} est l'estimateur de l'espérance. On divise par $n-1$ plutôt que n pour obtenir un estimateur sans biais.

3.3 Précision de l'estimateur

La largeur de l'intervalle de confiance est :

$$\Delta I_c = 2z_{1-\frac{\alpha}{2}} \frac{s}{\sqrt{n}}$$

La convergence se fait en $O(n^{-\frac{1}{2}}) = O(\frac{1}{\sqrt{n}})$. Cela signifie que pour diviser par 2 la largeur de l'intervalle, il faut générer **4 fois plus** de réalisations, et pour diviser par 10, il faut un échantillon **100 fois plus grand**.

i Info

La convergence des méthodes de Monte-Carlo est lente. Obtenir une précision supplémentaire demande beaucoup plus de calculs.

3.4 Intervalle de confiance

Avec un seuil de confiance $1-\alpha$, l'espérance $\mu = E(X)$ se trouve dans l'intervalle de confiance bilatéral :

$$I_c = \left[\bar{x} - z_{1-\frac{\alpha}{2}} \frac{s}{\sqrt{n}}, \bar{x} + z_{1-\frac{\alpha}{2}} \frac{s}{\sqrt{n}} \right]$$

où $z_{1-\frac{\alpha}{2}}$ est le quantile de la loi normale correspondant au seuil de confiance. Les seuils les plus courants sont :

- 95 % de confiance : $z_{1-\frac{\alpha}{2}} = 1.960$
- 99 % de confiance : $z_{1-\frac{\alpha}{2}} = 2.576$

En pratique, on doit avoir au moins $n \geq 30$ observations pour que le théorème central limite s'applique correctement.

3.5 Méthode d'acceptation-rejet

La méthode d'acceptation-rejet, aussi appelée « hit or miss », est une technique de Monte-Carlo utilisée pour estimer des intégrales ou des espérances lorsque la fonction à intégrer est complexe ou difficile à manipuler analytiquement.

Algorithme:

- 1) Poser $S = 0$ // compte le nombre de succès
- 2) Pour k de 1 à N faire
- 3) Générer $X_k \sim \mathcal{U}(a, b)$ et $U_k \sim \mathcal{U}(0, M)$
- 4) Si $U_k \leq g(X_k)$ poser $Y_k = 1$ // succès (*Hit*)
- 5) Sinon poser $Y_k = 0$ // échec (*Miss*)
- 6) Poser $S = S + Y_k$
- 7) Calculer $\bar{Y} = \frac{S}{N}$ et $\hat{\sigma}_Y^2 = \bar{Y}(1 - \bar{Y})$
- 8) Calculer $\hat{G} = M(b - a)\bar{Y}$ // estimateur de $G = \int_a^b g(x)dx$
- 9) et $\hat{\sigma}_{\hat{G}} = M(b - a)\sqrt{\frac{\hat{\sigma}_Y^2}{N}}$ // estimateur de l'écart-type de \hat{G}
- 10) Retourner \hat{G} et l'intervalle de confiance au seuil $1 - \alpha$: $\left[\hat{G} \pm z_{1-\frac{\alpha}{2}} \hat{\sigma}_{\hat{G}} \right]$

