

Chapitre 1: Introduction

Lisez d'abord l'appendice, qui rappelle certaines notions de base en probabilité et statistique, disponible sur la page web du cours.

ATTENTION:

Ces “slides” **ne sont pas** les notes de cours.

Elle ne sont qu'un support pour les présentations en classe et ne contiennent que des explications partielles.

Systèmes, Modèles et Simulation

Modèle: description simplifiée d'un système, dans le but de d'évaluer sa performance ou l'effet de certaines décisions.

Systèmes, Modèles et Simulation

Modèle: description simplifiée d'un système, dans le but de d'évaluer sa performance ou l'effet de certaines décisions.

Simulation: faire évoluer le modèle d'un système en fournissant les entrées appropriées et observer les résultats.

Systèmes, Modèles et Simulation

Modèle: description simplifiée d'un système, dans le but de d'évaluer sa performance ou l'effet de certaines décisions.

Simulation: faire évoluer le modèle d'un système en fournissant les entrées appropriées et observer les résultats.

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

Systèmes, Modèles et Simulation

Modèle: description simplifiée d'un système, dans le but de d'évaluer sa performance ou l'effet de certaines décisions.

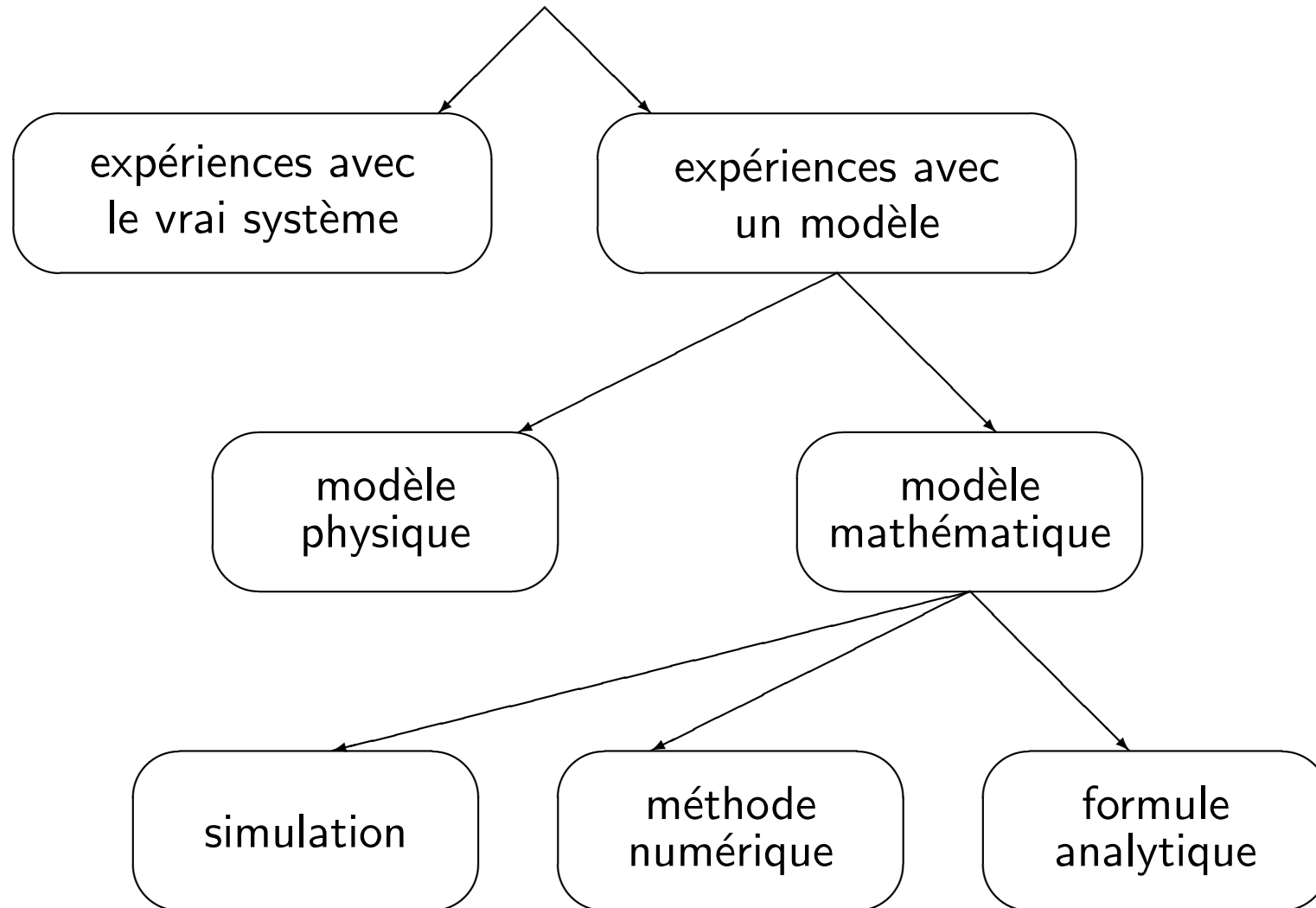
Simulation: faire évoluer le modèle d'un système en fournissant les entrées appropriées et observer les résultats.

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

Exemples:

- Prévisions météo; Simulateurs de vol; jeux vidéo;
- Outils pour la prise de décision: conception d'une usine; planification de la production; gestion des soins de santé; transport; télécommunications;
- simulation en finance, assurances, économie; etc.

Types de modèles



Avantages des modèles mathématiques et de la simulation:

- La simulation est **non destructrice**. Les erreurs ne sont pas coûteuses.
- Le système considéré n'a même pas besoin d'**exister**.
- On peut **répéter** à volonté des expériences identiques ou similaires dans les mêmes conditions.
- On peut souvent simuler un système beaucoup plus **rapidement** qu'il évolue dans la réalité (par ex., 2 années d'opération d'une usine en quelques minutes).
- On peut simuler des modèles très **complexes**, plus **réalistes** que ceux que l'on peut résoudre par des formules analytiques ou par les méthodes d'optimisation classiques.
- L'**animation graphique** peut permettre de voir évoluer le modèle.

Inconvénients:

- **Coût**: La modélisation et la programmation peuvent demander beaucoup d'effort, de temps et d'argent.
- Les **temps d'exécution** (CPU) peuvent devenir excessifs.
- L'**analyse statistique** des résultats n'est pas toujours simple.
- Ne fournit habituellement que des **estimations**; l'optimisation est beaucoup plus difficile par simulation que via les outils habituels de programmation mathématique.

Notre objectif dans ce cours: apprendre à **simuler efficacement** et à analyser correctement les résultats (du point de vue statistique).

Inconvénients:

- **Coût**: La modélisation et la programmation peuvent demander beaucoup d'effort, de temps et d'argent.
- Les **temps d'exécution** (CPU) peuvent devenir excessifs.
- L'**analyse statistique** des résultats n'est pas toujours simple.
- Ne fournit habituellement que des **estimations**; l'optimisation est beaucoup plus difficile par simulation que via les outils habituels de programmation mathématique.

Notre objectif dans ce cours: apprendre à **simuler efficacement** et à analyser correctement les résultats (du point de vue statistique).

Les **méthodes Monte Carlo** sont aussi utilisées pour des applications qui ne correspondent pas nécessairement à la simulation du comportement d'un système complexe. Exemples: en combinatoire, pour compter des configurations; en statistique, pour estimer le niveau ou la p -valeur d'un test d'hypothèse; etc.

Types de modèles mathématiques

Modèle analytique: on obtient une formule.

Difficile à mettre au point mais facile à utiliser.

Exemples: Black-Scholes, formule balistique, files d'attente $M/M/s$, etc.

Types de modèles mathématiques

Modèle analytique: on obtient une formule.

Difficile à mettre au point mais facile à utiliser.

Exemples: Black-Scholes, formule balistique, files d'attente $M/M/s$, etc.

Exemple: Soit W le temps d'attente d'un client pris au hasard dans un modèle de file $M/M/s$ avec taux d'arrivée λ , taux de service μ , $s > \lambda/\mu$ serveurs, et qui évolue indéfiniment.

Dans ce cas, il existe une formule exacte pour $\mathbb{P}[W = 0]$, et on sait que conditionnellement à $W > 0$, W suit une loi exponentielle de moyenne $1/(s\mu - \lambda)$.

Types de modèles mathématiques

Modèle analytique: on obtient une formule.

Difficile à mettre au point mais facile à utiliser.

Exemples: Black-Scholes, formule balistique, files d'attente $M/M/s$, etc.

Exemple: Soit W le temps d'attente d'un client pris au hasard dans un modèle de file $M/M/s$ avec taux d'arrivée λ , taux de service μ , $s > \lambda/\mu$ serveurs, et qui évolue indéfiniment.

Dans ce cas, il existe une formule exacte pour $\mathbb{P}[W = 0]$, et on sait que conditionnellement à $W > 0$, W suit une loi exponentielle de moyenne $1/(s\mu - \lambda)$.

Méthodes numériques: itératives et/ou approximatives. E.g., programmation linéaire et non linéaire, programmation dynamique, systèmes linéaires pour grosses chaînes de Markov, méthodes éléments finis pour équations différentielles.

Types de modèles mathématiques

Modèle analytique: on obtient une formule.

Difficile à mettre au point mais facile à utiliser.

Exemples: Black-Scholes, formule balistique, files d'attente $M/M/s$, etc.

Exemple: Soit W le temps d'attente d'un client pris au hasard dans un modèle de file $M/M/s$ avec taux d'arrivée λ , taux de service μ , $s > \lambda/\mu$ serveurs, et qui évolue indéfiniment.

Dans ce cas, il existe une formule exacte pour $\mathbb{P}[W = 0]$, et on sait que conditionnellement à $W > 0$, W suit une loi exponentielle de moyenne $1/(s\mu - \lambda)$.

Méthodes numériques: itératives et/ou approximatives. E.g., programmation linéaire et non linéaire, programmation dynamique, systèmes linéaires pour grosses chaînes de Markov, méthodes éléments finis pour équations différentielles.

Simulation stochastique: implique de l'incertitude.

On doit simuler des variables aléatoires ou des processus stochastiques.

Outils de base: probabilités et statistique.

Un modèle de simulation sera habituellement plus détaillé et réaliste qu'un modèle analytique. Par contre, une formule analytique peut donner une meilleure intuition des principales propriétés du système et sera moins coûteuse à utiliser.

Un modèle de simulation sera habituellement plus détaillé et **réaliste** qu'un modèle analytique. Par contre, une formule analytique peut donner une meilleure **intuition** des principales propriétés du système et sera moins coûteuse à utiliser.

Modèle déterministe: Aucun aléa.

Ex.: Équations différentielles; programmation linéaire; . . .

Modèle Stochastique: Comportement aléatoire.

Ex.: Files d'attente; modèles de fiabilité; marchés financiers;. . .

Un modèle de simulation sera habituellement plus détaillé et **réaliste** qu'un modèle analytique. Par contre, une formule analytique peut donner une meilleure **intuition** des principales propriétés du système et sera moins coûteuse à utiliser.

Modèle déterministe: Aucun aléa.

Ex.: Équations différentielles; programmation linéaire; . . .

Modèle Stochastique: Comportement aléatoire.

Ex.: Files d'attente; modèles de fiabilité; marchés financiers; . . .

Modèle statique: Le "temps" ne joue pas un rôle prépondérant.

Ex.: Fiabilité d'un réseau; . . .

Modèle dynamique: On s'intéresse à l'évolution dans le temps.

Ex.: Trafic routier ou téléphonique; Système d'inventaire; . . .

Simulation des modèles dynamiques:

État du système: Les valeurs d'un ensemble de variables décrivant le système à un instant donné.

Simulation des modèles dynamiques:

État du système: Les valeurs d'un ensemble de variables décrivant le système à un instant donné.

Modèle en temps continu (simulation continue): L'état change continûment dans le temps, habituellement selon des équations différentielles.

Ex.: Un objet qui se déplace; prédictions météo; simulateurs de vol; . . .

Simulation des modèles dynamiques:

État du système: Les valeurs d'un ensemble de variables décrivant le système à un instant donné.

Modèle en temps continu (simulation continue): L'état change continûment dans le temps, habituellement selon des équations différentielles.

Ex.: Un objet qui se déplace; prédictions météo; simulateurs de vol; . . .

Modèle à événements discrets: L'état change par sauts, lors de l'occurrence d'événements, qui se produisent à des instants habituellement aléatoires
 $0 = t_0 \leq t_1 \leq t_2 \leq t_3 \cdots$.

Simulation des modèles dynamiques:

État du système: Les valeurs d'un ensemble de variables décrivant le système à un instant donné.

Modèle en temps continu (simulation continue): L'état change continûment dans le temps, habituellement selon des équations différentielles.

Ex.: Un objet qui se déplace; prédictions météo; simulateurs de vol; . . .

Modèle à événements discrets: L'état change par sauts, lors de l'occurrence d'événements, qui se produisent à des instants habituellement aléatoires
 $0 = t_0 \leq t_1 \leq t_2 \leq t_3 \cdots$.

Modèle en temps discret: $t_i = i$ pour tout i .

Simulation des modèles dynamiques:

État du système: Les valeurs d'un ensemble de variables décrivant le système à un instant donné.

Modèle en temps continu (simulation continue): L'état change continûment dans le temps, habituellement selon des équations différentielles.

Ex.: Un objet qui se déplace; prédictions météo; simulateurs de vol; . . .

Modèle à événements discrets: L'état change par sauts, lors de l'occurrence d'événements, qui se produisent à des instants habituellement aléatoires $0 = t_0 \leq t_1 \leq t_2 \leq t_3 \cdots$.

Modèle en temps discret: $t_i = i$ pour tout i .

Modèle hybride: Mélange d'évolution continue et d'événements discrets.

Simulation des modèles dynamiques:

État du système: Les valeurs d'un ensemble de variables décrivant le système à un instant donné.

Modèle en temps continu (simulation continue): L'état change continûment dans le temps, habituellement selon des équations différentielles.

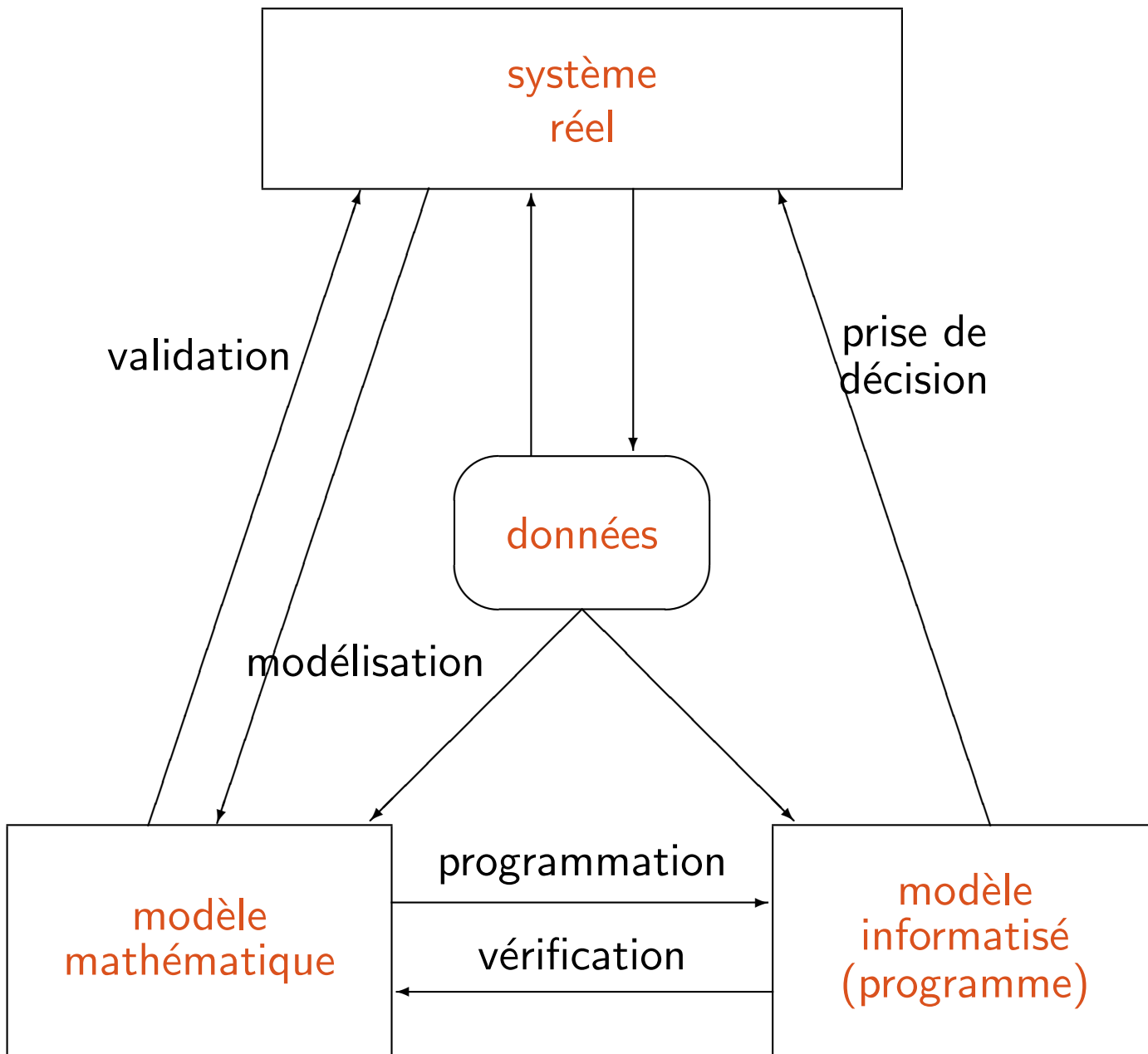
Ex.: Un objet qui se déplace; prédictions météo; simulateurs de vol; . . .

Modèle à événements discrets: L'état change par sauts, lors de l'occurrence d'événements, qui se produisent à des instants habituellement aléatoires $0 = t_0 \leq t_1 \leq t_2 \leq t_3 \dots$.

Modèle en temps discret: $t_i = i$ pour tout i .

Modèle hybride: Mélange d'évolution continue et d'événements discrets.

Intérêt principal dans ce cours: **la simulation stochastique** et (un peu) simulation à événements discrets.



Validité d'un modèle: doit reproduire les aspects importants du comportement du système **pour l'application visée**.

La notion de validité est relative et dépend du niveau de précision voulu. Il n'y a pas de modèle "100% valide". Un modèle est toujours une **approximation** de la réalité, construit en faisant des compromis.

Validité d'un modèle: doit reproduire les aspects importants du comportement du système **pour l'application visée**.

La notion de validité est relative et dépend du niveau de précision voulu. Il n'y a pas de modèle "100% valide". Un modèle est toujours une **approximation** de la réalité, construit en faisant des compromis.

Les modèles utiles sont aussi fréquemment modifiés avec le temps. On doit les concevoir et les programmer en tenant compte de cela.

Validité d'un modèle: doit reproduire les aspects importants du comportement du système **pour l'application visée**.

La notion de validité est relative et dépend du niveau de précision voulu. Il n'y a pas de modèle "100% valide". Un modèle est toujours une **approximation** de la réalité, construit en faisant des compromis.

Les modèles utiles sont aussi fréquemment modifiés avec le temps. On doit les concevoir et les programmer en tenant compte de cela.

Exemple: modélisation des activités d'un aéroport.

Validité d'un modèle: doit reproduire les aspects importants du comportement du système **pour l'application visée**.

La notion de validité est relative et dépend du niveau de précision voulu. Il n'y a pas de modèle "100% valide". Un modèle est toujours une **approximation** de la réalité, construit en faisant des compromis.

Les modèles utiles sont aussi fréquemment modifiés avec le temps. On doit les concevoir et les programmer en tenant compte de cela.

