

# Simulation et optimisation

Jean-François HÊCHE

Printemps 2018



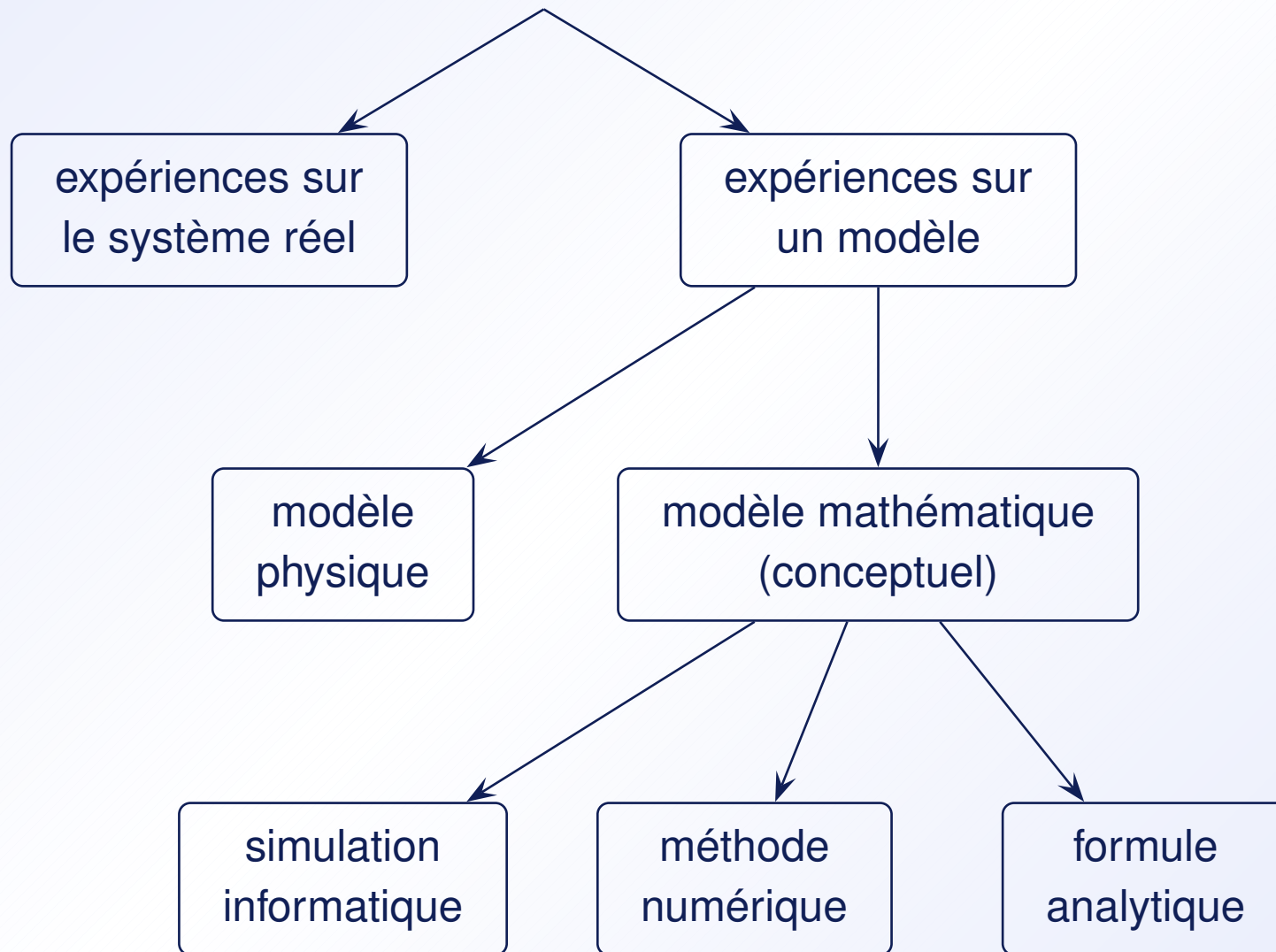
# La simulation informatique



La **modélisation** et la **simulation** sont des approches scientifiques courantes pour répondre à certaines **questions** concernant un **système**.

Elles constituent l'un des outils d'aide à la décision les plus efficaces pour la conception et la gestion des systèmes complexes.

- Un **modèle** est une description **simplifiée** d'un système, construit dans le but d'évaluer ses performances ou d'analyser, voire d'optimiser, son comportement sous l'effet de certaines actions ou décisions.
- **Simuler** c'est faire évoluer le modèle d'un système, typiquement en déroulant l'historique de son fonctionnement, en lui fournissant les données nécessaires et en observant et collectant les résultats de cette évolution.



# Pourquoi modéliser et simuler ?

Une question légitime se pose :

Pourquoi utiliser un modèle plutôt que de faire des expériences sur le système réel ?

Parce qu'il n'est pratiquement jamais possible ou envisageable de travailler directement avec le système réel car

- Le système n'existe pas ou pas encore ...      dimensionnement d'une chaîne de production
- Trop lent ou trop rapide ...      réaction chimique, usure
- Trop cher ...      construction d'un prototype
- Trop perturbant ...      redimensionnement d'un réseau
- Trop risqué ...      fiabilité d'une centrale nucléaire
- ...

La simulation permet d'expérimenter avec un système sans payer le vrai prix de nos erreurs.

La simulation est une approche très **générale** et **populaire**, mais elle présente certaines **difficultés** qu'il ne faut pas sous-estimer ni négliger :

- 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 peut 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.

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

## Un modèle est une abstraction de la réalité.

- Un modèle ne capture que quelques aspects d'un système.
- Un modèle n'est généralement efficace que pour l'étude de certaines propriétés d'un système et le choix du modèle ou d'une méthode doit prendre en compte la question que l'on se pose sur le système.
- Un bon modèle n'est pas forcément le plus détaillé.

Idéalement. . . un modèle fournit les mêmes sorties (*output*) que le système réel lorsqu'on l'alimente avec les mêmes données (*input*).

Attention. . . un tel principe appliqué trop strictement pousse à concevoir des modèles trop complexes.

**Parmi les modèles suffisamment réalistes, le plus simple est souvent le meilleur.**

## Simulation

Modélisation et simulation

Pourquoi modéliser et simuler ?

Modèle et système réel

## Classification

Générateur aléatoire

Génération de V.A.

Méthodes de Monte-Carlo

La simulation peut également être utile pour

- **former** ou **entraîner** (simulateur de vol, *serious games*, ...)
- **divertir** (jeux vidéo)
- appréhender des systèmes ou des domaines nouveaux ou mal compris (milieux granulaires, marchés économiques, ...), les tentatives de reproduire en simulation certains phénomènes pouvant amener à une meilleure compréhension des mécanismes en jeu
- ...

Un modèle de simulation peut être

- Statique ou **Dynamique**
- Déterministe ou **Stochastique**
- Discret ou **Continu**
- ...



# Génération de nombres aléatoires



# Comment alimenter un modèle ?

Typiquement, la simulation d'un système dynamique nécessite de disposer de valeurs décrivant le comportement (durée, fréquence, ...) des différents éléments en interaction.

- Lorsqu'il est possible de rassembler des **données réelles** (historiques des ventes, ...) leur utilisation (telles quelles ou randomisées) fournit des données très réalistes.

Une telle approche n'est cependant pas toujours possible (coût trop élevé d'accessibilité ou de récolte, données inexistantes).

- En l'absence de données réelles, il est nécessaire de **modéliser** les éléments interagissant dans le système afin de **générer** les valeurs nécessaires à la simulation.

Le choix et la validation de ces modèles statistiques requièrent soin et dialogues. Leur exploitation nécessite quant à elle

- ▶ une **source de hasard**, c'est-à-dire un générateur de nombres aléatoires ;
- ▶ des méthodes adéquates de **transformation** de ces nombres (afin d'obtenir des réalisations de variables aléatoires conformes aux lois souhaitées).