Fig. 3. – Capture des slides du cours – Méthode d'acceptation-rejet

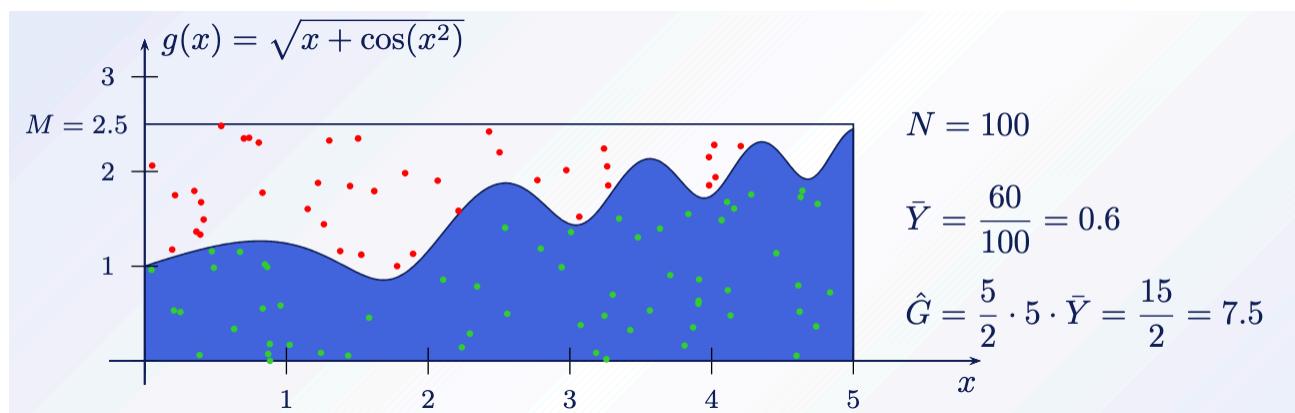


Fig. 4. – Capture des slides du cours – Méthode d'acceptation-rejet

3.6 Echantillonnage uniforme

- 1) Poser $S = 0$ et $Q = 0$ // sommes des Y_k et de leurs carrés
- 2) Pour k de 1 à N faire
- 3) Générer $X_k \sim \mathcal{U}(a, b)$
- 4) Calculer $Y_k = g(X_k)$
- 5) Poser $S = S + Y_k$ et $Q = Q + Y_k^2$
- 6) Calculer $\bar{Y} = \frac{S}{N}$ et $\hat{\sigma}_Y^2 = \frac{Q}{N} - \bar{Y}^2$
- 7) Calculer $\hat{G} = (b - a)\bar{Y}$ // estimateur de $G = \int_a^b g(x)dx$
- 8) et $\hat{\sigma}_{\hat{G}} = (b - a)\sqrt{\frac{\hat{\sigma}_Y^2}{N}}$ // estimateur de l'écart-type de \hat{G}
- 9) Retourner \hat{G} et l'intervalle de confiance au seuil $1 - \alpha$: $\left[\hat{G} \pm z_{1-\frac{\alpha}{2}} \hat{\sigma}_{\hat{G}} \right]$