Exemple: modélisation des activités d'un aéroport.

On ne peut pas tout modéliser. Qu'est-ce qu'on veut étudier?

Validité d'un modèle: doit reproduire les aspects importants du comportement du système **pour l'application visée**.

La notion de validité est relative et dépend du niveau de précision voulu. Il n'y a pas de modèle "100% valide". Un modèle est toujours une **approximation** de la réalité, construit en faisant des compromis.

Les modèles utiles sont aussi fréquemment modifiés avec le temps. On doit les concevoir et les programmer en tenant compte de cela.

Exemple: modélisation des activités d'un aéroport.

On ne peut pas tout modéliser. Qu'est-ce qu'on veut étudier?

L'attente aux comptoirs d'enregistrement?

L'achalandage aux guichets des douanes?

Les retards des arrivées et départs?

Exemples de modèles simples

Un réseau d'activités stochastique.

Graphe acyclique $(\mathcal{N}, \mathcal{A})$, de noeuds \mathcal{N} et arcs \mathcal{A} (les activités).

Le graphe donne les relations de précédence entre les activités.

Chaque activité $k \in \mathcal{A}$ a une durée aléatoire V_k de fonction de répartition F_k .

Si V_k est la longueur de l'arc k , la durée minimale du projet est la longueur T du plus long chemin dans le réseau.

Simulation: On répète n fois: générer les V_k et calculer T .

Si on veut estimer $\mathbb{P}[T > x]$ pour un certain x , par exemple, alors on calcule la proportion des réalisations de T qui dépassent x .

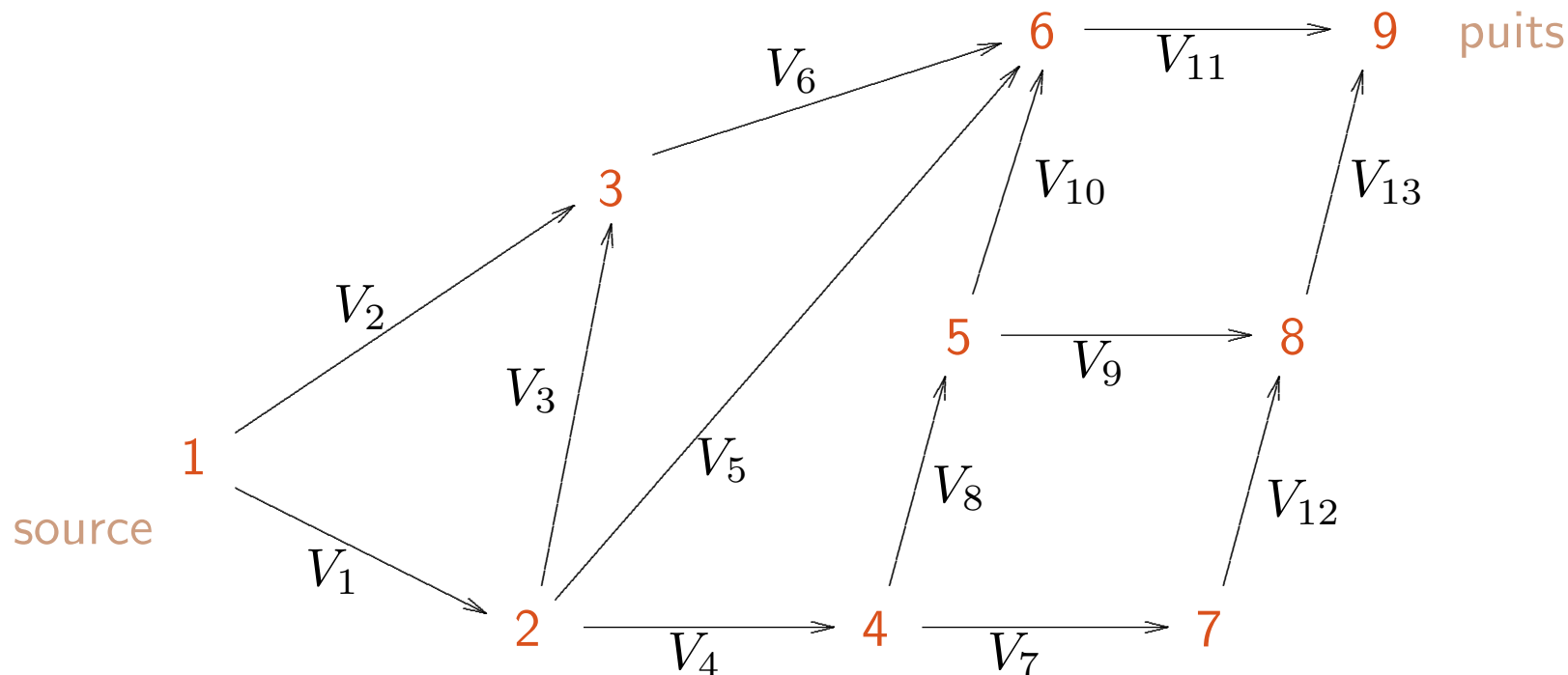


Illustration:

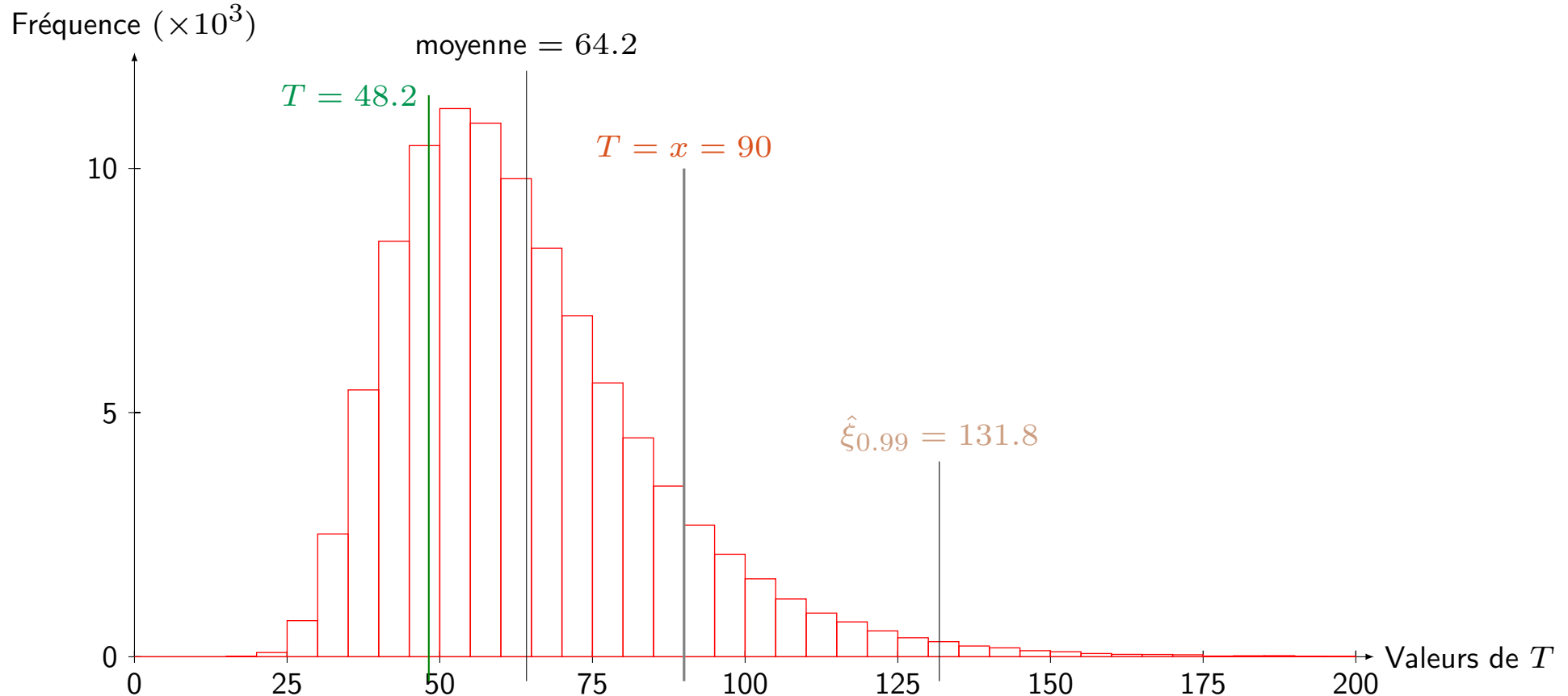
$V_j \sim N(\mu_j, \sigma_j^2)$ pour $j = 1, 2, 4, 11, 12$, et $V_j \sim \text{Exponentielle}(1/\mu_j)$ sinon.

μ_1, \dots, μ_{13} : 13.0, 5.5, 7.0, 5.2, 16.5, 14.7, 10.3, 6.0, 4.0, 20.0, 3.2, 3.2, 16.5.

Idée naïve: remplacer chaque V_j par son espérance. Donne $T = 48.2$.

On **simule** $n = 100\,000$ fois.

Valeurs de T (de 14.4 à 268.6; 11.57% dépassent $x = 90$):



Modèle de fiabilité statique.

Nombreuses applications (télécom., systèmes de production, militaires, etc.).

Un système a m composantes, dans l'état 0 (panne) ou 1 (fonctionne).

État du système: $\mathbf{Y} = (Y_1, \dots, Y_m)^t$

Fonction de structure: $\Phi : \{0, 1\}^m \rightarrow \{0, 1\}$.

Le système est en panne ssi $\Phi(\mathbf{Y}) = 0$.

fiabilité: $r = \mathbb{P}[\Phi(\mathbf{Y}) = 1]$.

Si les composantes sont indépendantes et $\mathbb{P}[Y_j = 1] = r_j$, alors

$$r = \sum_{\mathbf{Y} \in \mathcal{U}} \prod_{\{j: Y_j=1\}} r_j \prod_{\{j: Y_j=0\}} (1 - r_j),$$

où $\mathcal{U} = \{\mathbf{Y} : \Phi(\mathbf{Y}) = 1\}$.

Mais énumérer \mathcal{U} prend en général un temps exponentiel en m .

Modèle de fiabilité statique.

Nombreuses applications (télécom., systèmes de production, militaires, etc.).

Un système a m composantes, dans l'état 0 (panne) ou 1 (fonctionne).

État du système: $\mathbf{Y} = (Y_1, \dots, Y_m)^t$

Fonction de structure: $\Phi : \{0, 1\}^m \rightarrow \{0, 1\}$.

Le système est en panne ssi $\Phi(\mathbf{Y}) = 0$.

fiabilité: $r = \mathbb{P}[\Phi(\mathbf{Y}) = 1]$.

Si les composantes sont indépendantes et $\mathbb{P}[Y_j = 1] = r_j$, alors

$$r = \sum_{\mathbf{Y} \in \mathcal{U}} \prod_{\{j: Y_j=1\}} r_j \prod_{\{j: Y_j=0\}} (1 - r_j),$$

où $\mathcal{U} = \{\mathbf{Y} : \Phi(\mathbf{Y}) = 1\}$.

Mais énumérer \mathcal{U} prend en général un temps exponentiel en m .

Monte Carlo: Générer n copies i.i.d. de \mathbf{Y} , disons $\mathbf{Y}_1, \dots, \mathbf{Y}_n$, calculer $X_i = \Phi(\mathbf{Y}_i)$ pour chaque i , et estimer r par $\bar{X}_n = (X_1 + \dots + X_n)/n$.

On suppose ici que $\Phi(\mathbf{Y})$ est facilement calculable.

Souvent Φ est définie via un **graphe** $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ (noeuds et arêtes).
Chaque **arête** j relie deux **noeuds** et correspond à la composante j du système.
Elle est présente si la composante j fonctionne ($Y_j = 1$), et est absente sinon.
Soit $\mathcal{G}(\mathbf{Y})$ le sous-graphe de \mathcal{G} qui ne contient que les arêtes qui fonctionnent.

Souvent Φ est définie via un **graphe** $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ (noeuds et arêtes).
Chaque **arête** j relie deux **noeuds** et correspond à la composante j du système.
Elle est présente si la composante j fonctionne ($Y_j = 1$), et est absente sinon.
Soit $\mathcal{G}(\mathbf{Y})$ le sous-graphe de \mathcal{G} qui ne contient que les arêtes qui fonctionnent.

Parfois, $\Phi(\mathbf{Y}) = 1$ ssi les noeuds s_0 et s_1 sont reliés dans $\mathcal{G}(\mathbf{Y})$.

Plus général: $\Phi(\mathbf{Y}) = 1$ ssi tous les noeuds de $\mathcal{S} \subseteq \mathcal{N}$ sont reliés.

Facile à évaluer par des algorithmes de graphes (e.g., “minimal spanning tree”).

Souvent Φ est définie via un **graphe** $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ (noeuds et arêtes).
 Chaque **arête** j relie deux **noeuds** et correspond à la composante j du système.
 Elle est présente si la composante j fonctionne ($Y_j = 1$), et est absente sinon.
 Soit $\mathcal{G}(\mathbf{Y})$ le sous-graphe de \mathcal{G} qui ne contient que les arêtes qui fonctionnent.

Parfois, $\Phi(\mathbf{Y}) = 1$ ssi les noeuds s_0 et s_1 sont reliés dans $\mathcal{G}(\mathbf{Y})$.

Plus général: $\Phi(\mathbf{Y}) = 1$ ssi tous les noeuds de $\mathcal{S} \subseteq \mathcal{N}$ sont reliés.

Facile à évaluer par des algorithmes de graphes (e.g., “minimal spanning tree”).

Souvent, la **probabilité de panne** $1 - r$ est très proche de 0, et devient donc très difficile à estimer par MC. Par exemple, $1 - r = 10^{-10}$, le système sera en panne seulement une fois sur 10 milliards en moyenne. Trop coûteux d'estimer la prob. de panne par MC standard! Il faudra trouver des techniques plus efficaces pour cette situation d'**événements rares**.

Collisions dans un système de hachage.

Très grand ensemble d'identificateurs Φ , pas tous utilisés. Exemple: 48 bits.
Fonction de hachage $h : \Phi \rightarrow \{1, \dots, k\}$ (ensemble d'adresses), où $k \ll |\Phi|$.

Collisions dans un système de hachage.

Très grand ensemble d'identificateurs Φ , pas tous utilisés. Exemple: 48 bits.
Fonction de hachage $h : \Phi \rightarrow \{1, \dots, k\}$ (ensemble d'adresses), où $k \ll |\Phi|$.

On tire M éléments distincts de Φ , avec $M = \lambda_m$, ou encore $M \sim \text{Poisson}(\lambda_m)$.
 C = nombre de collisions dans les adresses (une variable aléatoire).

Modèle analytique simplifié:

Si k est grand et $\lambda = \lambda_m^2/(2k)$ est petit, alors $C \approx$ Poisson de moyenne λ :
 $\mathbb{P}[C = x] = e^{-\lambda} \lambda^x / x!$ pour $x = 0, 1, 2, \dots$

Modèle analytique simplifié:

Si k est grand et $\lambda = \lambda_m^2/(2k)$ est petit, alors $C \approx$ Poisson de moyenne λ :
 $\mathbb{P}[C = x] = e^{-\lambda} \lambda^x / x!$ pour $x = 0, 1, 2, \dots$

Simulation:

Si k est petit, ou si on veut évaluer la qualité de l'approx. par la loi de Poisson, on peut simuler le vrai modèle.

Modèle analytique simplifié:

Si k est grand et $\lambda = \lambda_m^2/(2k)$ est petit, alors $C \approx$ Poisson de moyenne λ :
 $\mathbb{P}[C = x] = e^{-\lambda} \lambda^x / x!$ pour $x = 0, 1, 2, \dots$

Simulation:

Si k est petit, ou si on veut évaluer la qualité de l'approx. par la loi de Poisson, on peut simuler le vrai modèle. On génère M v.a. i.i.d. uniformes sur $\{1, \dots, k\}$ et on compte le nombre C de collisions. On répète cela n fois et on peut examiner la répartition empirique des n valeurs de C , estimer $\mathbb{E}[C]$ par leur moyenne, etc.

Modèle analytique simplifié:

Si k est grand et $\lambda = \lambda_m^2/(2k)$ est petit, alors $C \approx$ Poisson de moyenne λ :
 $\mathbb{P}[C = x] = e^{-\lambda} \lambda^x / x!$ pour $x = 0, 1, 2, \dots$

Simulation:

Si k est petit, ou si on veut évaluer la qualité de l'approx. par la loi de Poisson, on peut simuler le vrai modèle. On génère M v.a. i.i.d. uniformes sur $\{1, \dots, k\}$ et on compte le nombre C de collisions. On répète cela n fois et on peut examiner la répartition empirique des n valeurs de C , estimer $\mathbb{E}[C]$ par leur moyenne, etc.

Illustration:

$k = 10\,000$ et $\lambda_m = 400$.

Formule analytique (approx.) $C \sim$ Poisson de moyenne $\lambda = m^2/(2k) = 8$.

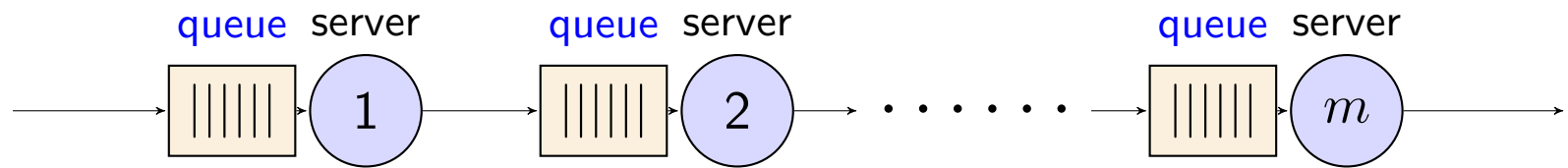
J'ai simulé de modèle $n = 10^7$ fois.

Moy. et variance empiriques:

7.87 et 7.47 pour M fixe, 7.89 et 8.10 pour $M \sim \text{Poisson}(400)$.

c	M fixé	$M \sim \text{Poisson}$	Approx. Poisson pour C
0	3181	4168	3354.6
1	25637	32257	26837.0
2	105622	122674	107348.0
3	288155	316532	286261.4
4	587346	614404	572522.8
5	948381	957951	916036.6
6	1269427	1247447	1221382.1
7	1445871	1397980	1395865.3
8	1434562	1377268	1395865.3
9	1251462	1207289	1240769.1
10	978074	958058	992615.3
11	688806	692416	721902.0
12	442950	459198	481268.0
13	260128	282562	296164.9
14	141467	162531	169237.1
15	71443	86823	90259.7
16	33224	43602	45129.9
17	14739	20827	21237.6
18	5931	9412	9438.9
19	2378	3985	3974.3
20	791	1629	1589.7
21	310	592	605.6
22	79	264	220.2
23	26	83	76.6
24	5	33	25.5
≥ 25	5	15	11.7

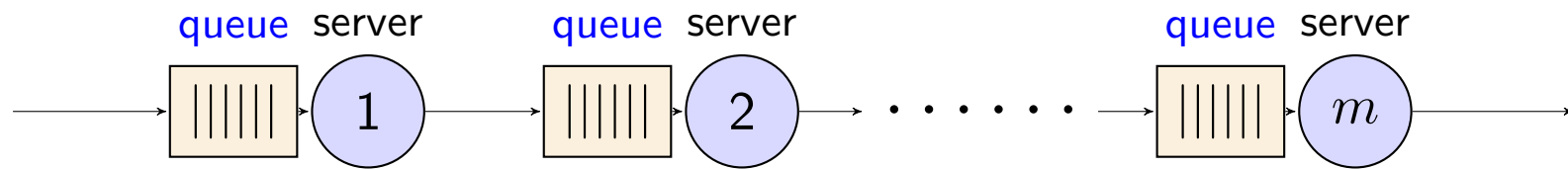
A tandem queue.