## Simulation

### Générateur aléatoire

#### Source de hasard

Générateurs aléatoire et pseudo-aléatoire

Critères de qualité

Méthode des congruences linéaires

Générateurs basés sur des récurrences sur  $\mathbb{Z}_2$

### Génération de V.A.

### Méthodes de Monte-Carlo

- La simulation stochastique ainsi que de nombreuses méthodes dans des domaines d'applications très divers (physique, finances, jeux, cryptologie, ...) ont besoin d'avoir accès à des réalisations de variables aléatoires suivant des lois aussi diverses que variées.
- En pratique ces valeurs sont obtenues
  - ▶ à partir de réalisations de variables uniformément distribuées dans l'intervalle  $(0, 1)$  (loi uniforme  $\mathcal{U}(0, 1)$ ),
  - ▶ ou à partir de suites de *bits* aléatoires (loi uniforme sur l'ensemble  $\{0, 1\}$ ).
- Idéalement de nombreux programmes ont donc besoin d'avoir accès à un **générateur aléatoire**, c'est-à-dire à un dispositif capable de générer une suite infinie de réalisations de **variables indépendantes et identiquement distribuées (i.i.d.)** selon une loi uniforme  $\mathcal{U}(0, 1)$ .

# Générateurs aléatoire et pseudo-aléatoire

## Simulation

### Générateur aléatoire

Source de hasard

**Générateurs aléatoire  
et pseudo-aléatoire**

Critères de qualité

Méthode des  
congruences linéaires

Générateurs basés sur  
des récurrences sur  $\mathbb{Z}_2$

### Génération de V.A.

### Méthodes de Monte-Carlo

- Les ordinateurs sont des machines déterministes ! Elles ne peuvent donc pas produire de suites de nombres « vraiment » aléatoires.
- Pour obtenir des nombres « vraiment » aléatoires il faut avoir recours à des appareillages externes qui se basent sur des processus physiques chaotiques (*Lavarand*), des phénomènes quantiques ou radioactifs (*Hotbits*) ou encore des bruits thermiques ou électromagnétiques (*Random.org*).
- Mis à part certaines applications sensibles (cryptologie, loteries, ...) on préfère utiliser des programmes générant des suites **déterministes** mais ressemblant à s'y méprendre à des suites de réalisations de variables aléatoires i.i.d. d'une loi  $\mathcal{U}(0, 1)$ .
- Techniquement, on parle alors de **générateurs pseudo-aléatoires** et de suites de **nombres pseudo-aléatoires**. Il est cependant courant en simulation et en algorithmique de laisser tomber le préfixe *pseudo*.

## Simulation

### Générateur aléatoire

Source de hasard

Générateurs aléatoire et pseudo-aléatoire

### Critères de qualité

Méthode des congruences linéaires

Générateurs basés sur des récurrences sur  $\mathbb{Z}_2$

### Génération de V.A.

### Méthodes de Monte-Carlo

- Il n'est pas aisé de définir précisément ce qui correspond à un bon générateur pseudo-aléatoire et, par extension, à une bonne suite de nombres pseudo-aléatoires. Pour citer D. H. Lehmer, inventeur de la méthode des congruences linéaires :

Une suite (pseudo-)aléatoire est une vague notion couvrant l'idée d'une suite dans laquelle chaque terme est imprévisible pour un non-initié, et dont les éléments satisfont un certain nombre de tests statistiques traditionnels, dépendant éventuellement de l'usage auquel est destinée la suite.

- Cette citation met en avant l'importance à accorder aux tests empiriques permettant de valider la qualité d'un générateur. On peut cependant énumérer plusieurs caractéristiques importantes, voire indispensables, que se doit de posséder un bon générateur.



## Simulation

### Générateur aléatoire

Source de hasard  
Générateurs aléatoire et pseudo-aléatoire

### Critères de qualité

Méthode des congruences linéaires

Générateurs basés sur des récurrences sur  $\mathbb{Z}_2$

### Génération de V.A.

### Méthodes de Monte-Carlo

- L'**imprévisibilité** (pour un non-initié) et l'**équirépartition** (on veut des réalisations d'une loi uniforme) sont deux caractéristiques minimales pour un bon générateur pseudo-aléatoire mais ce n'est pas suffisant. D'autres critères doivent également être étudiés et satisfaits.
- La suite générée doit avoir une **longue période**. Les générateurs utilisés reposent sur des relations de récurrence de la forme

$$x_k = f(x_{k-1})$$

où  $x_k$  est l'**état du générateur** et appartient à un ensemble  $S$  fini (typiquement  $x_k$  est un entier ou un tableau d'entiers).

La **fonction de transfert**  $f$  étant déterministe, la suite des états visités est forcément cyclique, la longueur maximale du cycle étant  $|S|$ . Un bon générateur doit donc avoir un ensemble d'états suffisamment grand et, idéalement, une période maximale égale à  $|S|$ .

## Simulation

### Générateur aléatoire

Source de hasard  
Générateurs aléatoire et  
pseudo-aléatoire

### Critères de qualité

Méthode des  
congruences linéaires

Générateurs basés sur  
des récurrences sur  $\mathbb{Z}_2$

### Génération de V.A.

### Méthodes de Monte-Carlo

- Parmi les tests d'équipartition envisageables, on peut chercher à vérifier que les couples, triplets ou plus généralement les  $n$ -uples de valeurs successives fournies par le générateur remplissent de manière suffisamment uniforme l'hypercube  $[0, 1]^n$  (en tout cas pour les premières valeurs de  $n$ ).
- Le très mauvais générateur RANDU implanté sur les systèmes 370 d'IBM est un exemple de générateurs utilisant la méthode des congruences linéaires.

La fonction de transfert est

$$x_k = (65539 \cdot x_{k-1}) \bmod 2^{31}$$

et les triplets formés à partir des valeurs consécutives obtenues par cette relation se trouvent sur 15 plans équidistants seulement.

## Simulation

### Générateur aléatoire

Source de hasard  
Générateurs aléatoire et  
pseudo-aléatoire

### Critères de qualité

Méthode des  
congruences linéaires  
Générateurs basés sur  
des récurrences sur  $\mathbb{Z}_2$

### Génération de V.A.

### Méthodes de Monte-Carlo

- Dans l'absolu, **la suite de valeurs générées ne doit pas pouvoir être différenciée, à l'aide de tests statistiques, d'une suite de réalisations i.i.d.  $\mathcal{U}(0, 1)$ .**
  - ▶ Il existe de très nombreux tests cherchant à identifier des écarts statistiquement significatifs (et seul l'imagination peut limiter le nombre de ces tests).
  - ▶ Même s'il est inutile d'espérer les passer tous (il est toujours possible de construire un test pour mettre un générateur donné en difficulté), un bon générateur doit en réussir la plupart.
  - ▶ Il existe plusieurs batteries de tests librement accessibles : DIEHARD (Marsaglia, 1996), batterie du NIST (*National Institute of Standards and Technology of the USA*, Rukhin et al., 2001), TestU01 (L'Ecuyer et al., 2007), ...



## Simulation

### Générateur aléatoire

Source de hasard  
Générateurs aléatoire et  
pseudo-aléatoire

### Critères de qualité

Méthode des  
congruences linéaires

Générateurs basés sur  
des récurrences sur  $\mathbb{Z}_2$

### Génération de V.A.

### Méthodes de Monte-Carlo

Finalement, un bon générateur doit aussi

- être rapide à l'exécution ;
- permettre de générer plusieurs fois les mêmes nombres (être « répétable ») ;
- être facile à programmer sur les architectures actuelles ;
- être portable d'un processeur à l'autre ou d'un langage à l'autre ;
- permettre de faire facilement des sauts dans la séquence afin de pouvoir offrir plusieurs suites et sous-suites disjointes ;

Fort heureusement, de nos jours les cas tel que RANDU n'existent plus (on croise les doigts) et les générateurs fournis par les différents langages et logiciels sont normalement de qualité mais un petit tour dans la doc s'impose quand même...