Fig. 5. – Capture des slides du cours – Echantillonnage uniforme

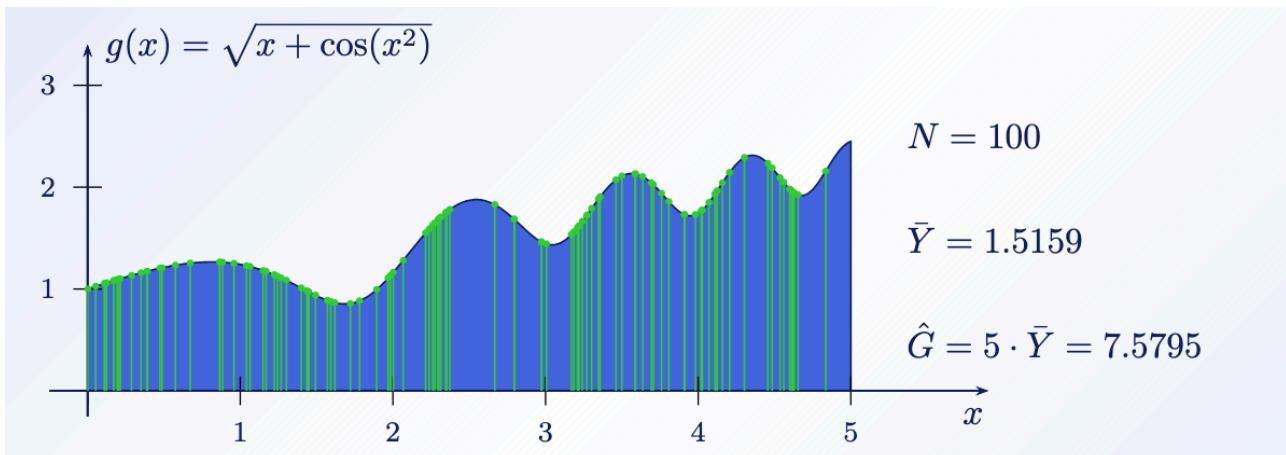


Fig. 6. – Capture des slides du cours – Echantillonnage uniforme

3.7 Echantillonnage préférentiel

Dans l'échantillonnage préférentiel, on choisit de générer des points dans des régions où cela favorise les points contribuant le plus à la valeur de G.

- 1) Poser $S = 0$ et $Q = 0$ // sommes des Y_k et de leurs carrés
- 2) Pour k de 1 à N faire
- 3) Générer X_k selon la densité $f(x)$
- 4) Calculer $Y_k = \frac{g(X_k)}{f(X_k)}$
- 5) Poser $S = S + Y_k$ et $Q = Q + Y_k^2$
- 6) Calculer $\bar{Y} = \frac{S}{N}$ et $\hat{\sigma}_Y^2 = \frac{Q}{N} - \bar{Y}^2$
- 7) Calculer $\hat{G} = \bar{Y}$ // estimateur de $G = \int_a^b g(x)dx$
- 8) et $\hat{\sigma}_{\hat{G}} = \sqrt{\frac{\hat{\sigma}_Y^2}{N}}$ // estimateur de l'écart-type de \hat{G}
- 9) Retourner \hat{G} et l'intervalle de confiance au seuil $1 - \alpha$: $\left[\hat{G} \pm z_{1-\frac{\alpha}{2}} \hat{\sigma}_{\hat{G}} \right]$

Fig. 7. – Capture des slides du cours – Echantillonnage préférentiel

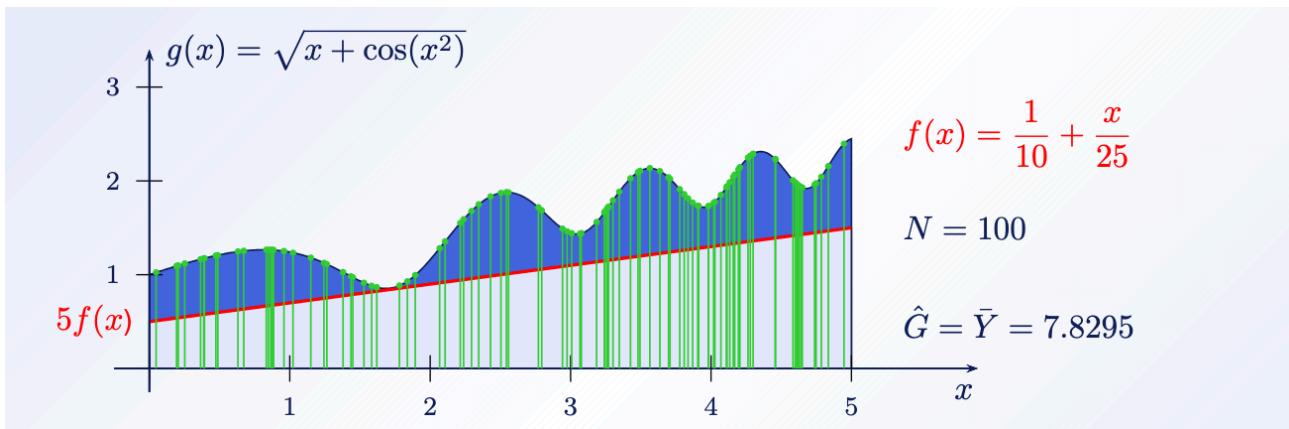


Fig. 8. – Capture des slides du cours – Echantillonnage préférentiel