Station j : single **server** and FIFO **queue** with **capacity** c_j including service spot (at first queue, $c_1 = \infty$).

A customer cannot leave server j when queue $j + 1$ is full; it is **blocked**.

A tandem queue.



Station j : single **server** and FIFO **queue** with **capacity** c_j including service spot (at first queue, $c_1 = \infty$).

A customer cannot leave server j when queue $j + 1$ is full; it is **blocked**.

T_i = arrival time of customer i at first queue, for $i \geq 1$, and $T_0 = 0$;

$A_i = T_i - T_{i-1}$ = time between arrivals $i - 1$ and i ;

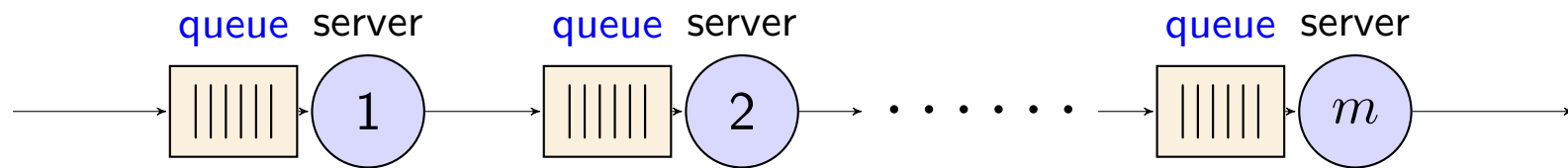
$W_{j,i}$ = **waiting time** in queue j for customer i ;

$S_{j,i}$ = **service time** at station j for customer i ;

$B_{j,i}$ = **blocked time** at station j for customer i ;

$D_{j,i}$ = **departure time** from station j for customer i .

A tandem queue.



Station j : single **server** and FIFO **queue** with **capacity** c_j including service spot (at first queue, $c_1 = \infty$).

A customer cannot leave server j when queue $j + 1$ is full; it is **blocked**.

T_i = arrival time of customer i at first queue, for $i \geq 1$, and $T_0 = 0$;

$A_i = T_i - T_{i-1}$ = time between arrivals $i - 1$ and i ;

$W_{j,i}$ = **waiting time** in queue j for customer i ;

$S_{j,i}$ = **service time** at station j for customer i ;

$B_{j,i}$ = **blocked time** at station j for customer i ;

$D_{j,i}$ = **departure time** from station j for customer i .

First customer arrives time $T_1 = A_1$, leaves station 1 at $D_{1,1} = T_1 + S_{1,1}$ and starts its service at station 2, leaves station 2 at time $D_{2,1} = D_{1,1} + S_{2,1}$ to join queue 3, and so on.

Customer 2 arrives at time $T_2 = T_1 + A_2$, starts its service at queue 1 at time $\max(T_2, D_{1,1})$, leaves station 1 at time $\max(T_2, D_{1,1}) + S_{1,2}$ if there is space available at station 2, etc.

Suppose we can generate successive interarrival times A_i and service times $S_{j,i}$. Then we can compute the T_i 's easily and the departure times $D_{j,i}$ via:

$$D_{j,i} = \max[D_{j-1,i} + S_{j,i}, D_{j,i-1} + S_{j,i}, D_{j+1,i-c_{j+1}}]$$

for $1 \leq j \leq m$ and $i \geq 1$, where $D_{0,i} = T_i$, $D_{j,i} = 0$ for $i \leq 0$, and $D_{m+1,i} = 0$ for all i . Then:

$$W_{j,i} = \max[0, D_{j,i-1} - D_{j-1,i}],$$

$$B_{j,i} = D_{j,i} - D_{j-1,i} - W_{j,i} - S_{j,i}.$$

Suppose we can generate successive interarrival times A_i and service times $S_{j,i}$. Then we can compute the T_i 's easily and the departure times $D_{j,i}$ via:

$$D_{j,i} = \max[D_{j-1,i} + S_{j,i}, D_{j,i-1} + S_{j,i}, D_{j+1,i-c_{j+1}}]$$

for $1 \leq j \leq m$ and $i \geq 1$, where $D_{0,i} = T_i$, $D_{j,i} = 0$ for $i \leq 0$, and $D_{m+1,i} = 0$ for all i . Then:

$$\begin{aligned} W_{j,i} &= \max[0, D_{j,i-1} - D_{j-1,i}], \\ B_{j,i} &= D_{j,i} - D_{j-1,i} - W_{j,i} - S_{j,i}. \end{aligned}$$

For **infinite buffer sizes** (no blocking), we always have $B_{j,i} = 0$ and

$$D_{j,i} = \max[D_{j-1,i}, D_{j,i-1}] + S_{j,i}.$$

Suppose we can generate successive interarrival times A_i and service times $S_{j,i}$. Then we can compute the T_i 's easily and the departure times $D_{j,i}$ via:

$$D_{j,i} = \max[D_{j-1,i} + S_{j,i}, D_{j,i-1} + S_{j,i}, D_{j+1,i-c_{j+1}}]$$

for $1 \leq j \leq m$ and $i \geq 1$, where $D_{0,i} = T_i$, $D_{j,i} = 0$ for $i \leq 0$, and $D_{m+1,i} = 0$ for all i . Then:

$$\begin{aligned} W_{j,i} &= \max[0, D_{j,i-1} - D_{j-1,i}], \\ B_{j,i} &= D_{j,i} - D_{j-1,i} - W_{j,i} - S_{j,i}. \end{aligned}$$

For **infinite buffer sizes** (no blocking), we always have $B_{j,i} = 0$ and

$$D_{j,i} = \max[D_{j-1,i}, D_{j,i-1}] + S_{j,i}.$$

A **single queue** with no blocking: $GI/GI/1$ model, obeys **Lindley recurrence**:

$$W_{1,i} = \max[0, D_{1,i-1} - T_i] = \max[0, W_{1,i-1} + S_{1,i-1} - A_i].$$

What we just saw is **production blocking**. In **communication blocking**, service at station j starts only when queue $j + 1$ is not full. Then:

$$D_{j,i} = \max[D_{j-1,i}, D_{j,i-1}, D_{j+1,i-c_{j+1}}] + S_{j,i}.$$

What we just saw is **production blocking**. In **communication blocking**, service at station j starts only when queue $j + 1$ is not full. Then:

$$D_{j,i} = \max[D_{j-1,i}, D_{j,i-1}, D_{j+1,i-c_{j+1}}] + S_{j,i}.$$

Algorithm: Simulating N_c customers with production blocking.

Let $T_0 = 0$ and $D_{0,0} = D_{1,0} = 0$;

For $j = 2, \dots, m$,

 For $i = -c_j + 1, \dots, 0$, let $D_{j,i} = 0$;

For $i = 1, \dots, N_c$,

 Generate A_i from its distribution and let $D_{0,i} = T_i = T_{i-1} + A_i$;

 Let $W_i = B_i = 0$;

 For $j = 1, \dots, m$,

 Generate $S_{j,i}$ from its distribution;

 Let $D_{j,i} = \max[D_{j-1,i} + S_{j,i}, D_{j,i-1} + S_{j,i}, D_{j+1,i-c_{j+1}}]$;

 Let $W_{j,i} = \max[0, D_{j,i-1} - D_{j-1,i}]$ and $W_i = W_i + W_{j,i}$;

 Let $B_{j,i} = D_{j,i} - D_{j-1,i} - W_{j,i} - S_{j,i}$ and $B_i = B_i + B_{j,i}$;

Compute and return the averages

$$\bar{W}_{N_c} = (W_1 + \dots + W_{N_c})/N_c \text{ and } \bar{B}_{N_c} = (B_1 + \dots + B_{N_c})/N_c.$$

What we just saw is **production blocking**. In **communication blocking**, service at station j starts only when queue $j + 1$ is not full. Then:

$$D_{j,i} = \max[D_{j-1,i}, D_{j,i-1}, D_{j+1,i-c_{j+1}}] + S_{j,i}.$$

Algorithm: Simulating N_c customers with production blocking.

Let $T_0 = 0$ and $D_{0,0} = D_{1,0} = 0$;

For $j = 2, \dots, m$,

 For $i = -c_j + 1, \dots, 0$, let $D_{j,i} = 0$;

For $i = 1, \dots, N_c$,

 Generate A_i from its distribution and let $D_{0,i} = T_i = T_{i-1} + A_i$;

 Let $W_i = B_i = 0$;

 For $j = 1, \dots, m$,

 Generate $S_{j,i}$ from its distribution;

 Let $D_{j,i} = \max[D_{j-1,i} + S_{j,i}, D_{j,i-1} + S_{j,i}, D_{j+1,i-c_{j+1}}]$;

 Let $W_{j,i} = \max[0, D_{j,i-1} - D_{j-1,i}]$ and $W_i = W_i + W_{j,i}$;

 Let $B_{j,i} = D_{j,i} - D_{j-1,i} - W_{j,i} - S_{j,i}$ and $B_i = B_i + B_{j,i}$;

Compute and return the averages

$$\bar{W}_{N_c} = (W_1 + \dots + W_{N_c})/N_c \text{ and } \bar{B}_{N_c} = (B_1 + \dots + B_{N_c})/N_c.$$

What if we want to simulate all customers who arrive before time T ?

Replace “For $i = 1, \dots, N_c$ ” by “For ($i = 1, T_i < T, i++$)”.

Mouvement Brownien géométrique et option financière.

Modèle de Black et Scholes: sous la mesure de risque neutre, la valeur au marché d'une action au temps t est

$$S(t) = S(0)e^{(r-\sigma^2/2)t+\sigma B(t)}$$

où r est le taux d'intérêt sans risque, σ est la volatilité, et $B(\cdot)$ est un mouvement Brownien standard.

Mouvement Brownien géométrique et option financière.

Modèle de Black et Scholes: sous la mesure de risque neutre, la valeur au marché d'une action au temps t est

$$S(t) = S(0)e^{(r-\sigma^2/2)t+\sigma B(t)}$$

où r est le taux d'intérêt sans risque, σ est la volatilité, et $B(\cdot)$ est un mouvement Brownien standard.

Pour $t_2 > t_1 \geq 0$, $B(t_2) - B(t_1)$ est une v.a. normale de moyenne 0 et variance $t_2 - t_1$, indépendante de $B(\cdot)$ en dehors de $[t_1, t_2]$.

Mouvement Brownien géométrique et option financière.

Modèle de Black et Scholes: sous la mesure de risque neutre, la valeur au marché d'une action au temps t est

$$S(t) = S(0)e^{(r-\sigma^2/2)t+\sigma B(t)}$$

où r est le taux d'intérêt sans risque, σ est la volatilité, et $B(\cdot)$ est un mouvement Brownien standard.

Pour $t_2 > t_1 \geq 0$, $B(t_2) - B(t_1)$ est une v.a. normale de moyenne 0 et variance $t_2 - t_1$, indépendante de $B(\cdot)$ en dehors de $[t_1, t_2]$.

Supposons qu'un contrat financier nous assure un revenu $g(S(t_1), \dots, S(t_d))$ au temps $T \geq t_d$, où t_1, \dots, t_d sont des instants d'observation fixes.

La juste valeur au temps 0 d'un tel contrat, si $S(0) = s$, est

$$v(s, T) = \mathbb{E} \left[e^{-rT} g(S(t_1), \dots, S(t_d)) \mid S(0) = s \right],$$

où \mathbb{E} est l'espérance sous la mesure de risque neutre et e^{-rT} est le facteur d'actualisation.

Si g est simple, on peut disposer d'une formule analytique pour $v(s, T)$.
Par exemple, pour une **option d'achat européenne** (sans dividende),

$$g(S(T)) = \max[0, S(T) - K],$$

où T est la date d'expiration et K est le prix d'achat au contrat.

Si g est simple, on peut disposer d'une formule analytique pour $v(s, T)$.
Par exemple, pour une **option d'achat européenne** (sans dividende),

$$g(S(T)) = \max[0, S(T) - K],$$

où T est la date d'expiration et K est le prix d'achat au contrat.

Si $S(T) > K$, le détenteur achète l'action au prix K et revend immédiatement au prix du marché $S(T)$. Sinon il déchire le contrat.

Si g est simple, on peut disposer d'une formule analytique pour $v(s, T)$.
Par exemple, pour une **option d'achat européenne** (sans dividende),

$$g(S(T)) = \max[0, S(T) - K],$$

où T est la date d'expiration et K est le prix d'achat au contrat.

Si $S(T) > K$, le détenteur achète l'action au prix K et revend immédiatement au prix du marché $S(T)$. Sinon il déchire le contrat.

Pour ce cas, la célèbre **formule de Black-Scholes** donne:

$$v(s, T) = s\Phi(-z_0 + \sigma\sqrt{T}) - Ke^{-rT}\Phi(-z_0),$$

où Φ est la f.r. normale standard et

$$z_0 = \frac{\ln(K/s) - (r - \sigma^2/2)T}{\sigma\sqrt{T}}.$$

Si g est plus compliqué: pas de formule.

Exemple: **option asiatique** à observations discrètes (Bermudienne):

$$g(S(t_1), \dots, S(t_d)) = \max \left(0, \frac{1}{d} \sum_{j=1}^d S(t_j) - K \right).$$

Si g est plus compliqué: pas de formule.

Exemple: **option asiatique** à observations discrètes (Bermudienne):

$$g(S(t_1), \dots, S(t_d)) = \max \left(0, \frac{1}{d} \sum_{j=1}^d S(t_j) - K \right).$$

On peut alors estimer $v(s, T)$ par **simulation**:

Pour $i = 1, \dots, n$,

Poser $B(t_0) = t_0 = 0$;

Pour $j = 1, \dots, d$,

Générer $Z_j \sim N(0, 1)$;

Poser $B(t_j) = B(t_{j-1}) + \sqrt{t_j - t_{j-1}} Z_j$

et $S(t_j) = S(0) \exp \left[(r - \sigma^2/2)t_j + \sigma B(t_j) \right]$;

Calculer le revenu $X_i = g(S(t_1), \dots, S(t_d))$;

Calculer \bar{X}_n , la moyenne des X_i , et multiplier par e^{-rT} .

Si g est plus compliqué: pas de formule.

Exemple: **option asiatique** à observations discrètes (Bermudienne):

$$g(S(t_1), \dots, S(t_d)) = \max \left(0, \frac{1}{d} \sum_{j=1}^d S(t_j) - K \right).$$

On peut alors estimer $v(s, T)$ par **simulation**:

Pour $i = 1, \dots, n$,

Poser $B(t_0) = t_0 = 0$;

Pour $j = 1, \dots, d$,

Générer $Z_j \sim N(0, 1)$;

Poser $B(t_j) = B(t_{j-1}) + \sqrt{t_j - t_{j-1}} Z_j$

et $S(t_j) = S(0) \exp \left[(r - \sigma^2/2)t_j + \sigma B(t_j) \right]$;

Calculer le revenu $X_i = g(S(t_1), \dots, S(t_d))$;

Calculer \bar{X}_n , la moyenne des X_i , et multiplier par e^{-rT} .

Raffinements: volatilité stochastique, sauts pour $S(\cdot)$, $S(t)$ est un vecteur, etc.

Illustration: Option asiatique Bermudienne avec:

$d = 12$, $T = 1$ (année), $t_j = j/12$ pour $j = 0, \dots, 12$, $K = 100$, $s_0 = 100$,
 $r = 0.05$, $\sigma = 0.5$,

On a fait $n = 10^6$ simulations indépendantes. Moyenne: 13.1. Max = 390.8

Dans 53.47% des cas, le revenu était 0.

Histogramme des valeurs positives:



Estimation d'un quantile.

Ce que l'on veut estimer n'est pas toujours une espérance mathématique.

Estimation d'un quantile.

Ce que l'on veut estimer n'est pas toujours une espérance mathématique.

Par exemple, soit Y la **perte nette** d'un investissement donné durant une période fixée.

Estimation d'un quantile.

Ce que l'on veut estimer n'est pas toujours une espérance mathématique.

Par exemple, soit Y la **perte nette** d'un investissement donné durant une période fixée. Pour p fixé (e.g., $p = 0.99$), on veut estimer la constante

$$\xi_p = \inf\{y : \mathbb{P}[Y \leq y] \geq p\}.$$

Cette constante est le p -quantile de la distribution de Y .

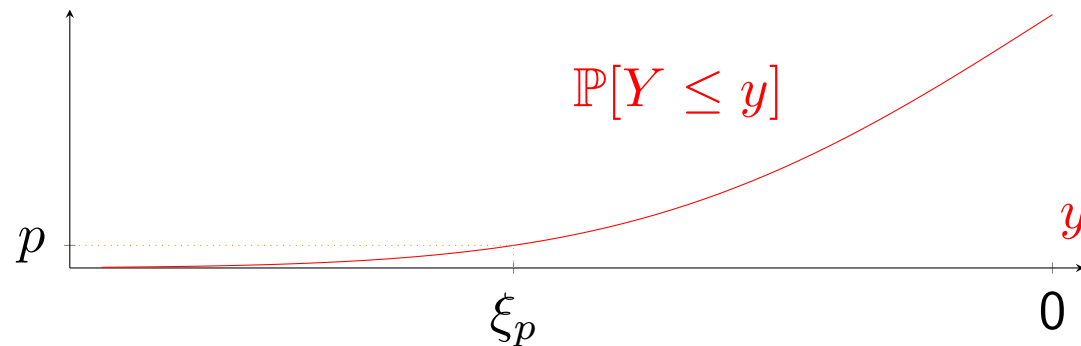
Estimation d'un quantile.

Ce que l'on veut estimer n'est pas toujours une espérance mathématique.

Par exemple, soit Y la **perte nette** d'un investissement donné durant une période fixée. Pour p fixé (e.g., $p = 0.99$), on veut estimer la constante

$$\xi_p = \inf\{y : \mathbb{P}[Y \leq y] \geq p\}.$$

Cette constante est le p -quantile de la distribution de Y .



Dans ce contexte, ξ_p s'appelle la **p -valeur à risque** (p -VaR).

Interprétation: la probabilité de perdre ξ_p ou plus ne dépasse pas $1 - p$.

Si la loi de Y est complexe, on ne peut pas calculer ξ_p exactement, mais on peut simuler n copies de Y , disons Y_1, \dots, Y_n , et estimer ξ_p par

$$\hat{\xi}_{p,n} = \inf \left\{ y \in \mathbb{R} : \frac{1}{n} \sum_{i=1}^n \mathbb{I}[Y_i \leq y] \geq p \right\} = Y_{(\lceil np \rceil)}.$$

Cet estimateur est biaisé pour n fini, mais fortement consistant.

Si la loi de Y est complexe, on ne peut pas calculer ξ_p exactement, mais on peut simuler n copies de Y , disons Y_1, \dots, Y_n , et estimer ξ_p par

$$\hat{\xi}_{p,n} = \inf \left\{ y \in \mathbb{R} : \frac{1}{n} \sum_{i=1}^n \mathbb{I}[Y_i \leq y] \geq p \right\} = Y_{(\lceil np \rceil)}.$$

Cet estimateur est biaisé pour n fini, mais fortement consistant.

Exemple: Pour le réseau d'activités stochastique, pour $p = 0.99$ on a obtenu $\hat{\xi}_{0.99,n} = 131.8$. On a donc $\mathbb{P}[T > 131.8] \approx 0.01$.

Nous reviendrons plus tard sur l'estimation de quantiles.

Modèle de choix discrets

On veut modéliser comment un individu choisit parmi plusieurs alternatives.