# Méthode des congruences linéaires

## Simulation

### Générateur aléatoire

Source de hasard  
Générateurs aléatoire et  
pseudo-aléatoire

Critères de qualité

### Méthode des congruences linéaires

Générateurs basés sur  
des récurrences sur  $\mathbb{Z}_2$

### Génération de V.A.

### Méthodes de Monte-Carlo

- Introduite en 1948 par D. H. Lehmer, cette méthode et ses généralisations sont encore parmi les plus couramment utilisées en simulation et en calcul numérique.

- La méthode utilise comme fonction de transfert la récurrence

$$x_k = (a \cdot x_{k-1} + c) \bmod m.$$

L'ensemble des états est donc  $S = \mathbb{Z}_m = \{0, \dots, m-1\}$ .

- Pour obtenir un nombre dans l'intervalle unité on pose simplement

$$u_k = \frac{x_k}{m}$$

ou, afin d'éviter de retourner zéro,

$$u_k = \frac{x_k + 1}{m + 1}.$$

# Méthode des congruences linéaires (2)

- Ce type de générateurs a été extensivement étudié et on connaît des critères à respecter lors du choix des paramètres  $a$ ,  $c$  et  $m$  afin d'obtenir la période maximale  $m$  (ou  $m - 1$  si on choisit  $c = 0$ ).
- Une faiblesse importante de cette méthode réside dans la structure de réseau présenté par les  $n$ -uples de valeurs successives de la suite générée.
- Afin d'améliorer les choses
  - ▶ On peut combiner les résultats de deux générateurs (ou plus). On peut ainsi obtenir une plus longue période (jusqu'à multiplier les périodes des deux générateurs de base dans le meilleur des cas) et améliorer la structure de la suite générée.  
Cela reste cependant une opération délicate qui peut aussi avoir l'effet inverse de celui escompté !
  - ▶ On peut aussi utiliser une récurrence d'ordre supérieur à 1 (générateur récursif multiple) dont la fonction de transfert est une relation de récurrence d'ordre  $p$  :

$$x_k = (a_1 x_{k-1} + a_2 x_{k-2} + \cdots + a_p x_{k-p}) \bmod m.$$

# Générateurs basés sur des récurrences sur $\mathbb{Z}_2$

D'autres familles de générateurs pseudo-aléatoires utilisent des récurrences matricielles de la forme

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1}$$

mais en effectuant toutes les opérations modulo 2. L'état  $\mathbf{x}_k$  est alors un vecteur de bits et en choisissant judicieusement la matrice  $\mathbf{A}$  (elle aussi formée uniquement de 0 et de 1), toutes les opérations peuvent se faire très simplement et rapidement.

Parmi les générateurs basés sur ce type d'approche on peut citer

- les générateurs de Tausworthe ;
- les méthodes de registres de décalage à rétroaction linéaire (*linear feedback shift register*, LFSR) et leurs généralisations (*general feedback shift register*, GFSR, et *twisted GFSR*) ;
- Mersenne Twister, WELL, xorshift ;
- ...

# Génération de variables aléatoires



## Simulation

### Générateur aléatoire

### Génération de V.A.

#### Cadre et objectifs

Méthode des fonctions inverses

Variables uniformes

Variables exponentielles

Méthode des mélanges

Hyper-exponentielles

Méthode des convolutions

Variables triangulaires

Méthode des rejets

Variables discrètes

### Méthodes de Monte-Carlo

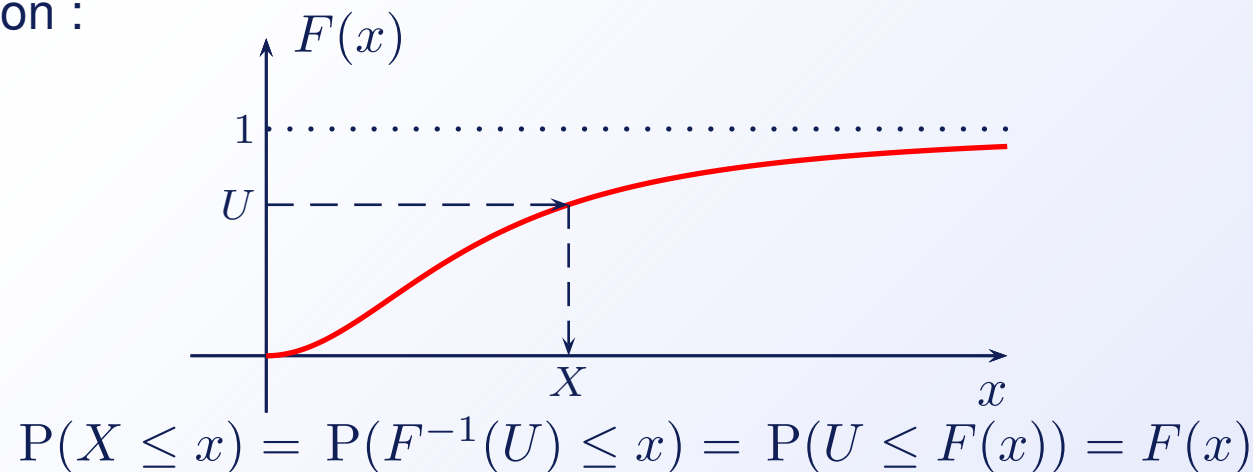
- On dispose d'une source aléatoire permettant de générer à choix
  - ▶ des réalisations indépendantes et identiquement distribuées (i.i.d.) de variables aléatoires (V.A.) suivant la loi uniforme  $\mathcal{U}(0, 1)$  (loi continue),
  - ▶ des réalisations indépendantes et uniformément distribuées dans l'ensemble  $\{0, \dots, n - 1\}$  (loi discrète).
- En pratique ces réalisations proviennent d'un générateur de nombres pseudo-aléatoires mais on fait comme si...
- On cherche à générer des réalisations i.i.d. suivant une loi de probabilités donnée (continue ou discrète).



- Soit  $X$  une variable aléatoire continue de fonction de répartition  $F(x)$  continue et strictement monotone.
- Pour générer une réalisation de  $X$  on utilise la procédure suivante :

- 1) Générer une réalisation  $U$  selon la loi  $\mathcal{U}(0, 1)$
- 2) Calculer  $X = F^{-1}(U)$
- 3) Retourner la valeur  $X$

- Justification :



## Exemple : Variables uniformes $\mathcal{U}(a, b)$

- La densité  $f(x)$  et la fonction de répartition  $F(x)$  d'une loi uniforme  $\mathcal{U}(a, b)$  avec  $a < b$  sont

$$f(x) = \begin{cases} 0 & \text{si } x < a \\ \frac{1}{b-a} & \text{si } a \leq x \leq b \\ 0 & \text{si } x > b \end{cases} \quad \text{et} \quad F(x) = \begin{cases} 0 & \text{si } x < a \\ \frac{x-a}{b-a} & \text{si } a \leq x \leq b \\ 1 & \text{si } x > b \end{cases}$$

- Pour  $0 \leq U \leq 1$  on a

$$U = F(X) = \frac{X-a}{b-a} \quad \Longleftrightarrow \quad X = F^{-1}(U) = a + U(b-a)$$