Un modèle **logit mixte multinomial**: L'utilité de l'alternative j pour l'individu q est

$$U_{q,j} = \beta_q^t \mathbf{x}_{q,j} + \epsilon_{q,j} = \sum_{\ell=1}^d \beta_{q,\ell} x_{q,j,\ell} + \epsilon_{q,j}, \text{ où}$$

$\beta_q^t = (\beta_{q,1}, \dots, \beta_{q,d})$ indique les **goûts** de l'individu q ,

$\mathbf{x}_{q,j}^t = (x_{q,j,1}, \dots, x_{q,j,d})$ les **attributs** du choix j pour l'individu q ,

$\epsilon_{q,j}$ bruit; v.a. indép. Gumbel de moyenne 0 et paramètre d'échelle $\beta = 1$.

L'individu q choisit l'alternative ayant la plus grande utilité $U_{q,j}$.

On peut observer les $\mathbf{x}_{q,j}$ et les choix y_q des individus, mais pas le reste.

Modèle de choix discrets

On veut modéliser comment un individu choisit parmi plusieurs alternatives.

Un modèle **logit mixte multinomial**: L'utilité de l'alternative j pour l'individu q est

$$U_{q,j} = \beta_q^t \mathbf{x}_{q,j} + \epsilon_{q,j} = \sum_{\ell=1}^d \beta_{q,\ell} x_{q,j,\ell} + \epsilon_{q,j}, \text{ où}$$

$\beta_q^t = (\beta_{q,1}, \dots, \beta_{q,d})$ indique les **goûts** de l'individu q ,

$\mathbf{x}_{q,j}^t = (x_{q,j,1}, \dots, x_{q,j,d})$ les **attributs** du choix j pour l'individu q ,

$\epsilon_{q,j}$ bruit; v.a. indép. Gumbel de moyenne 0 et paramètre d'échelle $\beta = 1$.

L'individu q choisit l'alternative ayant la plus grande utilité $U_{q,j}$.

On peut observer les $\mathbf{x}_{q,j}$ et les choix y_q des individus, mais pas le reste.

On suppose que pour un individu choisi au hasard, β_q est un vecteur aléatoire de densité f_{θ} , qui dépend d'un vecteur de paramètres θ .

Pour β_q fixé, on peut montrer que la probabilité de choisir l'alternative j est

$$L_q(j, \beta_q) = \frac{\exp[\beta_q^t \mathbf{x}_{q,j}]}{\sum_{a \in \mathcal{A}(q)} \exp[\beta_q^t \mathbf{x}_{q,a}]},$$

où $\mathcal{A}(q)$ est l'ensemble des alternatives disponibles. C'est le [modèle logit](#).

Pour β_q fixé, on peut montrer que la probabilité de choisir l'alternative j est

$$L_q(j, \beta_q) = \frac{\exp[\beta_q^t \mathbf{x}_{q,j}]}{\sum_{a \in \mathcal{A}(q)} \exp[\beta_q^t \mathbf{x}_{q,a}]},$$

où $\mathcal{A}(q)$ est l'ensemble des alternatives disponibles. C'est le [modèle logit](#).

La probabilité (non conditionnelle) de choisir j est alors

$$p_q(j, \theta) = \int L_q(j, \beta) f_{\theta}(\beta) d\beta.$$

Pour β_q fixé, on peut montrer que la probabilité de choisir l'alternative j est

$$L_q(j, \beta_q) = \frac{\exp[\beta_q^t \mathbf{x}_{q,j}]}{\sum_{a \in \mathcal{A}(q)} \exp[\beta_q^t \mathbf{x}_{q,a}]},$$

où $\mathcal{A}(q)$ est l'ensemble des alternatives disponibles. C'est le **modèle logit**.

La probabilité (non conditionnelle) de choisir j est alors

$$p_q(j, \theta) = \int L_q(j, \beta) f_{\theta}(\beta) d\beta.$$

Supposons que l'on a des données (les $\mathbf{x}_{q,j}$ pour chaque j et le choix y_q) pour m individus ($q = 1, \dots, m$) et on veut estimer θ .

Méthode standard: **vraisemblance maximale**. On maximise (le logarithme de) la probabilité conjointe $L(\boldsymbol{\theta})$ (la vraisemblance) de l'échantillon en fonction de θ :

$$\ln L(\boldsymbol{\theta}) = \ln \prod_{q=1}^m p_q(y_q, \boldsymbol{\theta}) = \sum_{q=1}^m \ln p_q(y_q, \boldsymbol{\theta}).$$

Méthode standard: **vraisemblance maximale**. On maximise (le logarithme de) la probabilité conjointe $L(\boldsymbol{\theta})$ (la vraisemblance) de l'échantillon en fonction de θ :

$$\ln L(\boldsymbol{\theta}) = \ln \prod_{q=1}^m p_q(y_q, \boldsymbol{\theta}) = \sum_{q=1}^m \ln p_q(y_q, \boldsymbol{\theta}).$$

Mais en général, on n'a pas de formule explicite pour $p_q(j, \boldsymbol{\theta})$, et on utilise Monte Carlo pour évaluer l'intégrale, pour chaque q .

Pour $\boldsymbol{\theta}$ fixé, on génère n réalisations indépendantes de β selon $f_{\boldsymbol{\theta}}$, disons $\beta_q^{(1)}(\boldsymbol{\theta}), \dots, \beta_q^{(n)}(\boldsymbol{\theta})$, et on estime $\ln L(\boldsymbol{\theta})$ par

$$\ln(\hat{L}(\boldsymbol{\theta})) = \sum_{q=1}^m \ln \left(\frac{1}{n} \sum_{i=1}^n L_q(j, \beta_q^{(i)}(\boldsymbol{\theta})) \right).$$

Méthode standard: **vraisemblance maximale**. On maximise (le logarithme de) la probabilité conjointe $L(\boldsymbol{\theta})$ (la vraisemblance) de l'échantillon en fonction de $\boldsymbol{\theta}$:

$$\ln L(\boldsymbol{\theta}) = \ln \prod_{q=1}^m p_q(y_q, \boldsymbol{\theta}) = \sum_{q=1}^m \ln p_q(y_q, \boldsymbol{\theta}).$$

Mais en général, on n'a pas de formule explicite pour $p_q(j, \boldsymbol{\theta})$, et on utilise Monte Carlo pour évaluer l'intégrale, pour chaque q .

Pour $\boldsymbol{\theta}$ fixé, on génère n réalisations indépendantes de β selon $f_{\boldsymbol{\theta}}$, disons $\beta_q^{(1)}(\boldsymbol{\theta}), \dots, \beta_q^{(n)}(\boldsymbol{\theta})$, et on estime $\ln L(\boldsymbol{\theta})$ par

$$\ln(\hat{L}(\boldsymbol{\theta})) = \sum_{q=1}^m \ln \left(\frac{1}{n} \sum_{i=1}^n L_q(j, \beta_q^{(i)}(\boldsymbol{\theta})) \right).$$

La prochaine étape consiste à maximiser $\ln(\hat{L}(\boldsymbol{\theta}))$ en fonction de $\boldsymbol{\theta}$.

Le vecteur $\hat{\boldsymbol{\theta}}$ qui maximise $\ln(\hat{L}(\boldsymbol{\theta}))$ est un estimateur (biaisé) de celui qui maximise $\ln(L(\boldsymbol{\theta}))$. Dans cet exemple, on n'estime pas une espérance, mais un optimum.

Génération de V.A. Uniformes

Qu'est-ce qu'on veut?

Produire des suites de nombres qui ont l'air d'être choisis complètement au hasard:

- suites de bits: 011010100110110101001101100101000111...
- suite d'entiers de 0 à 100: 31, 83, 02, 72, 54, 26, ...
- suites d'entiers de 1 à n , 1 à $n - 1$, 1 à $n - 2$, ... (permutation aléatoire)
- suite de nombres réels entre 0 et 1.

Génération de V.A. Uniformes

Qu'est-ce qu'on veut?

Produire des suites de nombres qui ont l'air d'être choisis complètement au hasard:

- suites de bits: 011010100110110101001101100101000111...
- suite d'entiers de 0 à 100: 31, 83, 02, 72, 54, 26, ...
- suites d'entiers de 1 à n , 1 à $n - 1$, 1 à $n - 2$, ... (permutation aléatoire)
- suite de nombres réels entre 0 et 1.

Mécanismes physiques: lancer les dés, bouliers, roulettes,

bruit thermique dans les résistances de circuits électroniques,
capteurs de radiations, autres mécanismes basés sur la physique quantique,
microsecondes de l'horloge de l'ordinateur, ou d'un temps d'accès au disque, etc.

Génération de V.A. Uniformes

Qu'est-ce qu'on veut?

Produire des suites de nombres qui ont l'air d'être choisis complètement au hasard:

- suites de bits: 011010100110110101001101100101000111...
- suite d'entiers de 0 à 100: 31, 83, 02, 72, 54, 26, ...
- suites d'entiers de 1 à n , 1 à $n - 1$, 1 à $n - 2$, ... (permutation aléatoire)
- suite de nombres réels entre 0 et 1.

Mécanismes physiques: lancer les dés, bouliers, roulettes,

bruit thermique dans les résistances de circuits électroniques,
capteurs de radiations, autres mécanismes basés sur la physique quantique,
microsecondes de l'horloge de l'ordinateur, ou d'un temps d'accès au disque, etc.

Contient de la vraie entropie (incertitude), mais encombrant, pas facilement reproductible, pas toujours fiable, peu ou pas d'analyse mathématique possible.

Certains de ces mécanismes sont brevetés.

Plusieurs sont disponibles commercialement.

Pour améliorer les propriétés statistiques: combiner des blocs de bits (XOR).

Générateurs algorithmiques (ou pseudo-aléatoires) (GPA). Une fois les paramètres et l'état initial du GPA choisis, la suite produite est complètement déterministe.

Générateurs algorithmiques (ou pseudo-aléatoires) (GPA). Une fois les paramètres et l'état initial du GPA choisis, la suite produite est complètement déterministe.

Avantages: pas de matériel à installer, un logiciel suffit; souvent plus rapide; on peut facilement répéter la même séquence.

Générateurs algorithmiques (ou pseudo-aléatoires) (GPA). Une fois les paramètres et l'état initial du GPA choisis, la suite produite est complètement déterministe.

Avantages: pas de matériel à installer, un logiciel suffit; souvent plus rapide; on peut facilement répéter la même séquence.

Qualités requises? Dépend des applications.

1. Jeux d'ordinateurs personnels: L'apparence suffit.

Générateurs algorithmiques (ou pseudo-aléatoires) (GPA). Une fois les paramètres et l'état initial du GPA choisis, la suite produite est complètement déterministe.

Avantages: pas de matériel à installer, un logiciel suffit; souvent plus rapide; on peut facilement répéter la même séquence.

Qualités requises? Dépend des applications.

1. Jeux d'ordinateurs personnels: L'apparence suffit.

2. Simulation stochastique (Monte Carlo):

On veut que les propriétés statistiques du modèle mathématique soient bien reproduites par le simulateur.

Générateurs algorithmiques (ou pseudo-aléatoires) (GPA). Une fois les paramètres et l'état initial du GPA choisis, la suite produite est complètement déterministe.

Avantages: pas de matériel à installer, un logiciel suffit; souvent plus rapide; on peut facilement répéter la même séquence.

Qualités requises? Dépend des applications.

1. Jeux d'ordinateurs personnels: L'apparence suffit.

2. Simulation stochastique (Monte Carlo):

On veut que les propriétés statistiques du modèle mathématique soient bien reproduites par le simulateur.

3. Loteries, machines de casinos, casinos sur Internet, ...

Il ne faut pas que quiconque puisse obtenir un avantage pour inférer les prochains numéros ou encore des combinaisons plus probables. Conditions plus exigeantes que pour la simulation.

Générateurs algorithmiques (ou pseudo-aléatoires) (GPA). Une fois les paramètres et l'état initial du GPA choisis, la suite produite est complètement déterministe.

Avantages: pas de matériel à installer, un logiciel suffit; souvent plus rapide; on peut facilement répéter la même séquence.

Qualités requises? Dépend des applications.

1. Jeux d'ordinateurs personnels: L'apparence suffit.

2. Simulation stochastique (Monte Carlo):

On veut que les propriétés statistiques du modèle mathématique soient bien reproduites par le simulateur.

3. Loteries, machines de casinos, casinos sur Internet, ...

Il ne faut pas que quiconque puisse obtenir un avantage pour inférer les prochains numéros ou encore des combinaisons plus probables. Conditions plus exigeantes que pour la simulation.

4. Cryptologie: Encore plus exigeant. L'observation d'une partie de l'output ne doit nous aider d'aucune manière à deviner quoi que ce soit dans le reste.

Besoins pour la Simulation Stochastique

On utilise habituellement un GPA qui **imite** une suite U_0, U_1, U_2, \dots de variables aléatoires indépendantes de loi uniforme sur l'intervalle $(0, 1)$.

Besoins pour la Simulation Stochastique

On utilise habituellement un GPA qui **imite** une suite U_0, U_1, U_2, \dots de variables aléatoires indépendantes de loi uniforme sur l'intervalle $(0, 1)$.

Pour générer des v.a. selon d'autres lois, on applique des transformations à ces U_j .

Besoins pour la Simulation Stochastique

On utilise habituellement un GPA qui **imite** une suite U_0, U_1, U_2, \dots de variables aléatoires indépendantes de loi uniforme sur l'intervalle $(0, 1)$.

Pour générer des v.a. selon d'autres lois, on applique des transformations à ces U_j .

Comparaison de systèmes semblables avec valeurs aléatoires communes.

On simule un réseau de communication, ou un centre d'appels téléphoniques, ou un réseau de distribution de biens, ou une usine, ou le trafic automobile dans une ville, ou la gestion dynamique d'un portefeuille d'investissements (finance), etc.

On veut **comparer** deux configurations (ou politiques de gestion) semblables du système.

Une partie de la **différence** de performance sera due à la différence de configuration, et une autre partie sera due au bruit stochastique. On veut minimiser cette seconde partie.

Idée de base: simuler les deux configurations avec les mêmes valeurs uniformes U_j , utilisées exactement aux mêmes endroits.

On verra plus tard des résultats théoriques sur l'amélioration d'efficacité (réduction de variance) que cela apporte.

Mais l'implantation, avec synchronisation des v.a., peut être compliquée lorsque les deux configurations n'utilisent pas le même nombre de U_j (e.g., parfois on doit générer une v.a. dans un cas et pas dans l'autre).

Idée de base: simuler les deux configurations avec les mêmes valeurs uniformes U_j , utilisées exactement aux mêmes endroits.

On verra plus tard des résultats théoriques sur l'amélioration d'efficacité (réduction de variance) que cela apporte.

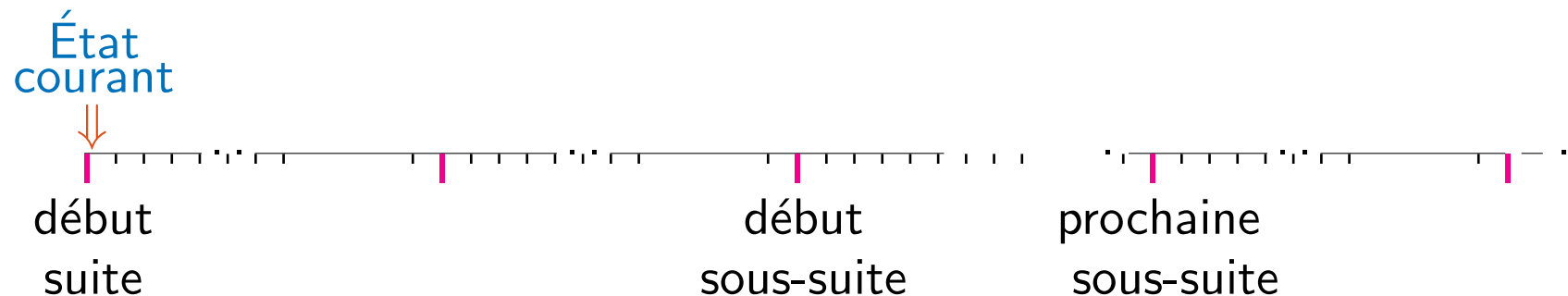
Mais l'implantation, avec synchronisation des v.a., peut être compliquée lorsque les deux configurations n'utilisent pas le même nombre de U_j (e.g., parfois on doit générer une v.a. dans un cas et pas dans l'autre).

Une solution: GPA avec **suites et sous-suites multiples**.

Chaque **suite** peut être vue comme un GPA virtuel.

Elle est partitionnée en **sous-suites**.

On peut créer autant de suites (distinctes et "indépendantes") que l'on veut.



Idée de base: simuler les deux configurations avec les mêmes valeurs uniformes U_j , utilisées exactement aux mêmes endroits.

On verra plus tard des résultats théoriques sur l'amélioration d'efficacité (réduction de variance) que cela apporte.

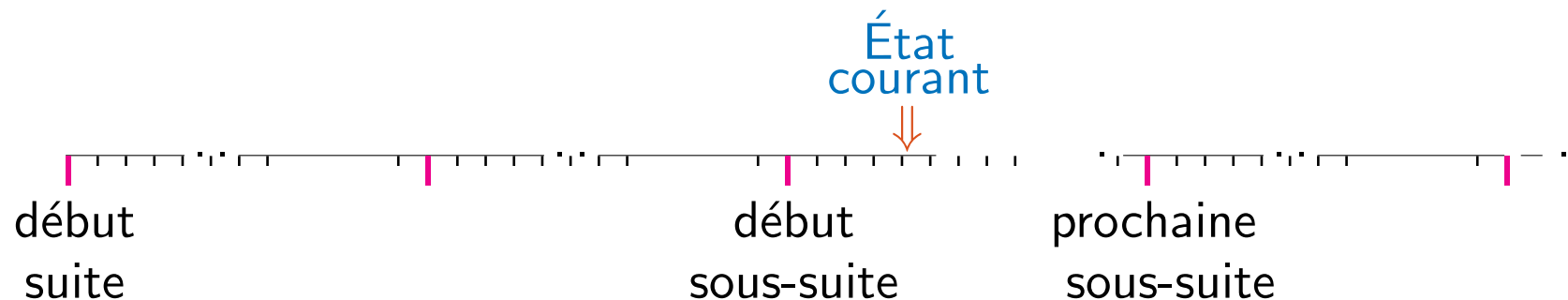
Mais l'implantation, avec synchronisation des v.a., peut être compliquée lorsque les deux configurations n'utilisent pas le même nombre de U_j (e.g., parfois on doit générer une v.a. dans un cas et pas dans l'autre).

Une solution: GPA avec **suites et sous-suites multiples**.

Chaque **suite** peut être vue comme un GPA virtuel.

Elle est partitionnée en **sous-suites**.

On peut créer autant de suites (distinctes et "indépendantes") que l'on veut.



Idée de base: simuler les deux configurations avec les mêmes valeurs uniformes U_j , utilisées exactement aux mêmes endroits.

On verra plus tard des résultats théoriques sur l'amélioration d'efficacité (réduction de variance) que cela apporte.

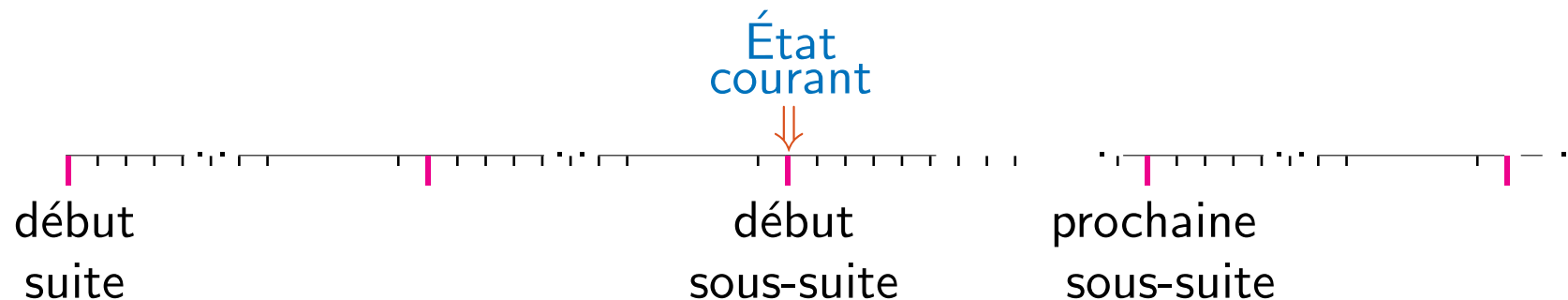
Mais l'implantation, avec synchronisation des v.a., peut être compliquée lorsque les deux configurations n'utilisent pas le même nombre de U_j (e.g., parfois on doit générer une v.a. dans un cas et pas dans l'autre).

Une solution: GPA avec **suites et sous-suites multiples**.

Chaque **suite** peut être vue comme un GPA virtuel.

Elle est partitionnée en **sous-suites**.

On peut créer autant de suites (distinctes et "indépendantes") que l'on veut.



Idée de base: simuler les deux configurations avec les mêmes valeurs uniformes U_j , utilisées exactement aux mêmes endroits.

On verra plus tard des résultats théoriques sur l'amélioration d'efficacité (réduction de variance) que cela apporte.

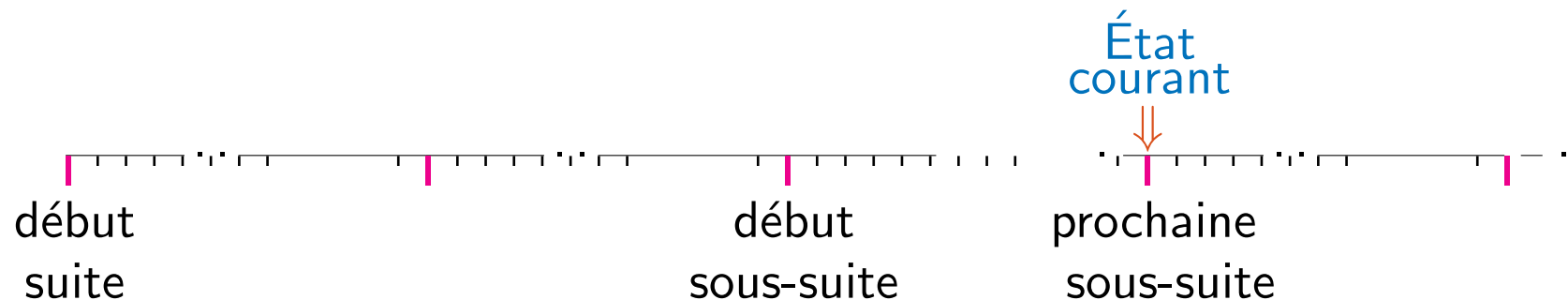
Mais l'implantation, avec synchronisation des v.a., peut être compliquée lorsque les deux configurations n'utilisent pas le même nombre de U_j (e.g., parfois on doit générer une v.a. dans un cas et pas dans l'autre).

Une solution: GPA avec **suites et sous-suites multiples**.

Chaque **suite** peut être vue comme un GPA virtuel.

Elle est partitionnée en **sous-suites**.

On peut créer autant de suites (distinctes et "indépendantes") que l'on veut.



Exemple d'interface Java (dans SSJ)

```
public interface RandomStream {  
    public void  resetStartStream ();  
        Réinitialise la suite à son état initial.  
    public void  resetStartSubstream ();  
        Réinitialise la suite au début de sa sous-suite courante. substream.  
    public void  resetNextSubstream ();  
        Réinitialise la suite au début de sa prochaine sous-suite. substream.  
    public double nextDouble ();  
        Retourne une v.a.  $U(0, 1)$  de cette suite et avance d'un pas.  
    public int  nextInt (int i, int j);  
        Retourne une v.a. uniforme sur  $\{i, i + 1, \dots, j\}$ .  
}
```

```
public class MRG32k3a implements RandomStream {
```

Une implantation particulière, basée sur un générateur de période $\approx 2^{181}$, partitionnée en suites disjointes de longueur 2^{127} , et sous-suites de longueur 2^{76} .

```
    public MRG32k3a();
```

Construit une nouvelle suite.

```
}
```

```
public class LFSR113 implements RandomStream {
```

Une implantation basée sur une combinaison de "LFSR", de période $\approx 2^{113}$.

```
    public LFSR113();
```

Construit une nouvelle suite.

```
}
```

Ce système, basé sur **MRG32k3a**, a été récemment adopté dans plusieurs logiciels de simulation et de statistique, tels que SAS, Arena, Witness, Simul8, Automod, ns2, MATLAB, ...

Génération de valeurs non-uniformes

Lois continues ou discrètes. Méthode par défaut: [inversion](#).

```
public class RandomVariateGen {  
    public RandomVariateGen (RandomStream s, Distribution dist)  
        Crée un générateur pour la loi dist, avec la suite s.  
    public double nextDouble()  
        Génère une nouvelle valeur.
```


Génération de valeurs non-uniformes

Lois continues ou discrètes. Méthode par défaut: [inversion](#).

```
public class RandomVariateGen {  
    public RandomVariateGen (RandomStream s, Distribution dist)  
        Crée un générateur pour la loi dist, avec la suite s.  
    public double nextDouble()  
        Génère une nouvelle valeur.
```

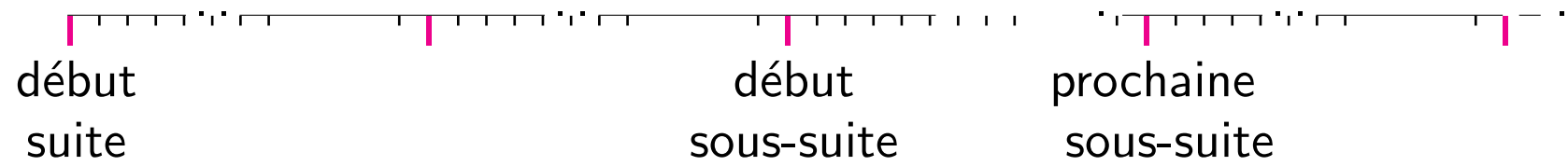
Il y a aussi des générateurs spécialisés pour plusieurs distributions.

```
public class Normal extends RandomVariateGen {  
    public Normal (RandomStream s, double mu, double sigma);  
        Crée un générateur de v.a. normales.  
    public static double nextDouble (RandomStream s, double mu,  
                                     double sigma);  
        Génère une nouvelle v.a. normale, en utilisant la suite s.
```

Supposons que pour une suite de clients, on doit générer:

- (1) un instant d'arrivée pour chaque client,
- (2) une durée de service au serveur A pour chaque client
- (3) une durée de service au serveur B pour des groupes de 10 à 20 clients.

Alors on peut utiliser une suite pour chacun de ces trois usages.



Exemple de comparaison de deux configuration:

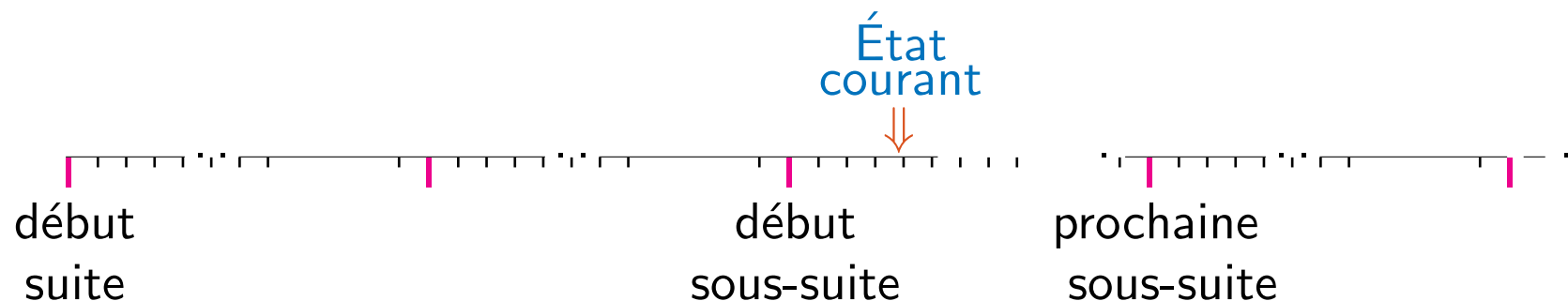
```

:
RandomStream genArr    = new RandomStream(); // Inter-arriv.
RandomStream genServA  = new RandomStream(); // Serveur A.
RandomStream genServB  = new RandomStream(); // Serveur B.

:
for (int rep = 0; rep < n; rep++) {
    genArr.resetNextSubstream();
    genServA.resetNextSubstream();
    genServB.resetNextSubstream();
    --- simuler configuration 1 ---

    genArr.resetStartSubstream();
    genServA.resetStartSubstream();
    genServB.resetStartSubstream();
    --- simuler configuration 2 ---
}

```



Un exemple (trop) simple: générateur à congruence linéaire.

L'état $x_n \in \{0, \dots, m-1\}$ suit la récurrence

$$x_n = ax_{n-1} \bmod m, \quad \text{et l'output est } u_n = x_n/m.$$

Un exemple (trop) simple: générateur à congruence linéaire.

L'état $x_n \in \{0, \dots, m-1\}$ suit la récurrence

$$x_n = ax_{n-1} \bmod m, \quad \text{et l'output est } u_n = x_n/m.$$

Période $\rho \leq m-1$ (peut être atteinte si m est premier).

Un exemple (trop) simple: générateur à congruence linéaire.

L'état $x_n \in \{0, \dots, m-1\}$ suit la récurrence

$$x_n = ax_{n-1} \bmod m, \quad \text{et l'output est } u_n = x_n/m.$$

Période $\rho \leq m-1$ (peut être atteinte si m est premier).

Exemple jouet: $m = 101$ et $a = 12$. Si $x_0 = 10$, alors

$$x_0 = 10$$

$$u_0 = 10/101 \approx 0.09901,$$

Un exemple (trop) simple: générateur à congruence linéaire.

L'état $x_n \in \{0, \dots, m-1\}$ suit la récurrence

$$x_n = ax_{n-1} \bmod m, \quad \text{et l'output est } u_n = x_n/m.$$

Période $\rho \leq m-1$ (peut être atteinte si m est premier).

Exemple jouet: $m = 101$ et $a = 12$. Si $x_0 = 10$, alors

$$\begin{array}{ll} x_0 &= 10 & u_0 &= 10/101 \approx 0.09901, \\ x_1 &= 12 \times 10 \bmod 101 = 19, & u_1 &= 19/101 \approx 0.18812, \end{array}$$

Un exemple (trop) simple: générateur à congruence linéaire.

L'état $x_n \in \{0, \dots, m-1\}$ suit la récurrence

$$x_n = ax_{n-1} \bmod m, \quad \text{et l'output est } u_n = x_n/m.$$

Période $\rho \leq m-1$ (peut être atteinte si m est premier).

Exemple jouet: $m = 101$ et $a = 12$. Si $x_0 = 10$, alors

x_0	$=$	10	u_0	$=$	$10/101$	\approx	0.09901,
x_1	$=$	$12 \times 10 \bmod 101 = 19$,	u_1	$=$	$19/101$	\approx	0.18812,
x_2	$=$	$12 \times 19 \bmod 101 = 26$,	u_2	$=$	$26/101$	\approx	0.25742,

Un exemple (trop) simple: générateur à congruence linéaire.

L'état $x_n \in \{0, \dots, m-1\}$ suit la récurrence

$$x_n = ax_{n-1} \bmod m, \quad \text{et l'output est } u_n = x_n/m.$$

Période $\rho \leq m-1$ (peut être atteinte si m est premier).

Exemple jouet: $m = 101$ et $a = 12$. Si $x_0 = 10$, alors

$x_0 = 10$	$u_0 = 10/101 \approx 0.09901,$
$x_1 = 12 \times 10 \bmod 101 = 19,$	$u_1 = 19/101 \approx 0.18812,$
$x_2 = 12 \times 19 \bmod 101 = 26,$	$u_2 = 26/101 \approx 0.25742,$
$x_3 = 12 \times 26 \bmod 101 = 9,$	$u_3 = 9/101 \approx 0.08911,$

etc. La suite est périodique de période $\rho = 100$.

Uniformité: tous les nombres $\{1/101, 2/101, \dots, 100/101\}$ apparaissent exactement une fois dans le cycle.

Un exemple (trop) simple: générateur à congruence linéaire.

L'état $x_n \in \{0, \dots, m-1\}$ suit la récurrence

$$x_n = ax_{n-1} \bmod m, \quad \text{et l'output est } u_n = x_n/m.$$

Période $\rho \leq m-1$ (peut être atteinte si m est premier).

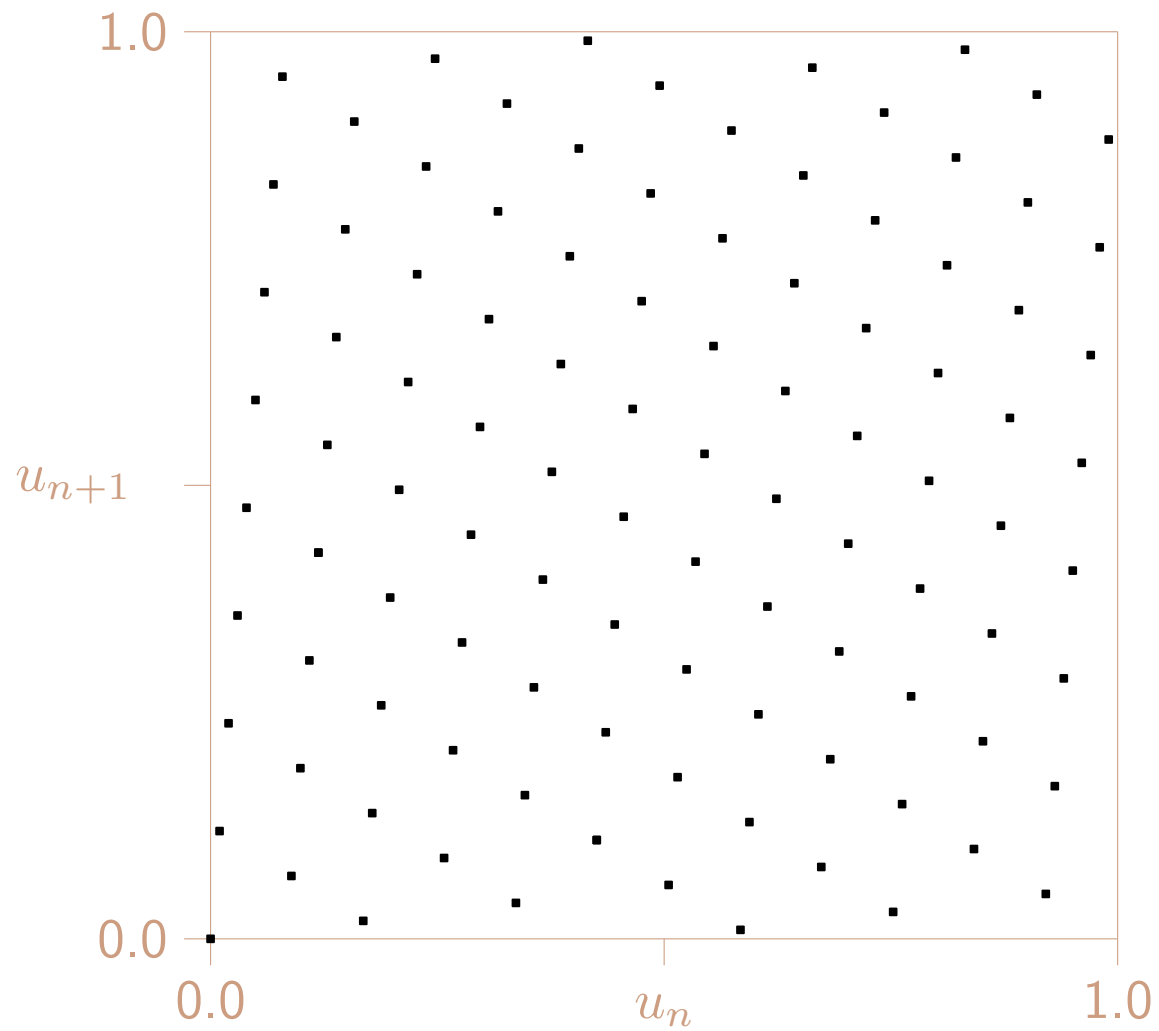
Exemple jouet: $m = 101$ et $a = 12$. Si $x_0 = 10$, alors

$x_0 = 10$	$u_0 = 10/101 \approx 0.09901,$
$x_1 = 12 \times 10 \bmod 101 = 19,$	$u_1 = 19/101 \approx 0.18812,$
$x_2 = 12 \times 19 \bmod 101 = 26,$	$u_2 = 26/101 \approx 0.25742,$
$x_3 = 12 \times 26 \bmod 101 = 9,$	$u_3 = 9/101 \approx 0.08911,$

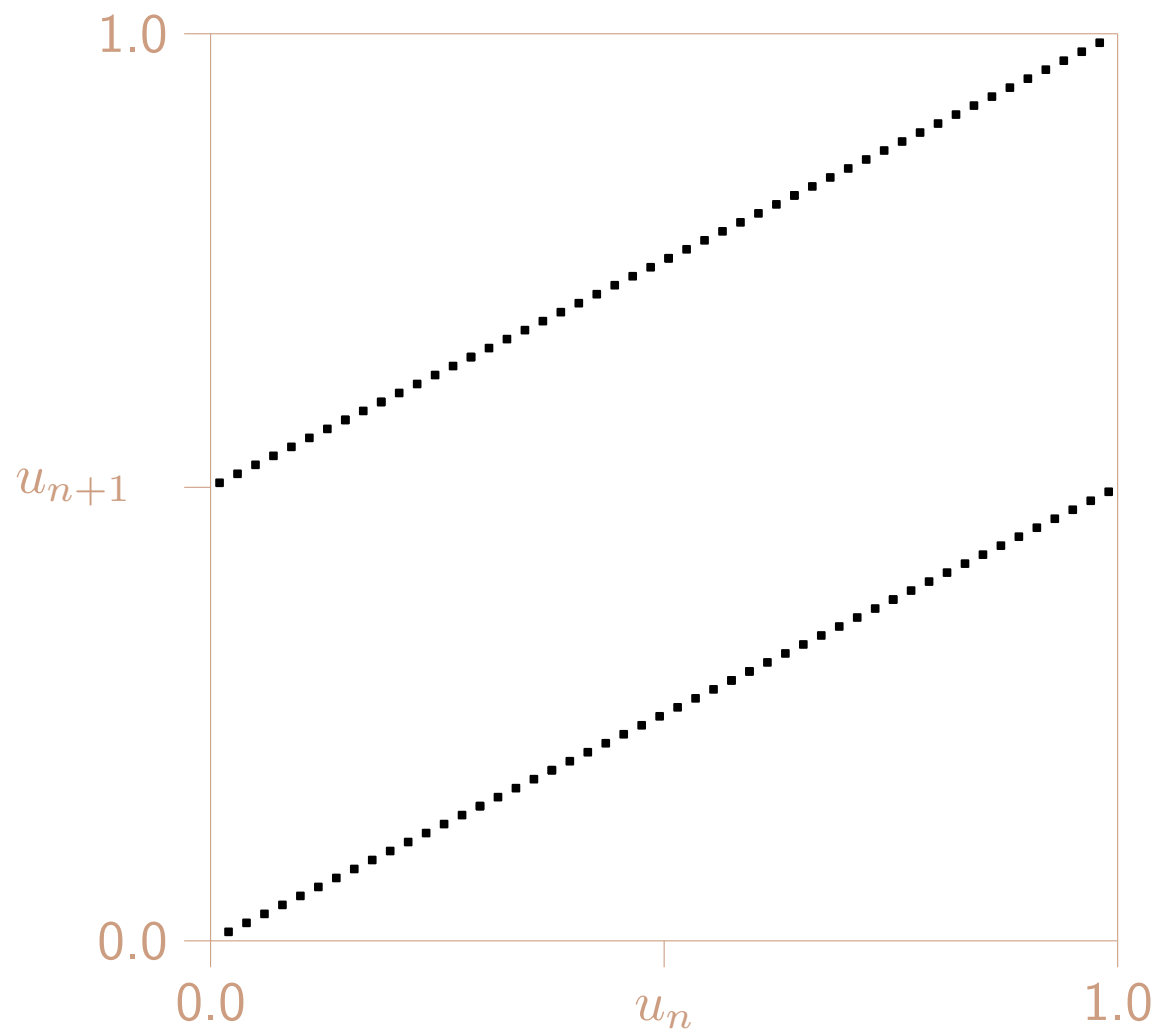
etc. La suite est périodique de période $\rho = 100$.