- Algorithme : 1) Générer  $U \sim \mathcal{U}(0, 1)$

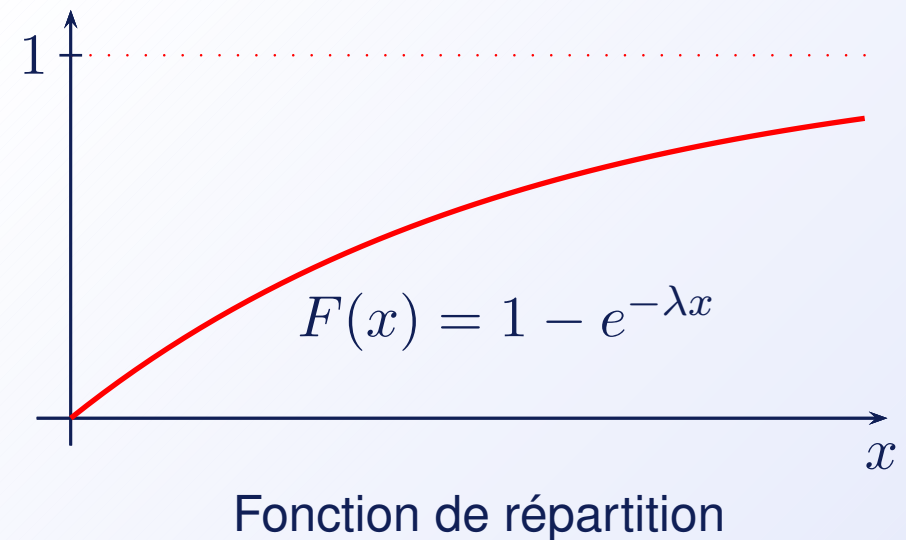
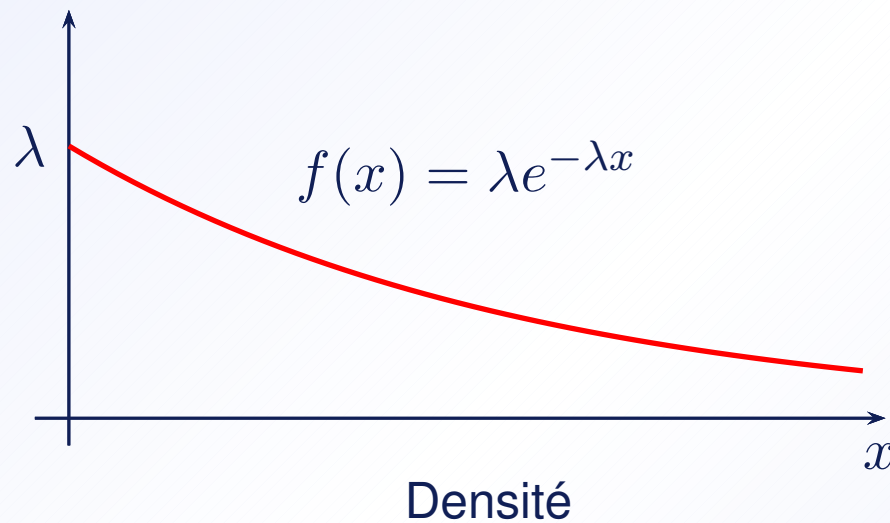
2) Retourner  $X = a + U(b-a)$



# Exemple : Variables exponentielles $\mathcal{E}(\lambda)$

- Une variable aléatoire exponentielle, de paramètre  $\lambda > 0$ , est une variable non négative dont la fonction de densité  $f(x)$  et la fonction de répartition  $F(x)$  sont

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases} \quad \text{et} \quad F(x) = \begin{cases} 1 - e^{-\lambda x} & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$



## Exemple : Variables exponentielles $\mathcal{E}(\lambda)$ (2)

■ Pour  $0 \leq U < 1$  on a

$$U = F(X) = 1 - e^{-\lambda X} \quad \Longleftrightarrow \quad X = F^{-1}(U) = -\frac{1}{\lambda} \ln(1 - U)$$

et la méthode des fonctions inverses aboutit à l'algorithme suivant pour générer des réalisations d'une variable exponentielle de paramètre  $\lambda$ .

Algorithme : 1) Générer  $U \sim \mathcal{U}(0, 1)$

$$2) \text{ Retourner } X = -\frac{1}{\lambda} \ln(1 - U)$$

■ REMARQUE. Au point 2) ci-dessus on peut également retourner la valeur  $X = -\frac{1}{\lambda} \ln(U)$  car si  $U \sim \mathcal{U}(0, 1)$  alors  $1 - U \sim \mathcal{U}(0, 1)$  et réciproquement.

Le choix entre les deux options peut être motivé par les spécificités du générateur pseudo-aléatoire utilisé et plus particulièrement sur le fait qu'il puisse retourner des réalisations  $U$  valant 0 ou 1.

- La **méthode des mélanges** s'applique lorsque la fonction de répartition de la variable aléatoire  $X$  peut s'écrire

$$F(x) = \sum_{k=1}^n p_k F_k(x) \quad \text{avec} \quad \sum_{k=1}^n p_k = 1 \quad \text{et} \quad p_k \geq 0 \quad \forall k$$

et où chaque  $F_k$  est une fonction de répartition.

- S'il est facile de générer des réalisations selon les lois  $F_k$ , on peut générer des réalisations de la variable  $X$  comme suit :

1) Générer un indice  $K$  obéissant à la loi discrète

$$P(K = k) = p_k, \quad k = 1, \dots, n$$

2) Générer  $X$  selon la fonction de répartition  $F_K(x)$

# Exemple : Variables hyper-exponentielles

- La fonction de répartition  $F(x)$  d'une variable hyper-exponentielle  $X$  s'écrit

$$F(x) = p_1 F_1(x) + \cdots + p_n F_n(x) = \sum_{k=1}^n p_k F_k(x)$$

où les poids  $p_k$  sont non négatifs et somment à 1 et les  $F_k$  sont des fonctions de répartition de lois exponentielles de paramètres respectifs  $\lambda_k > 0$ ,  $k = 1, \dots, n$ .

- Une variable hyper-exponentielle  $X$  est donc un mélange (une combinaison convexe) de variables exponentielles indépendantes (mais de taux différents).
- Algorithme : 1) Générer  $K$  tel que  $P(K = k) = p_k$ ,  $k = 1, \dots, n$   
2) Générer  $X \sim \mathcal{E}(\lambda_K)$

Simulation

Générateur aléatoire

Génération de V.A.

Cadre et objectifs

Méthode des fonctions  
inverses

Variables uniformes

Variables exponentielles

Méthode des mélanges

Hyper-exponentielles

**Méthode des  
convolutions**

Variables triangulaires

Méthode des rejets

Variables discrètes

Méthodes de Monte-Carlo

- Une **convolution** est une somme de variables aléatoires indépendantes.

- Si la variable  $X$  est la somme de  $n$  variables indépendantes  $X_k$  :

$$X = X_1 + \cdots + X_n = \sum_{k=1}^n X_k$$

et s'il est facile de générer des réalisations des  $X_k$ , échantillonner selon la loi de  $X$  n'est pas difficile.

- Algorithme :

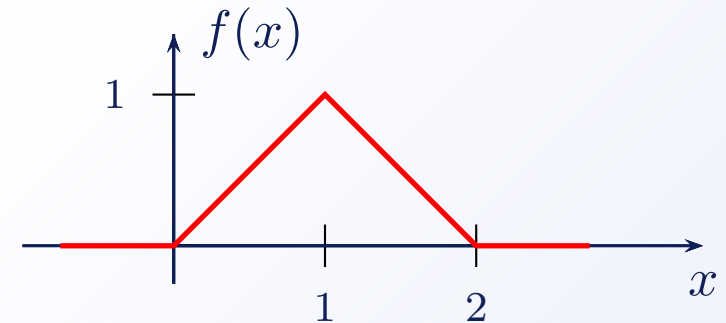
1) Générer  $X_k$  pour  $k = 1, \dots, n$

2) Retourner  $X = X_1 + \cdots + X_n$

## Exemple : Variables triangulaires symétriques $\mathcal{T}(0, 1, 2)$

- La fonction de densité  $f(x)$  d'une variable triangulaire symétrique  $X \sim \mathcal{T}(0, 1, 2)$  est

$$f(x) = \begin{cases} x & \text{si } 0 \leq x \leq 1 \\ 2 - x & \text{si } 1 \leq x \leq 2 \\ 0 & \text{sinon} \end{cases}$$



- La variable  $X$  est simplement la somme de deux variables uniformes  $\mathcal{U}(0, 1)$  indépendantes.