Uniformité: tous les nombres $\{1/101, 2/101, \dots, 100/101\}$ apparaissent exactement une fois dans le cycle.

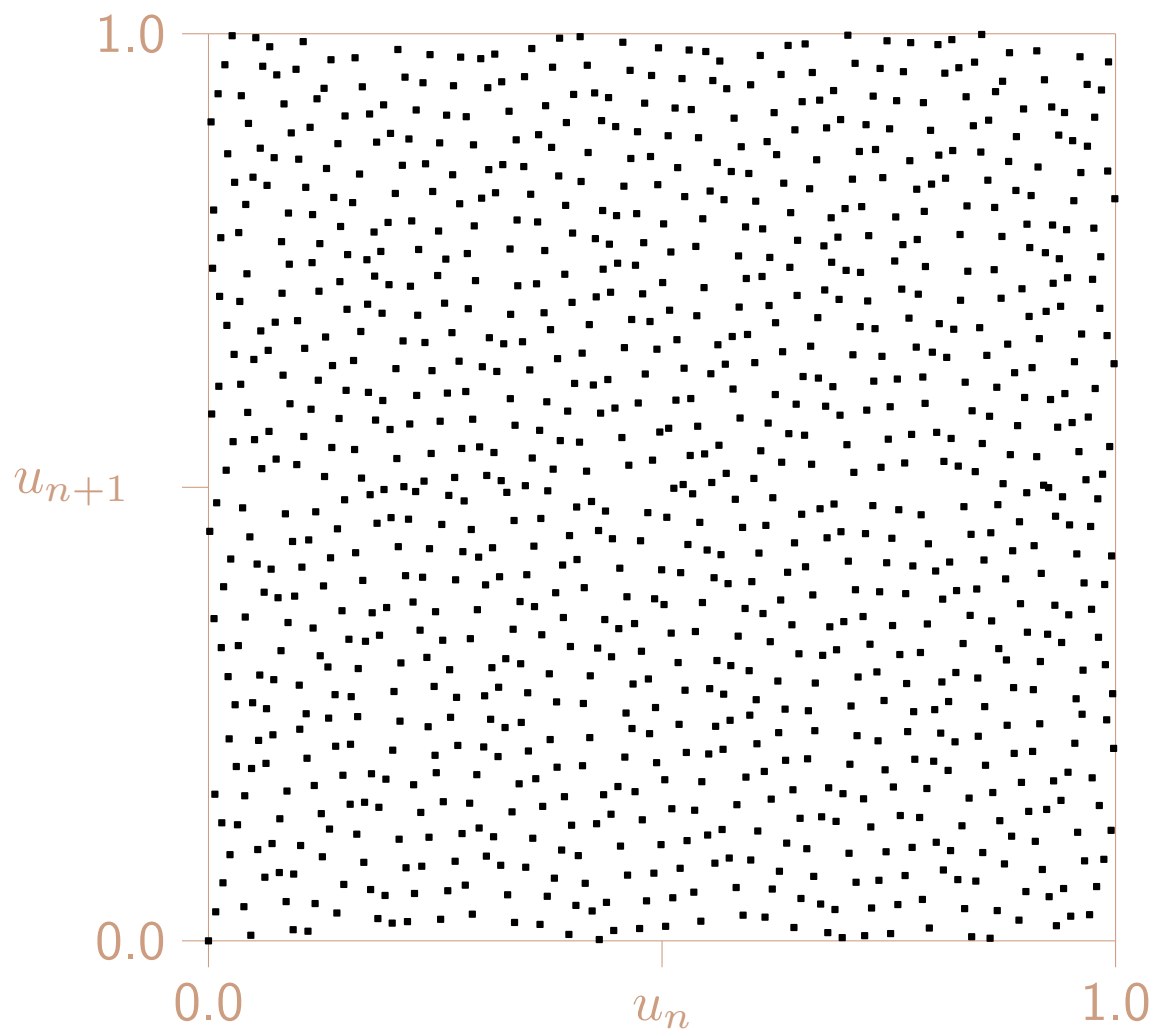
Populaire pour $m = 2^{31} - 1$ ou $m = 2^{32}$ ou $m = 2^{48}$. Mais à rejeter.



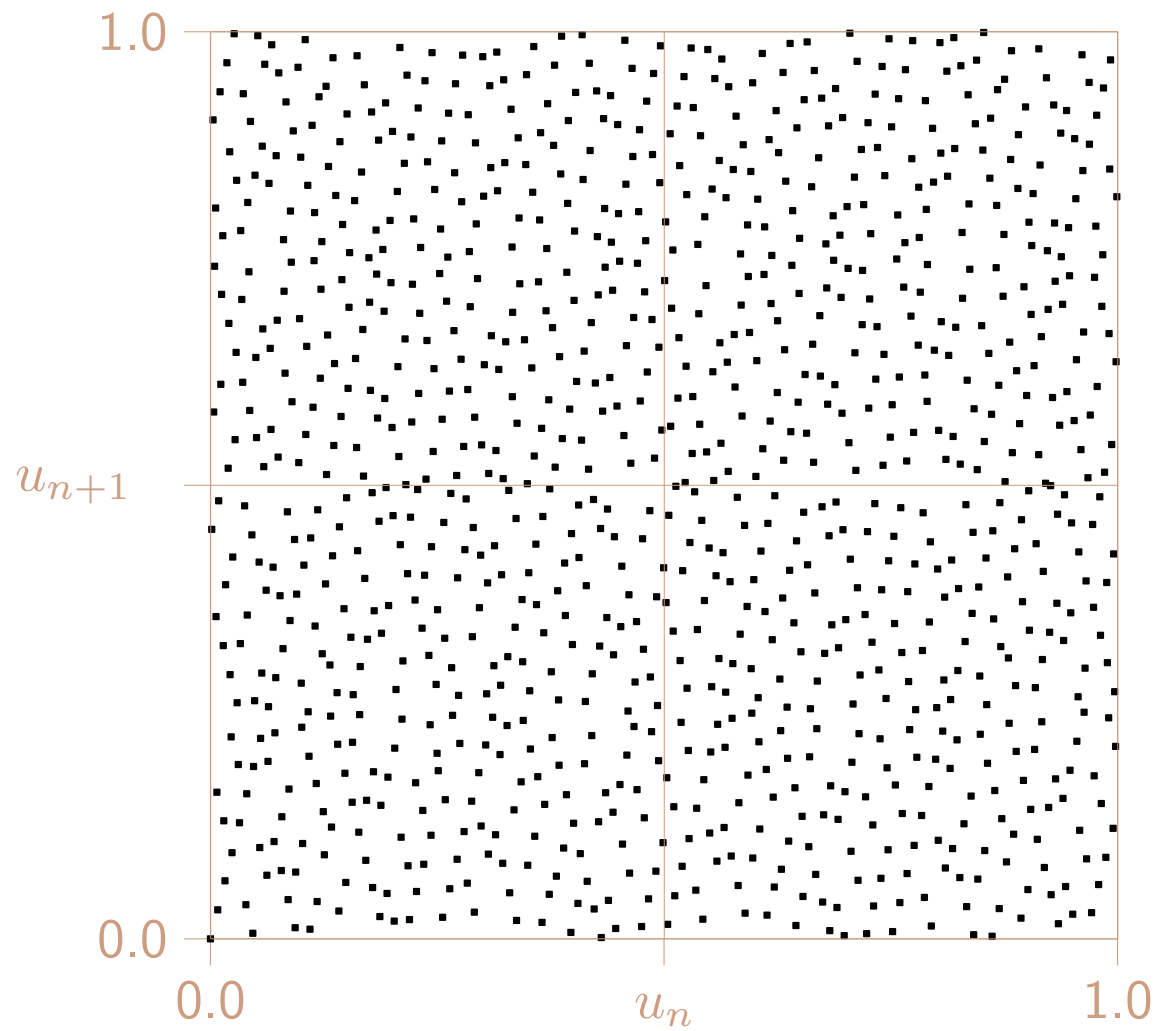
Exemple jouet: LCG avec $m = 101$ et $a = 12$.



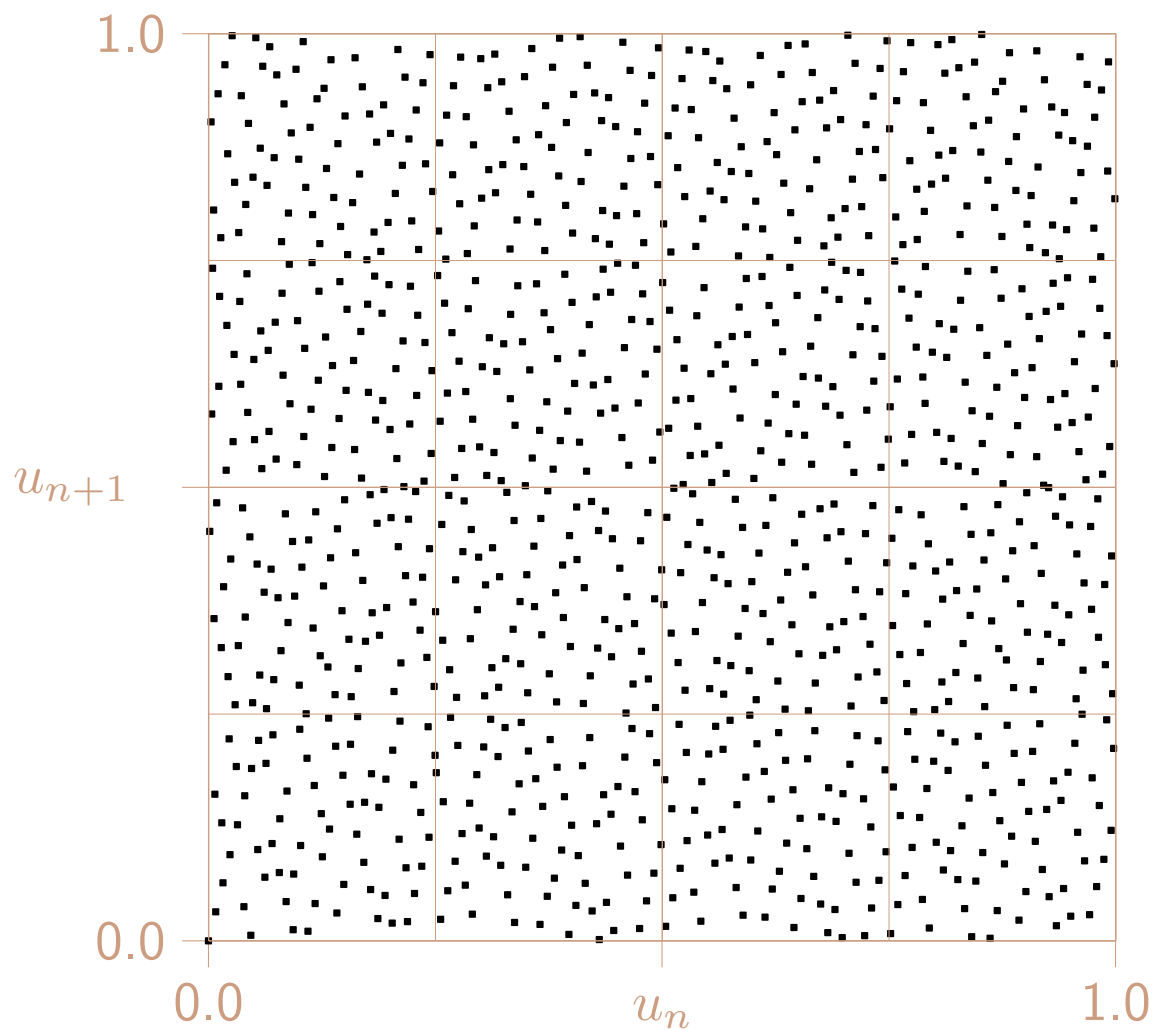
Exemple jouet: LCG avec $m = 101$ et $a = 51$.



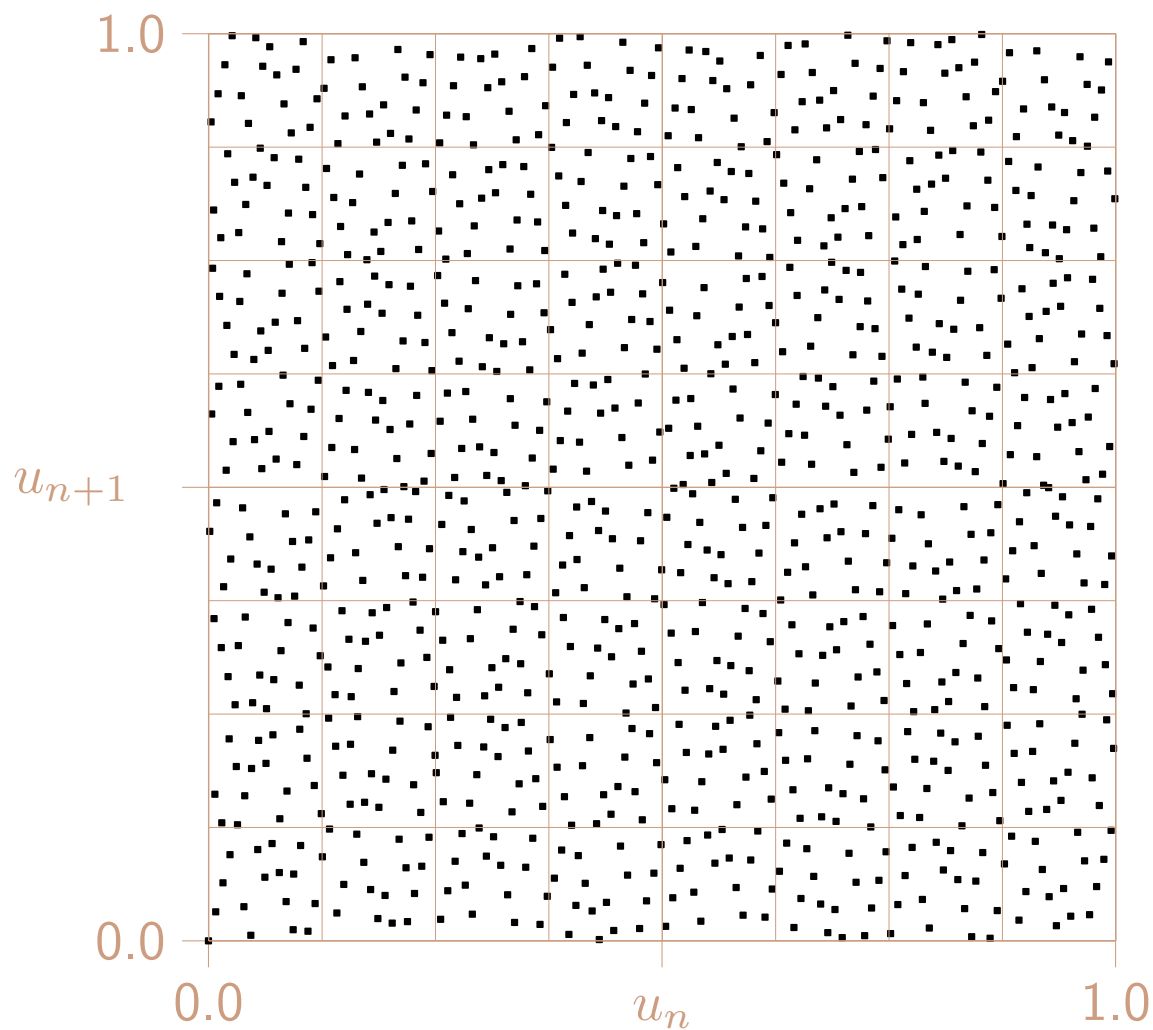
Exemple jouet: LFSR (Tausworthe combiné) avec $n = 1024 = 2^{10}$.



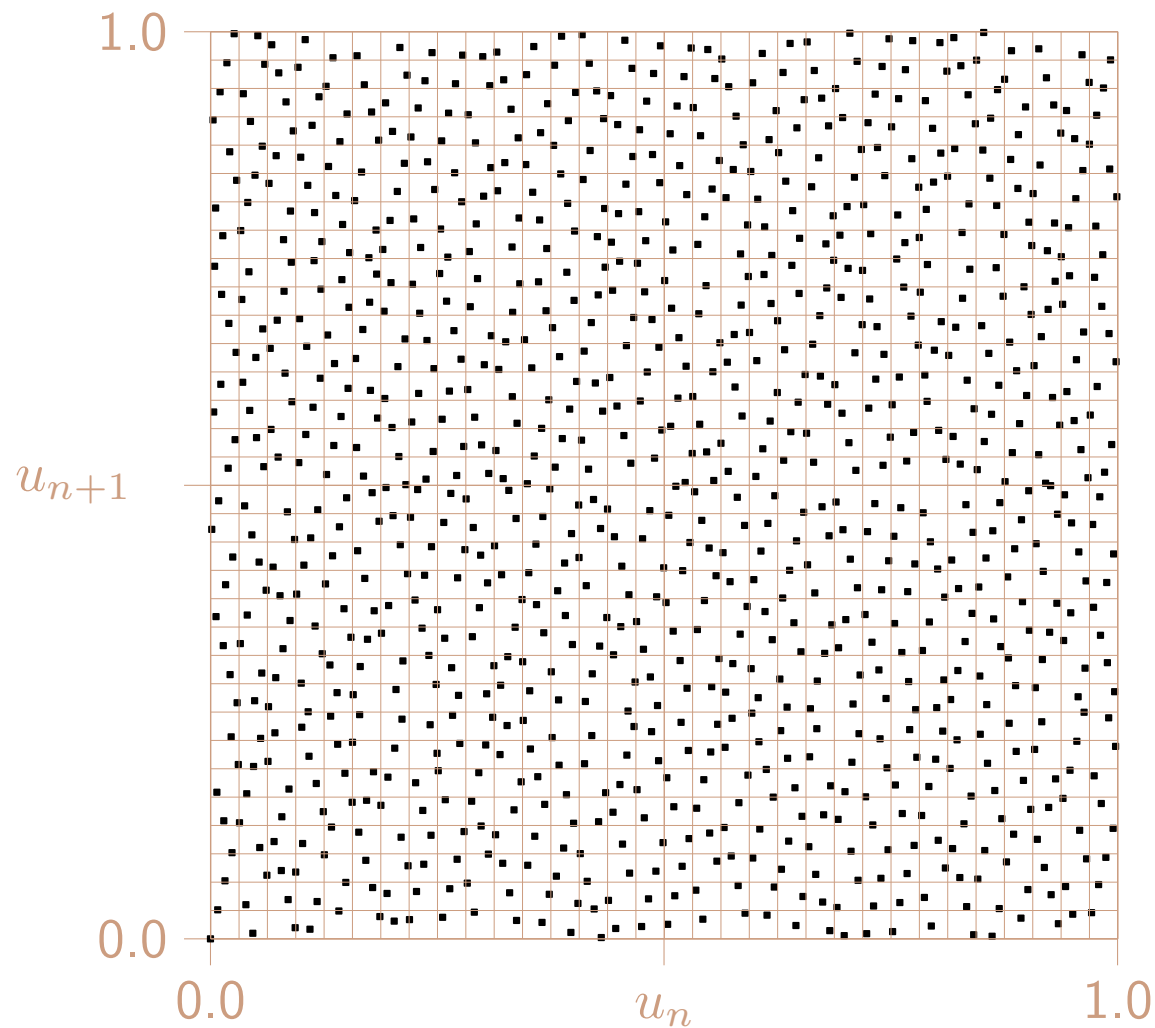
Exemple jouet: LFSR (Tausworthe combiné) avec $n = 1024 = 2^{10}$.