- Algorithme : 1) Générer  $U_1$  et  $U_2$ , indépendantes, de loi  $\mathcal{U}(0, 1)$

2) Retourner  $X = U_1 + U_2$

- Pour générer  $X \sim \mathcal{T}(a, \frac{a+b}{2}, b)$ , avec  $a < b$ , il suffit de retourner

$$X = a + \frac{b - a}{2}(U_1 + U_2).$$



- La **méthode des rejets** (ou **d'acceptation-rejet**) s'applique dans le cadre suivant :

- ▶ on désire générer des réalisations d'une variable aléatoire  $X$  de densité  $f(x)$  (sans connaître de méthodes pour cela) ;
- ▶ on sait échantillonner des variables aléatoires  $Y$  de densité  $g(y)$  ;
- ▶ il existe une constante  $M \geq 1$  telle que  $f(x) \leq M g(x)$  pour tout  $x$  (autrement dit le graphe de  $M g(x)$  est au-dessus de celui de  $f(x)$ ).

- Algorithme :

1) Générer  $Y$  selon la densité  $g$

2) Générer  $U \sim \mathcal{U}(0, 1)$  indépendante de  $Y$

3) Si  $U \leq \frac{f(Y)}{M g(Y)}$  accepter  $Y$  et retourner  $X = Y$

Sinon retourner au point 1)

# Validité de la méthode des rejets

- On doit vérifier que  $P(X \leq x) = \int_{-\infty}^x f(y)dy$ .
- Par construction, on a  $P(X \leq x) = P(Y \leq x \mid Y \text{ est acceptée})$ .
- Maintenant, pour  $Y$  fixée, la probabilité d'accepter  $Y$  est  $P\left(U \leq \frac{f(Y)}{Mg(Y)}\right) = \frac{f(Y)}{Mg(Y)}$  car  $U$  est uniforme et  $Y$  et  $U$  sont indépendantes.
- On a donc  $P(Y \leq x \text{ et } Y \text{ acceptée}) = \int_{-\infty}^x \frac{f(y)}{Mg(y)}g(y)dy = \frac{1}{M} \int_{-\infty}^x f(y)dy$ .
- Si on fait tendre  $x$  vers l'infini, on obtient

$$P(Y \text{ acceptée}) = \lim_{x \rightarrow +\infty} P(Y \leq x \text{ et } Y \text{ acceptée}) = \lim_{x \rightarrow +\infty} \frac{1}{M} \int_{-\infty}^x f(y)dy = \frac{1}{M}.$$

- Ainsi  $P(X \leq x) = \frac{P(Y \leq x \text{ et } Y \text{ acceptée})}{P(Y \text{ acceptée})} = \int_{-\infty}^x f(y)dy$ .



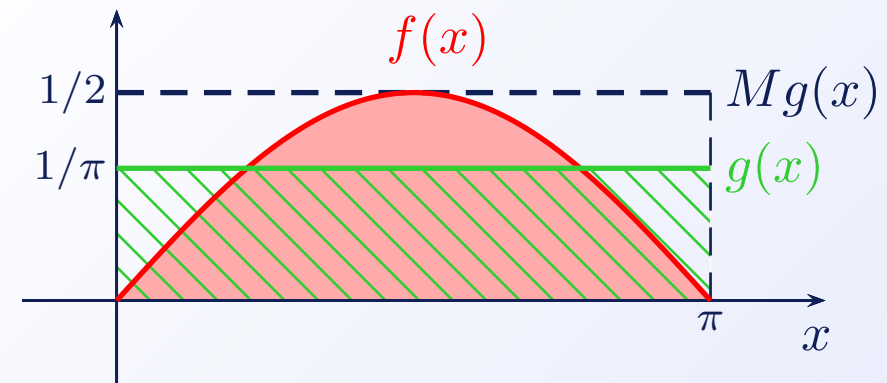
- On désire échantillonner des variables aléatoires  $X$  de densité

$$f(x) = \frac{\sin(x)}{2}$$

pour  $0 \leq x \leq \pi$ .

- Soit  $g(x) = \frac{1}{\pi}$  la densité d'une variable  $\mathcal{U}(0, \pi)$

- On a  $f(x) \leq \frac{\pi}{2} \cdot g(x)$ , pour  $0 \leq x \leq \pi$ , et  $M = \frac{\pi}{2}$ .



- Algorithme : 1) Générer  $Y$  et  $U \sim \mathcal{U}(0, 1)$

2) Si  $U \leq \sin(\pi Y)$  accepter  $Y$  et retourner  $X = \pi Y$

Sinon retourner au point 1)

# Comportement de la méthode des rejets

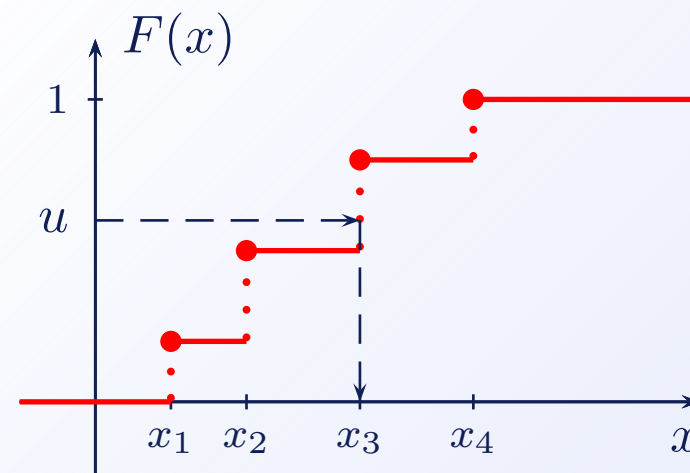
- La probabilité d'accepter  $Y$  est  $1/M$ .
- Le nombre d'itérations avant de retourner une réalisation  $X$  est donc une variable aléatoire géométrique de paramètre  $1/M$  et il faut en moyenne  $M$  itérations pour chaque génération.
- Les performances de la méthode dépendent
  - ▶ de la constance  $M$  : plus elle est proche de 1 mieux c'est. Il faut donc rechercher une densité  $g(x)$  qui épouse au mieux la densité  $f(x)$ .
  - ▶ de la facilité à échantillonner la densité  $g(x)$ .
  - ▶ de la facilité à évaluer  $f(x)/g(x)$ .

- Soit  $X$  une variable aléatoire discrète pouvant prendre  $n$  valeurs différentes et obéissant à la loi de probabilité

$$P(X = x_k) = p_k, \quad k = 1, \dots, n.$$

- La fonction de répartition de  $X$  est une fonction « en escalier » :

$$F(x) = P(X \leq x) = \sum_{x_k \leq x} p_k.$$



- On peut générer des réalisations  $X$  à l'aide de la **méthode des fonctions inverses généralisées** en définissant

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

## Variables aléatoires discrètes (2)

■ Algorithme :

1) Générer  $U \sim \mathcal{U}(0, 1)$

2) Déterminer  $j$  tel que  $\sum_{k=1}^{j-1} p_k < U \leq \sum_{k=1}^j p_k$  et retourner  $X = x_j$

■ Variable de Bernoulli  $X \sim \mathcal{B}(p)$  :  $P(X = 0) = 1 - p$  et  $P(X = 1) = p$

Algorithme : 1) Générer  $U \sim \mathcal{U}(0, 1)$

2) Si  $U \leq p$  retourner  $X = 1$ , sinon retourner  $X = 0$

■ Variable de Poisson  $X \sim \mathcal{P}(\lambda)$  :  $P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}$ ,  $k = 0, 1, \dots$

Vu comme un processus de comptage, une variable de Poisson  $X$  est égale au nombre d'événements prenant place dans un intervalle de durée 1 si les temps entre deux événements successifs sont i.i.d. selon une loi exponentielle  $\mathcal{E}(\lambda)$ .