Exemple jouet: LFSR (Tausworthe combiné) avec $n = 1024 = 2^{10}$.



Exemple jouet: LFSR (Tausworthe combiné) avec $n = 1024 = 2^{10}$.



Exemple jouet: LFSR (Tausworthe combiné) avec $n = 1024 = 2^{10}$.

Vrai aussi en 3, 4, 5, ... 10 dimensions.

Exemple d'un meilleur générateur: MRG32k3a

On choisit 6 entiers:

x_{-2}, x_{-1}, x_0 dans $\{0, 1, \dots, 4294967086\}$ (pas tous 0) et
 y_{-2}, y_{-1}, y_0 dans $\{0, 1, \dots, 4294944442\}$ (pas tous 0). Pour $n = 1, 2, \dots$, on pose

$$x_n = (1403580x_{n-2} - 810728x_{n-3}) \bmod 4294967087,$$

$$y_n = (527612y_{n-1} - 1370589y_{n-3}) \bmod 4294944443,$$

$$u_n = [(x_n - y_n) \bmod 4294967087] / 4294967087.$$

Exemple d'un meilleur générateur: MRG32k3a

On choisit 6 entiers:

x_{-2}, x_{-1}, x_0 dans $\{0, 1, \dots, 4294967086\}$ (pas tous 0) et
 y_{-2}, y_{-1}, y_0 dans $\{0, 1, \dots, 4294944442\}$ (pas tous 0). Pour $n = 1, 2, \dots$, on pose

$$x_n = (1403580x_{n-2} - 810728x_{n-3}) \bmod 4294967087,$$

$$y_n = (527612y_{n-1} - 1370589y_{n-3}) \bmod 4294944443,$$

$$u_n = [(x_n - y_n) \bmod 4294967087] / 4294967087.$$

(x_{n-2}, x_{n-1}, x_n) visite chacune des $4294967087^3 - 1$ valeurs possibles.

(y_{n-2}, y_{n-1}, y_n) visite chacune des $4294944443^3 - 1$ valeurs possibles.

La suite u_0, u_1, u_2, \dots est périodique, avec 2 cycles de période

$$\approx 2^{191} \approx 3.1 \times 10^{57}.$$

Exemple d'un meilleur générateur: MRG32k3a

On choisit 6 entiers:

x_{-2}, x_{-1}, x_0 dans $\{0, 1, \dots, 4294967086\}$ (pas tous 0) et
 y_{-2}, y_{-1}, y_0 dans $\{0, 1, \dots, 4294944442\}$ (pas tous 0). Pour $n = 1, 2, \dots$, on pose

$$x_n = (1403580x_{n-2} - 810728x_{n-3}) \bmod 4294967087,$$

$$y_n = (527612y_{n-1} - 1370589y_{n-3}) \bmod 4294944443,$$

$$u_n = [(x_n - y_n) \bmod 4294967087] / 4294967087.$$

(x_{n-2}, x_{n-1}, x_n) visite chacune des $4294967087^3 - 1$ valeurs possibles.

(y_{n-2}, y_{n-1}, y_n) visite chacune des $4294944443^3 - 1$ valeurs possibles.

La suite u_0, u_1, u_2, \dots est périodique, avec 2 cycles de période

$$\approx 2^{191} \approx 3.1 \times 10^{57}.$$

Utilisé dans/par SAS, R, MATLAB, Arena, Automod, Witness, Spielo gaming, ...

Exemple: générateur subtract-with-borrow (SWB)

Paramètres b et $k > r > 0$, **état** $(x_{n-k}, \dots, x_{n-1}, c_{n-1})$ où $x_n \in \{0, \dots, b-1\}$ et $c_n \in \{0, 1\}$ pour tout n .

$$x_n = (x_{n-r} - x_{n-k} - c_{n-1}) \bmod b,$$

$$c_n = \mathbb{I}[x_{n-r} - x_{n-k} - c_{n-1} < 0],$$

$$u_n = x_n/b,$$

Exemple: Dans **Mathematica**, versions ≤ 5.2 , SWB modifié avec $(b, r, k) = (2^{31}, 8, 48)$ et output $u_n = x_{2n}/2^{62} + x_{2n+1}/2^{31}$. **Période** $\rho \approx 2^{1479}$.

Exemple: générateur subtract-with-borrow (SWB)

Paramètres b et $k > r > 0$, **état** $(x_{n-k}, \dots, x_{n-1}, c_{n-1})$ où $x_n \in \{0, \dots, b-1\}$ et $c_n \in \{0, 1\}$ pour tout n .

$$x_n = (x_{n-r} - x_{n-k} - c_{n-1}) \bmod b,$$

$$c_n = \mathbb{I}[x_{n-r} - x_{n-k} - c_{n-1} < 0],$$

$$u_n = x_n/b,$$

Exemple: Dans **Mathematica**, versions ≤ 5.2 , SWB modifié avec $(b, r, k) = (2^{31}, 8, 48)$ et output $u_n = x_{2n}/2^{62} + x_{2n+1}/2^{31}$. **Période** $\rho \approx 2^{1479}$.

Uniformité des points Ψ_s ? À $1/b$ près, les points de Ψ_s pour le SWB sont les mêmes que pour un GCL de modulo $m = b^k - b^r + 1$ et a qui satisfait $ab \bmod m = 1$. De plus, tous les points non nuls de

$$\Psi_{\{0, k-r, k\}} = \{(u_n, u_{n+k-r}, u_{n+k}) : n \geq 0\}$$

pour ce GCL sont contenus dans seulement deux plans dans le cube unitaire, définis par $u_{n+k} - u_{n+k-r} + u_n = q$ pour $q = 0$ ou 1 . Très mauvais!

Définition et conception d'un GPA

\mathcal{S} , espace d'états fini;

$f : \mathcal{S} \rightarrow \mathcal{S}$, fonction de transition;

s_0 , germe (état initial);

$$s_n = f(s_{n-1})$$

Définition et conception d'un GPA

\mathcal{S} , espace d'états fini;

$f : \mathcal{S} \rightarrow \mathcal{S}$, fonction de transition;

\mathcal{U} , espace des valeurs de sortie;

$g : \mathcal{S} \rightarrow \mathcal{U}$, fonction de sortie.

s_0 , germe (état initial);

$$s_n = f(s_{n-1})$$

$$u_n = g(s_n)$$

Définition et conception d'un GPA

\mathcal{S} , espace d'états fini;

$f : \mathcal{S} \rightarrow \mathcal{S}$, fonction de transition;

\mathcal{U} , espace des valeurs de sortie;

$g : \mathcal{S} \rightarrow \mathcal{U}$, fonction de sortie.

s_0 , germe (état initial);

$$s_n = f(s_{n-1})$$

$$u_n = g(s_n)$$

s_0

Définition et conception d'un GPA

\mathcal{S} , espace d'états fini;

$f : \mathcal{S} \rightarrow \mathcal{S}$, fonction de transition;

\mathcal{U} , espace des valeurs de sortie;

$g : \mathcal{S} \rightarrow \mathcal{U}$, fonction de sortie.

s_0 , germe (état initial);

$$s_n = f(s_{n-1})$$

$$u_n = g(s_n)$$

$$\begin{array}{c} s_0 \\ \downarrow g \\ u_0 \end{array}$$

Définition et conception d'un GPA

\mathcal{S} , espace d'états fini;

$f : \mathcal{S} \rightarrow \mathcal{S}$, fonction de transition;

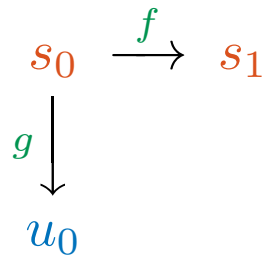
\mathcal{U} , espace des valeurs de sortie;

$g : \mathcal{S} \rightarrow \mathcal{U}$, fonction de sortie.

s_0 , germe (état initial);

$$s_n = f(s_{n-1})$$

$$u_n = g(s_n)$$



Définition et conception d'un GPA

\mathcal{S} , espace d'états fini;

$f : \mathcal{S} \rightarrow \mathcal{S}$, fonction de transition;

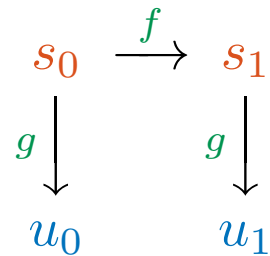
\mathcal{U} , espace des valeurs de sortie;

$g : \mathcal{S} \rightarrow \mathcal{U}$, fonction de sortie.

s_0 , germe (état initial);

$$s_n = f(s_{n-1})$$

$$u_n = g(s_n)$$



Définition et conception d'un GPA

\mathcal{S} , espace d'états fini;

$f : \mathcal{S} \rightarrow \mathcal{S}$, fonction de transition;

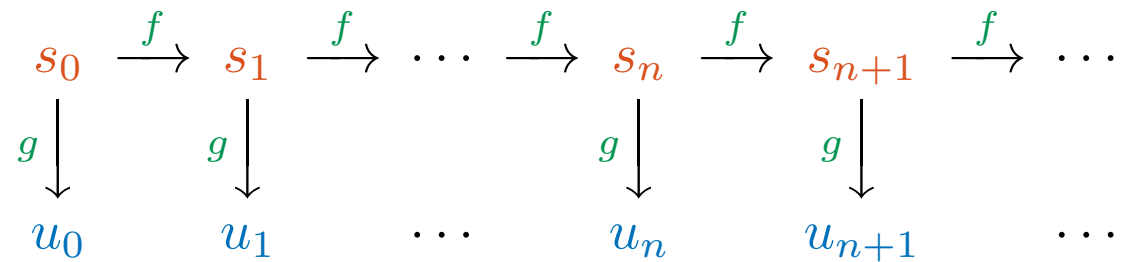
\mathcal{U} , espace des valeurs de sortie;

$g : \mathcal{S} \rightarrow \mathcal{U}$, fonction de sortie.

s_0 , germe (état initial);

$$s_n = f(s_{n-1})$$

$$u_n = g(s_n)$$



Définition et conception d'un GPA

\mathcal{S} , espace d'états fini;

$f : \mathcal{S} \rightarrow \mathcal{S}$, fonction de transition;

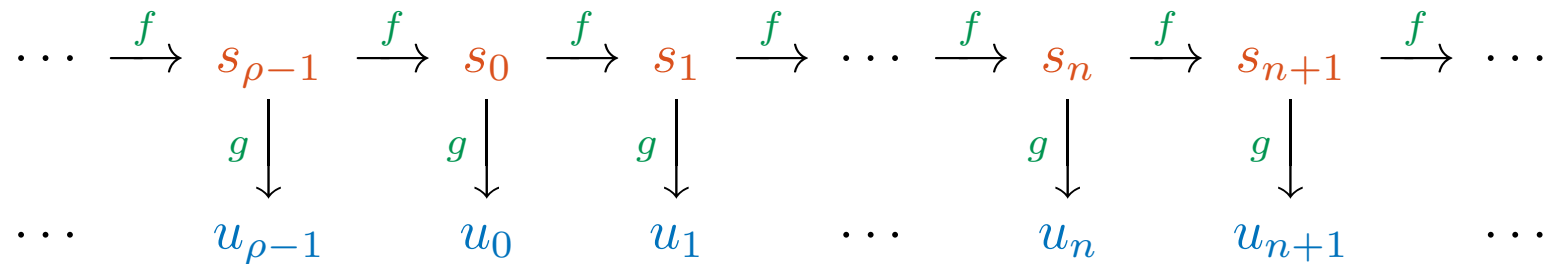
\mathcal{U} , espace des valeurs de sortie;

$g : \mathcal{S} \rightarrow \mathcal{U}$, fonction de sortie.

s_0 , germe (état initial);

$$s_n = f(s_{n-1})$$

$$u_n = g(s_n)$$



Définition et conception d'un GPA

\mathcal{S} , espace d'états fini;

$f : \mathcal{S} \rightarrow \mathcal{S}$, fonction de transition;

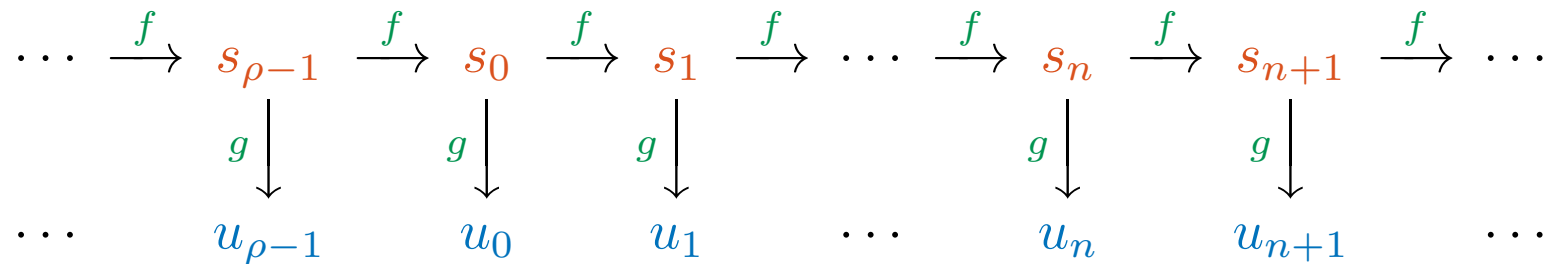
\mathcal{U} , espace des valeurs de sortie;

$g : \mathcal{S} \rightarrow \mathcal{U}$, fonction de sortie.

s_0 , germe (état initial);

$$s_n = f(s_{n-1})$$

$$u_n = g(s_n)$$



Période: $\rho \leq \text{card}(\mathcal{S})$. $s_{n+\rho} = s_n \quad \forall n \geq \tau$.

Définition et conception d'un GPA

\mathcal{S} , espace d'états fini;

$f : \mathcal{S} \rightarrow \mathcal{S}$, fonction de transition;

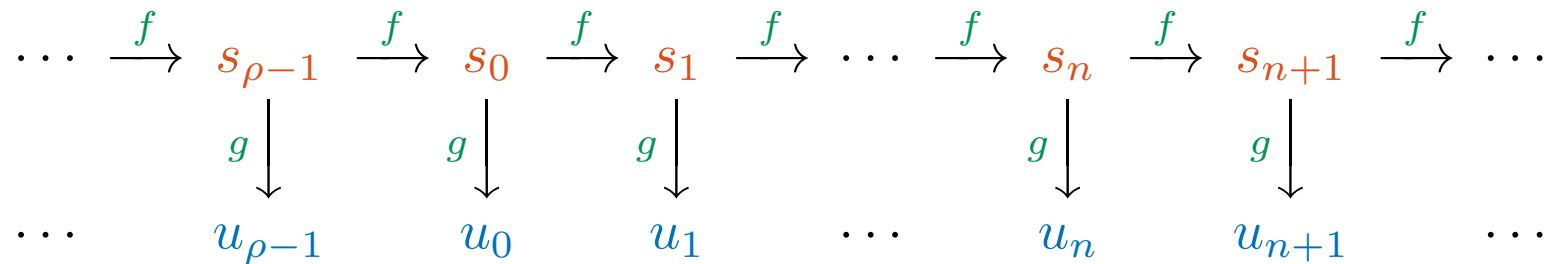
\mathcal{U} , espace des valeurs de sortie;

$g : \mathcal{S} \rightarrow \mathcal{U}$, fonction de sortie.

s_0 , germe (état initial);

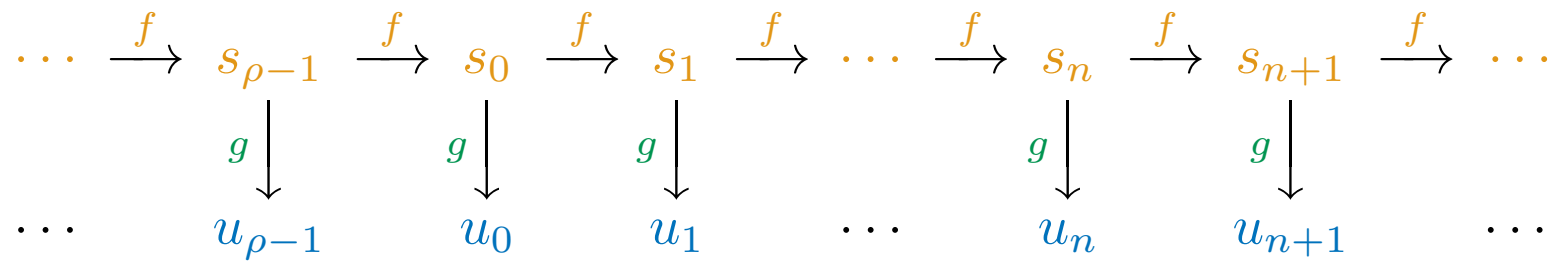
$$s_n = f(s_{n-1})$$

$$u_n = g(s_n)$$

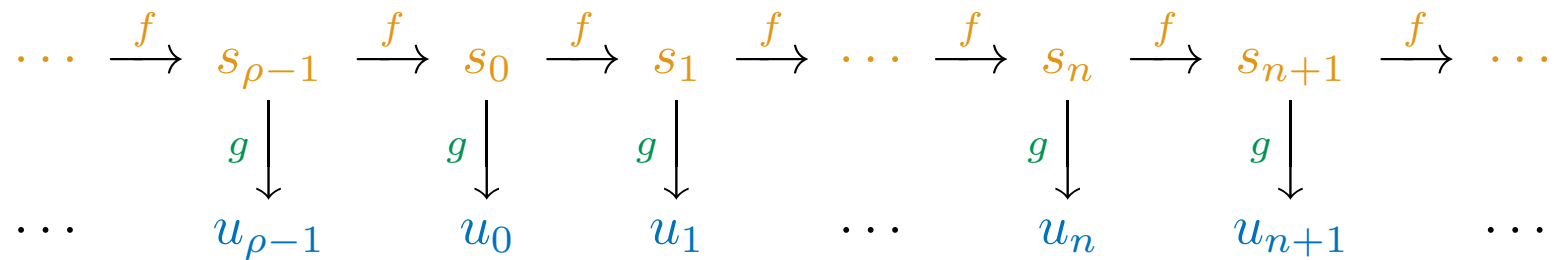


Période: $\rho \leq \text{card}(\mathcal{S})$. $s_{n+\rho} = s_n \quad \forall n \geq \tau$.

On supposera que $\tau = 0$ et $\mathcal{U} = [0, 1)$.



Objectif: en observant seulement (u_0, u_1, \dots) , il doit être difficile de distinguer cette suite de la réalisation d'une suite de v.a. i.i.d. uniformes sur \mathcal{U} .