Algorithme : 1) Générer  $Y_0, Y_1, \dots \sim \mathcal{E}(1)$  jusqu'à la première valeur  $k$  telle que

$$\sum_{j=0}^k Y_j > \lambda$$

2) Retourner  $X = k$

# Méthodes de Monte-Carlo



Simulation

Générateur aléatoire

Génération de V.A.

Méthodes de Monte-Carlo

**Méthodes de  
Monte-Carlo**

Exemple : Estim. de  $\pi$

Estimateur de  
l'espérance

Estimateur de la  
variance

Intervalle de confiance  
pour  $\mu = E(X)$

Convergence

Exemple

- En termes généraux, les **méthodes de Monte-Carlo** désignent des techniques de calcul numérique cherchant à évaluer une quantité **déterministe** à l'aide de tirages **aléatoires** répétés.
- Ce sont des méthodes d'**approximation**, **probabilistes**, qui fournissent des **estimations** des grandeurs cherchées.
- Historiquement, le développement de la méthode remonte aux années 1940 avec les travaux de S. Ulman et de son équipe sur la mise au point de la bombe atomique. L'article fondateur, *The Monte-Carlo Method*, N. Metropolis et S. Ulman, a été publié en 1949.
- Le terme Monte-Carlo, dû à N. Metropolis, fait référence aux jeux de hasard pratiqués dans les casinos de la Principauté monégasque.

Simulation

Générateur aléatoire

Génération de V.A.

Méthodes de Monte-Carlo

**Méthodes de  
Monte-Carlo**

Exemple : Estim. de  $\pi$

Estimateur de  
l'espérance

Estimateur de la  
variance

Intervalle de confiance  
pour  $\mu = E(X)$

Convergence

Exemple

Les méthodes de Monte-Carlo trouvent des applications dans de très nombreux domaines :

- en physique : physique atomique et quantique, physique statistique, ...
- en ingénierie : réseaux de télécommunications, théorie de la fiabilité, ...
- en mathématiques et en statistiques : calcul d'intégrales (multiples), résolution de systèmes linéaires (de très grande taille), recuit simulé, méthodes MCMC (*Markov Chain Monte Carlo*), ...
- en informatique : infographie (*ray tracing*), ...
- en économie : analyse de risque en finances, ...
- en théorie des jeux : backgammon (TD-Gammon), bridge (Bridge Baron), go (MoGo, AlphaGo), ...
- ...



# Exemple : Estimation de $\pi$

Simulation

Générateur aléatoire

Génération de V.A.

Méthodes de Monte-Carlo

Méthodes de  
Monte-Carlo

**Exemple : Estim. de  $\pi$**

Estimateur de  
l'espérance

Estimateur de la  
variance

Intervalle de confiance  
pour  $\mu = E(X)$

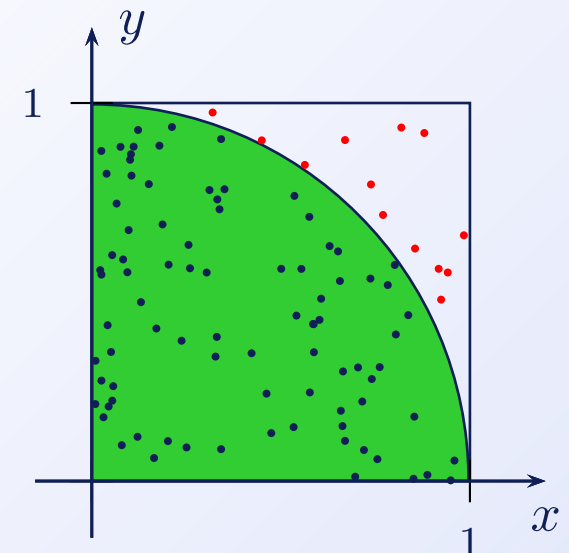
Convergence

Exemple

- On considère le quart de cercle contenu dans le carré unité  $[0, 1] \times [0, 1]$ .
- L'aire de ce quart de cercle est égale à  $\frac{\pi}{4}$  ( $\simeq 0.7854$ ).
- La probabilité  $p$  qu'un point tiré au hasard dans le carré unité (c.-à-d. selon une loi uniforme) se situe dans le quart de cercle est

$$p = \frac{\text{aire du quart de cercle}}{\text{aire du carré unité}} = \frac{\frac{\pi}{4}}{1} = \frac{\pi}{4}.$$

- Si on génère  $n$  points indépendants et uniformément distribués dans le carré unité on peut estimer la valeur de  $p$  en comptant le nombre de points atterrissant dans le quart de cercle.
- Pour  $n = 100$  points, l'échantillon représenté ci-contre donne l'estimation  $\hat{p} = \frac{87}{100} = 0.87$ .





Pour caricatural et inefficace qu'il soit, l'exemple précédent contient déjà les éléments présents dans la plupart des méthodes de Monte-Carlo :

- On part d'un problème numérique consistant à calculer une grandeur déterministe : Calculer l'aire d'un quart de cercle unité, c.-à-d. la valeur de l'intégrale

$$G = \int_0^1 \sqrt{1 - x^2} dx.$$

- On reformule le problème en termes probabilistes : Calculer la probabilité  $p$  qu'un point aléatoire (c.-à-d. uniformément distribué) dans le carré unité se retrouve dans le quart de cercle ou, encore, calculer l'espérance  $E(f(X))$  de la fonction

$$f(X) = \begin{cases} 1 & \text{si } x^2 + y^2 \leq 1 \\ 0 & \text{sinon} \end{cases}$$

où  $X(x, y)$  est un vecteur aléatoire uniformément distribué dans le carré unité.

## Analyse de l'exemple (2)

- On simule le problème probabiliste, c.-à-d. on génère un **échantillon** de  $N$  points indépendants et uniformément distribués dans le carré unité et on compte le nombre de points tombant dans le quart de cercle (le nombre de succès) :
  - 1) Poser  $S = 0$       // **compte le nombre de succès**
  - 2) Pour  $k$  de 1 à  $N$  faire
  - 3)     Générer  $X$  et  $Y \sim \mathcal{U}(0, 1)$
  - 4)     Si  $X^2 + Y^2 \leq 1$  incrémenter  $S$
  - 5) Retourner  $\hat{p} = \frac{S}{N}$       // **estimateur de  $p = \frac{\pi}{4}$**
- On approche la grandeur déterministe cherchée à l'aide d'un **estimateur** (très souvent une simple moyenne arithmétique) de l'espérance  $E(f(X))$ .

# Estimateur classique de l'espérance

- Un problème de base en simulation stochastique consiste à calculer, ou plutôt à estimer, l'**espérance**  $\mu = E(X)$  d'une variable aléatoire  $X$ .
- Dans l'exemple, la variable  $X$  est une variable de Bernoulli qui vaut 1 si le point se trouve dans le quart de cercle et 0 sinon.
- Plus généralement, cette variable  $X$  représente une **mesure de performance** que l'on cherche à évaluer. Par exemple, le temps d'attente d'un client cherchant à contacter un *call center* ou celui s'écoulant entre deux suppressions de trains sur le réseau CFF.
- Le calcul exact (analytique) de  $E(X)$  n'est souvent pas possible, d'où le recours à la simulation. Cette dernière permet de générer des **observations**  $(x_1, x_2, \dots, x_n)$  (indépendantes ou non) de  $X$ , c.-à-d. un échantillon de  $n$  réalisations de  $X$ .
- L'espérance  $E(X)$  correspondant à la **valeur moyenne** que prend la variable  $X$ , l'**estimateur** classique  $\hat{\mu} = \bar{x}$  de cette espérance n'est donc rien d'autre que la **moyenne arithmétique** des valeurs générées :

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

# Avantage et limitation de l'estimateur $\bar{x}$

- Outre sa simplicité, l'un des avantages principaux de l'estimateur  $\hat{\mu} = \bar{x}$  est qu'il est **consistant** (il converge vers la vraie valeur  $\mu$  de l'espérance lorsque le nombre d'observations augmente) **même si les valeurs mesurées ne sont pas indépendantes**.
- Si l'espérance est une grandeur essentielle dans l'étude d'une variable aléatoire  $X$ , elle ne fournit, en revanche, aucune information sur la **variabilité** de  $X$ .
- SOLUTION (PARTIELLE). Mesurer la variabilité de  $X$  par rapport à son espérance en calculant (ou, plus exactement, en estimant) également sa **variance** (ou son écart-type).
- La **variance**  $\sigma^2 = \text{Var}(X)$  d'une variable aléatoire  $X$  d'espérance  $\mu$  est égale à

$$\sigma^2 = \text{Var}(X) = \text{E}((X - \mu)^2).$$

Autrement dit, la variance est égale à l'espérance de l'écart au carré entre  $X$  et son espérance  $\mu = \text{E}(X)$ .

- Pour estimer la variance, on remplace l'espérance de l'écart au carré par la moyenne arithmétique de ces écarts (sur les  $n$  valeurs de l'échantillon) :

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{k=1}^n (x_k - \mu)^2.$$

- PROBLÈME. L'espérance  $\mu = E(X)$  de la variable  $X$  n'est pas connue en général !
- SOLUTION (PARTIELLE). Remplacer dans l'expression précédente l'espérance  $\mu$  par son estimateur  $\hat{\mu} = \bar{x}$  afin d'obtenir l'estimateur de la variance :

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



## Estimateur de la variance (2)

- PROBLÈME. L'estimateur précédent est **consistant** (il converge vers  $\sigma^2$  lorsque  $n$  tend vers l'infini) mais il est **biaisé**. En effet, son espérance est

$$E(\hat{\sigma}^2) = \frac{n-1}{n}\sigma^2.$$

- SOLUTION. Pour éliminer ce biais il suffit de multiplier  $\hat{\sigma}^2$  par  $\frac{n}{n-1}$ . On obtient alors l'estimateur classique de la variance

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

- REMARQUE. Pour  $n$  grand le rapport  $\frac{n}{n-1}$  est très proche de 1 et la différence entre les estimateurs  $\hat{\sigma}^2$  et  $s^2$  devient plus théorique que pratique.



- Au niveau pratique, une question souvent plus importante que la variabilité de la variable aléatoire  $X$  par rapport à son espérance  $\mu$  réside dans la **précision** de l'estimateur  $\bar{x}$  de cette espérance.
- SOLUTION. Ne pas se limiter à estimer  $E(X)$  par la moyenne  $\bar{x}$  mais calculer également un **intervalle de confiance** pour  $\hat{\mu} = \bar{x}$ .
- Un tel intervalle s'obtient à partir du **théorème central limite** :

**Théorème central limite.** Soit  $(X_1, X_2, \dots, X_n)$  une suite de variables aléatoires **in-dépendantes** et **identiquement distribuées** (i.i.d.), d'espérance  $\mu$  et de variance  $\sigma^2$ , alors la distribution de

$$\frac{(X_1 + X_2 + \dots + X_n) - n\mu}{\sigma\sqrt{n}} = \frac{\frac{1}{n} \sum_{k=1}^n X_k - \mu}{\sigma/\sqrt{n}}$$

tend vers une loi normale centrée réduite  $\mathcal{N}(0, 1)$  lorsque  $n$  tend vers l'infini.

- Si on note  $z_{1-\alpha}$  le quantile de seuil  $1 - \alpha$  de la loi normale centrée réduite, c'est-à-dire la valeur pour laquelle une variable aléatoire  $Z \sim \mathcal{N}(0, 1)$  vérifie

$$P(Z \leq z_{1-\alpha}) = 1 - \alpha \quad \Longleftrightarrow \quad P(Z \geq z_{1-\alpha}) = \alpha$$

on a, pour  $n$  suffisamment grand,

$$P\left(-z_{1-\frac{\alpha}{2}} \leq \frac{(X_1 + X_2 + \dots + X_n) - n\mu}{\sigma\sqrt{n}} \leq z_{1-\frac{\alpha}{2}}\right) = 1 - \alpha$$

- En appliquant quelques transformations au résultat précédent, on obtient

$$\begin{aligned} 1 - \alpha &= P\left(-z_{1-\frac{\alpha}{2}} \leq \frac{(X_1 + X_2 + \dots + X_n) - n\mu}{\sigma\sqrt{n}} \leq z_{1-\frac{\alpha}{2}}\right) \\ &= P\left(\sum_{k=1}^n X_k - z_{1-\frac{\alpha}{2}}\sigma\sqrt{n} \leq n\mu \leq \sum_{k=1}^n X_k + z_{1-\frac{\alpha}{2}}\sigma\sqrt{n}\right) \\ &= P\left(\frac{1}{n}\sum_{k=1}^n X_k - z_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}} \leq \mu \leq \frac{1}{n}\sum_{k=1}^n X_k + z_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}}\right) \end{aligned}$$

# Intervalle de confiance pour $\mu = E(X)$

Simulation

Générateur aléatoire

Génération de V.A.

Méthodes de Monte-Carlo

Méthodes de  
Monte-Carlo

Exemple : Estim. de  $\pi$

Estimateur de  
l'espérance

Estimateur de la  
variance

**Intervalle de confiance  
pour  $\mu = E(X)$**

Convergence

Exemple

- Ainsi, avec un **seuil de confiance  $1 - \alpha$** , l'espérance  $\mu = E(x)$  (inconnue) se trouve dans l'**intervalle de confiance bilatéral**

$$\left[ \frac{1}{n} \sum_{k=1}^n X_k - z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}, \frac{1}{n} \sum_{k=1}^n X_k + z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}} \right] = \left[ \bar{X} \pm z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}} \right]$$

- **PROBLÈME.** En pratique l'écart-type  $\sigma$  de la variable  $X$  n'est que rarement connu et doit être estimé par

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

où  $\bar{x}$  est l'estimateur empirique de l'espérance  $\mu$  :

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

# Intervalle de confiance pour $\mu = E(X)$ (2)

Simulation

Générateur aléatoire

Génération de V.A.

Méthodes de Monte-Carlo

Méthodes de  
Monte-Carlo

Exemple : Estim. de  $\pi$

Estimateur de  
l'espérance

Estimateur de la  
variance

**Intervalle de confiance  
pour  $\mu = E(X)$**

Convergence

Exemple

■ En résumé, si

$$(x_1, x_2, \dots, x_n)$$

sont  $n$  réalisations **indépendantes** de la variable aléatoire  $X$ , l'**intervalle de confiance bilatéral au seuil  $1 - \alpha$**  pour l'espérance  $\mu = E(X)$ , basé sur l'estimateur  $\hat{\mu} = \bar{x}$ , est

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

où

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

et

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

# Remarques et précisions pratiques

- Le théorème central limite est un résultat **asymptotique**. Il est donc nécessaire d'avoir un nombre  $n$  d'observations suffisamment grand.
  - ▶ En pratique on cherchera à avoir  $n \geq 30$ , la valeur « magique » de 30 étant considérée comme suffisante pour une convergence acceptable dans le théorème central limite.
- Comme on utilise l'estimateur empirique  $s$  de l'écart-type, on doit (ou devrait) remplacer le quantile  $z_{1-\frac{\alpha}{2}}$  de la loi normale par le quantile  $t_{n-1, 1-\frac{\alpha}{2}}$  de la loi  $t$  de Student à  $n - 1$  degrés de liberté.
  - ▶ Pour  $n$  grand (nettement plus grand que 30), la différence entre les deux lois devient minime et est (ou peut être) négligée.
- Les deux seuils  $1 - \alpha$  les plus utilisés en pratique et les quantiles  $z_{1-\frac{\alpha}{2}}$  et  $t_{n-1, 1-\frac{\alpha}{2}}$  correspondants (arrondis à trois chiffres après la virgule et pour quelques valeurs de  $n$ ) sont

$1 - \alpha$	$z_{1-\frac{\alpha}{2}}$	$t_{29, 1-\frac{\alpha}{2}}$	$t_{99, 1-\frac{\alpha}{2}}$	$t_{199, 1-\frac{\alpha}{2}}$	$t_{499, 1-\frac{\alpha}{2}}$
95%	1.960	2.045	1.984	1.972	1.965
99%	2.576	2.756	2.626	2.601	2.586

# Vitesse de convergence des méthodes de Monte-Carlo

Simulation

Générateur aléatoire

Génération de V.A.

Méthodes de Monte-Carlo

Méthodes de  
Monte-Carlo

Exemple : Estim. de  $\pi$

Estimateur de  
l'espérance

Estimateur de la  
variance

Intervalle de confiance  
pour  $\mu = E(X)$

**Convergence**

Exemple

- La largeur  $\Delta_{I_c}$  de l'intervalle de confiance est

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

et la convergence se fait en  $O(n^{-1/2}) = O\left(\frac{1}{\sqrt{n}}\right)$ .

- **CONSÉQUENCE.** Pour diviser par 2 la largeur de l'intervalle de confiance il faut générer 4 fois plus de réalisations et pour diviser par 10 cette largeur, et gagner une décimale de précision, il faut un échantillon 100 fois plus grand !
- Les **méthodes de réduction de variance** tentent de réduire la largeur des intervalles de confiance d'un **facteur constant**, pour une même taille d'échantillon  $n$ , en agissant sur l'écart-type  $\sigma$  de  $X$  et par conséquent sur son estimateur  $s$ .



- Pour l'estimation de

$$p = E(X) = \frac{\pi}{4} \simeq 0.785398,$$

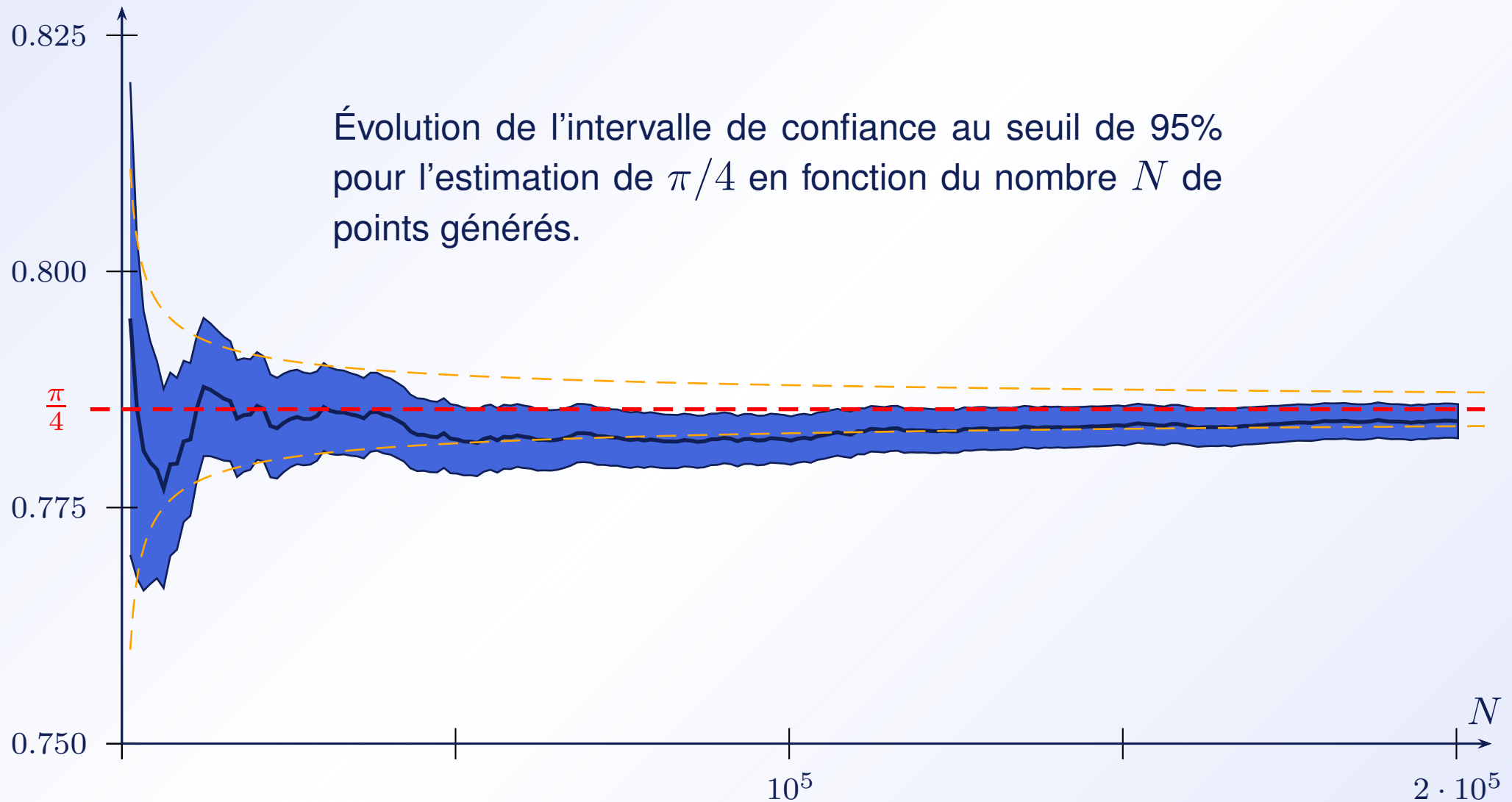
la variable  $X$  est une variable de Bernoulli de paramètre  $p$  et sa variance est  $\sigma^2 = p(1 - p)$ .

- Résultats pour des échantillons de différentes tailles  $N$  ( $\Delta_{I_c}$  = largeur de l'I.C.) :

$N$	$\hat{p}$	$\hat{\sigma}$	$I_c$ à 95%	$\Delta_{I_c}$
$10^3$	0.79500	0.40370	[0.76998, 0.82002]	0.05004
$10^4$	0.78220	0.41275	[0.77411, 0.79029]	0.01618
$10^5$	0.78208	0.41283	[0.77952, 0.78464]	0.00512
$10^6$	0.78516	0.41071	[0.78435, 0.78596]	0.00161
$10^7$	0.78539	0.41055	[0.78514, 0.78565]	0.00051
$10^8$	0.78536	0.41057	[0.78528, 0.78544]	0.00016

# Vitesse de convergence

Évolution de l'intervalle de confiance au seuil de 95% pour l'estimation de  $\pi/4$  en fonction du nombre  $N$  de points générés.



Représentation de 100 intervalles de confiance au seuil de 95% pour l'estimation de  $\pi/4$  obtenus en générant des échantillons de taille  $N = 10^6$  points.

Sur les 100 intervalles, 96 contiennent  $\pi/4$  ce qui correspond à un seuil de confiance empirique de 96%.