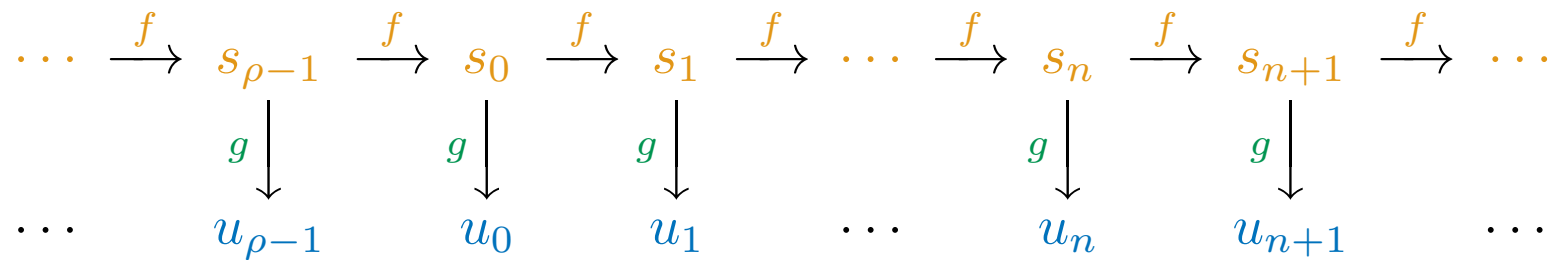


Objectif: en observant seulement (u_0, u_1, \dots) , il doit être difficile de distinguer cette suite de la réalisation d'une suite de v.a. i.i.d. uniformes sur \mathcal{U} .

Utopie: ne pas pouvoir distinguer mieux qu'en tirant à pile ou face.

Autrement dit, que la suite passe **tous** les tests statistiques imaginables.

Cela est impossible! On y reviendra plus loin.



Objectif: en observant seulement (u_0, u_1, \dots) , il doit être difficile de distinguer cette suite de la réalisation d'une suite de v.a. i.i.d. uniformes sur \mathcal{U} .

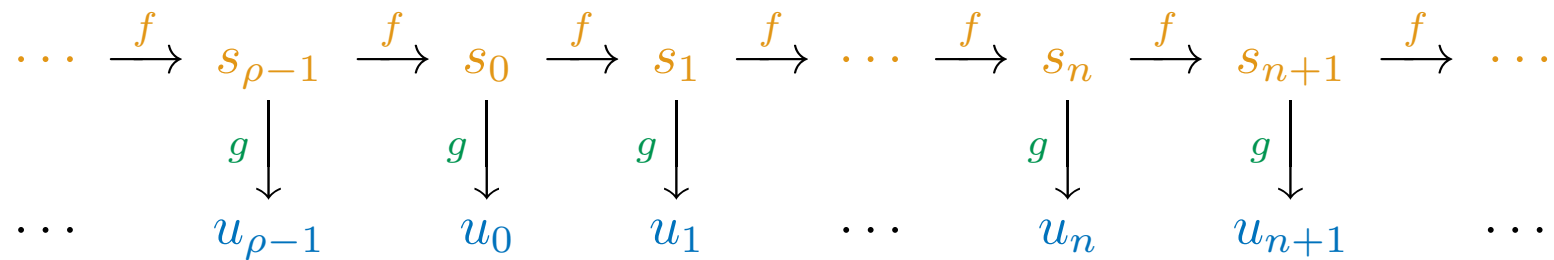
Utopie: ne pas pouvoir distinguer mieux qu'en tirant à pile ou face.

Autrement dit, que la suite passe **tous** les tests statistiques imaginables.

Cela est impossible! On y reviendra plus loin.

On veut aussi: vitesse, facilité d'implantation, suites reproductibles.

Compromis entre vitesse / bonnes propriétés statistiques / (im)prévisibilité.



Objectif: en observant seulement (u_0, u_1, \dots) , il doit être difficile de distinguer cette suite de la réalisation d'une suite de v.a. i.i.d. uniformes sur \mathcal{U} .

Utopie: ne pas pouvoir distinguer mieux qu'en tirant à pile ou face.

Autrement dit, que la suite passe **tous** les tests statistiques imaginables.

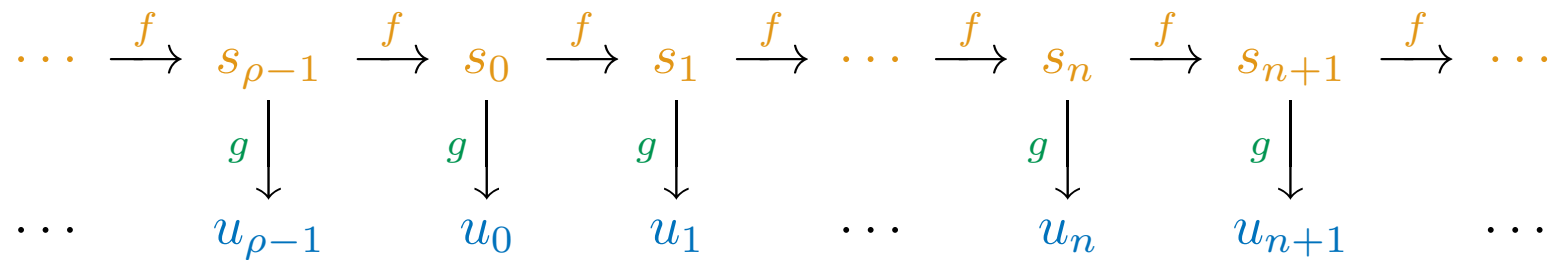
Cela est impossible! On y reviendra plus loin.

On veut aussi: vitesse, facilité d'implantation, suites reproductibles.

Compromis entre vitesse / bonnes propriétés statistiques / (im)prévisibilité.

Si l'état initial s_0 est choisi au hasard, le GPA est comme une roulette géante:

Pour générer t nombres aléatoires, on tourne la roulette pour choisir s_0 , puis on retient $\mathbf{u} = (u_0, \dots, u_{t-1})$.



Objectif: en observant seulement (u_0, u_1, \dots) , il doit être difficile de distinguer cette suite de la réalisation d'une suite de v.a. i.i.d. uniformes sur \mathcal{U} .

Utopie: ne pas pouvoir distinguer mieux qu'en tirant à pile ou face.

Autrement dit, que la suite passe **tous** les tests statistiques imaginables.

Cela est impossible! On y reviendra plus loin.

On veut aussi: vitesse, facilité d'implantation, suites reproductibles.

Compromis entre vitesse / bonnes propriétés statistiques / (im)prévisibilité.

Si l'état initial s_0 est choisi au hasard, le GPA est comme une roulette géante:

Pour générer t nombres aléatoires, on tourne la roulette pour choisir s_0 , puis on retient $\mathbf{u} = (u_0, \dots, u_{t-1})$.

Machines de casinos et loteries: on réinitialise s_0 très souvent.

La loi uniforme sur $[0, 1]^t$.

Choisir s_0 au hasard correspond à choisir un point au hasard dans l'espace échantillonnal $\Psi_t = \{\mathbf{u} = (u_0, \dots, u_{t-1}) = (g(s_0), \dots, g(s_{t-1})), s_0 \in \mathcal{S}\}$, qui peut être interprété comme une approximation de $[0, 1]^t$.

La loi uniforme sur $[0, 1]^t$.

Choisir s_0 au hasard correspond à choisir un point au hasard dans l'espace échantillonnal $\Psi_t = \{\mathbf{u} = (u_0, \dots, u_{t-1}) = (g(s_0), \dots, g(s_{t-1})), s_0 \in \mathcal{S}\}$, qui peut être interprété comme une approximation de $[0, 1]^t$.

Critère: Ψ_t doit recouvrir $[0, 1]^t$ très uniformément pour t jusqu'à (disons) t_0 .

La loi uniforme sur $[0, 1]^t$.

Choisir s_0 au hasard correspond à choisir un point au hasard dans l'espace échantillonnal $\Psi_t = \{\mathbf{u} = (u_0, \dots, u_{t-1}) = (g(s_0), \dots, g(s_{t-1})), s_0 \in \mathcal{S}\}$, qui peut être interprété comme une approximation de $[0, 1]^t$.

Critère: Ψ_t doit recouvrir $[0, 1]^t$ **très uniformément** pour t jusqu'à (disons) t_0 .

Il nous faut une mesure d'**uniformité** de Ψ_t (ou une mesure de **discrépance** entre la loi empirique de Ψ_t et la loi uniforme). Plusieurs définitions possibles.

Important: doit être facilement **calculable** sans générer les points.

Pour cela, il faut bien comprendre la structure mathématique de Ψ_t .

Pour cette raison, la plupart des GPA utilisés en simulation sont basés sur des récurrences linéaires (simples).

La loi uniforme sur $[0, 1]^t$.

Choisir s_0 au hasard correspond à choisir un point au hasard dans l'espace échantillonnal $\Psi_t = \{\mathbf{u} = (u_0, \dots, u_{t-1}) = (g(s_0), \dots, g(s_{t-1})), s_0 \in \mathcal{S}\}$, qui peut être interprété comme une approximation de $[0, 1]^t$.

Critère: Ψ_t doit recouvrir $[0, 1]^t$ **très uniformément** pour t jusqu'à (disons) t_0 .

Il nous faut une mesure d'**uniformité** de Ψ_t (ou une mesure de **discrépance** entre la loi empirique de Ψ_t et la loi uniforme). Plusieurs définitions possibles.

Important: doit être facilement **calculable** sans générer les points.

Pour cela, il faut bien comprendre la structure mathématique de Ψ_t .

Pour cette raison, la plupart des GPA utilisés en simulation sont basés sur des récurrences linéaires (simples).

Pourquoi ne pas insister que Ψ_t lui-même ressemble à un ensemble de points choisis au hasard (e.g., ne soit pas trop uniforme)?

En fait, on veut cela seulement pour la fraction infime de Ψ_t que l'on utilise.

La loi uniforme sur $[0, 1]^t$.

Choisir s_0 au hasard correspond à choisir un point au hasard dans l'espace échantillonnal $\Psi_t = \{\mathbf{u} = (u_0, \dots, u_{t-1}) = (g(s_0), \dots, g(s_{t-1})), s_0 \in \mathcal{S}\}$, qui peut être interprété comme une approximation de $[0, 1]^t$.

Critère: Ψ_t doit recouvrir $[0, 1]^t$ **très uniformément** pour t jusqu'à (disons) t_0 .

Il nous faut une mesure d'**uniformité** de Ψ_t (ou une mesure de **discrépance** entre la loi empirique de Ψ_t et la loi uniforme). Plusieurs définitions possibles.

Important: doit être facilement **calculable** sans générer les points.

Pour cela, il faut bien comprendre la structure mathématique de Ψ_t .

Pour cette raison, la plupart des GPA utilisés en simulation sont basés sur des récurrences linéaires (simples).

Pourquoi ne pas insister que Ψ_t lui-même ressemble à un ensemble de points choisis au hasard (e.g., ne soit pas trop uniforme)?

En fait, on veut cela seulement pour la fraction infime de Ψ_t que l'on utilise.

Généralisation: mesurer l'uniformité de $\Psi_I = \{(u_{i_1}, \dots, u_{i_t}) \mid s_0 \in \mathcal{S}\}$ pour une classe choisie d'ensembles d'indices (non successifs) de forme $I = \{i_1, i_2, \dots, i_t\}$.

Pour une famille donnée \mathcal{I} de sous-ensembles de $\{0, 1, \dots, t\}$, on s'assure que Ψ_I est suffisamment uniforme pour tout $I \in \mathcal{I}$.

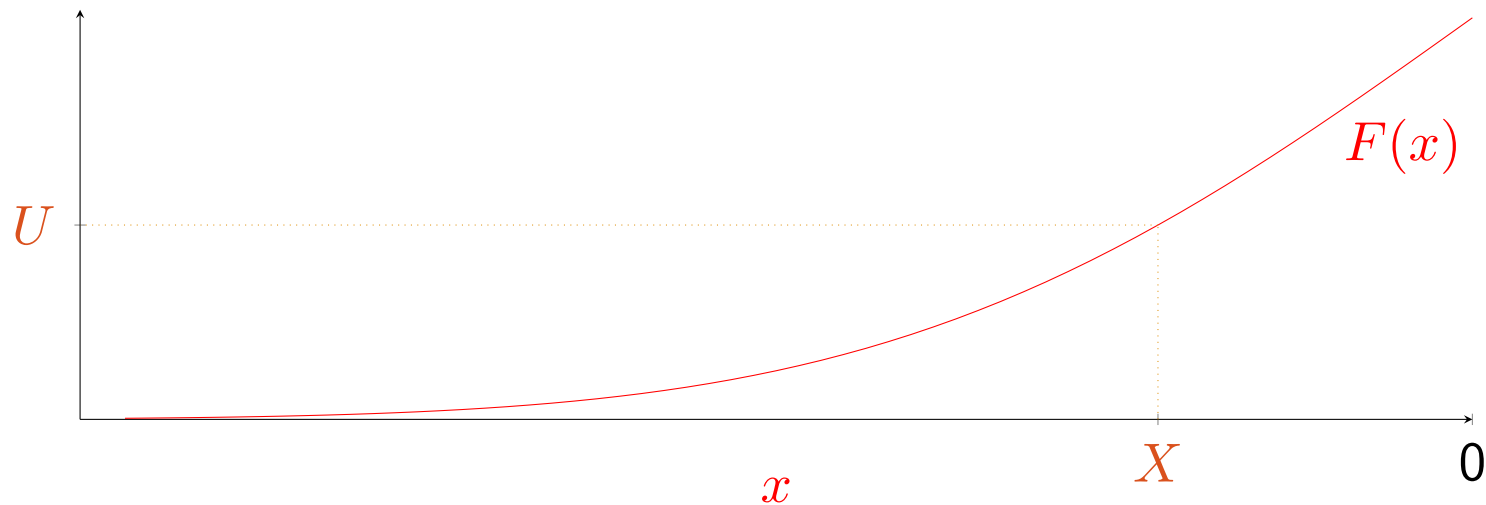
V.A. Non Uniformes: Inversion

Soit F la fonction de répartition de X , i.e., $F(x) = \mathbb{P}[X \leq x]$ pour tout $x \in \mathbb{R}$.

V.A. Non Uniformes: Inversion

Soit F la fonction de répartition de X , i.e., $F(x) = \mathbb{P}[X \leq x]$ pour tout $x \in \mathbb{R}$.
Pour générer X , générer $U \sim U(0, 1)$ et poser:

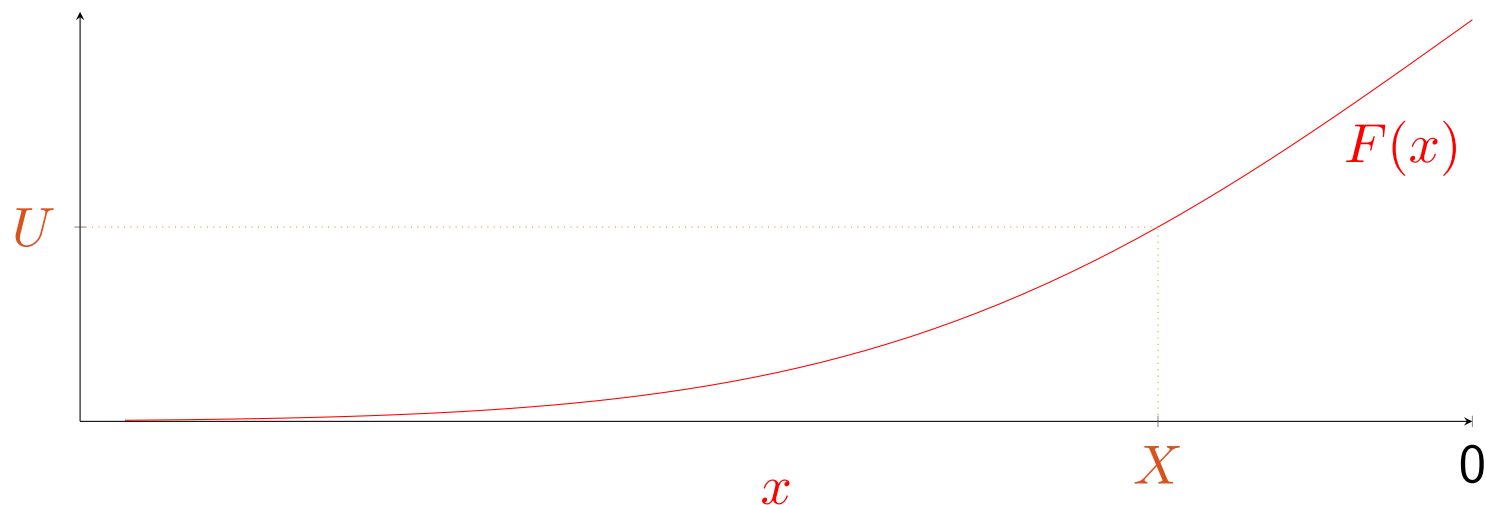
$$X = F^{-1}(U) = \min\{x \mid F(x) \geq U\}.$$



V.A. Non Uniformes: Inversion

Soit F la fonction de répartition de X , i.e., $F(x) = \mathbb{P}[X \leq x]$ pour tout $x \in \mathbb{R}$.
Pour générer X , générer $U \sim U(0, 1)$ et poser:

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

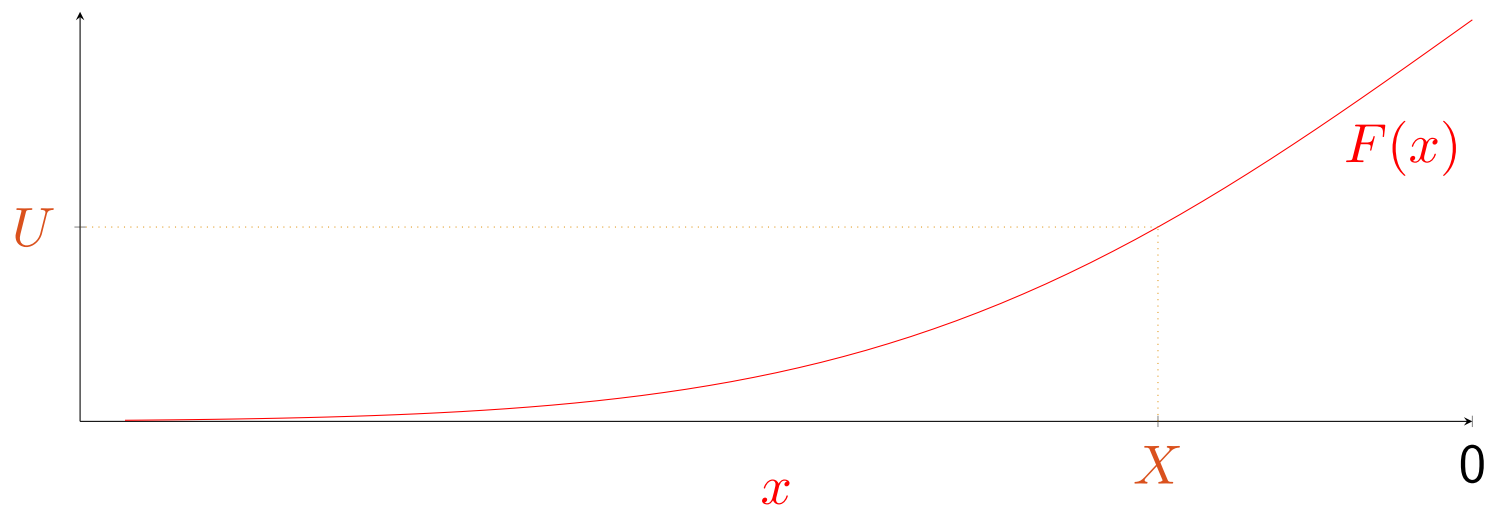


On a en effet $\mathbb{P}[X \leq x] = \mathbb{P}[F^{-1}(U) \leq x]$

V.A. Non Uniformes: Inversion

Soit F la fonction de répartition de X , i.e., $F(x) = \mathbb{P}[X \leq x]$ pour tout $x \in \mathbb{R}$.
Pour générer X , générer $U \sim U(0, 1)$ et poser:

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

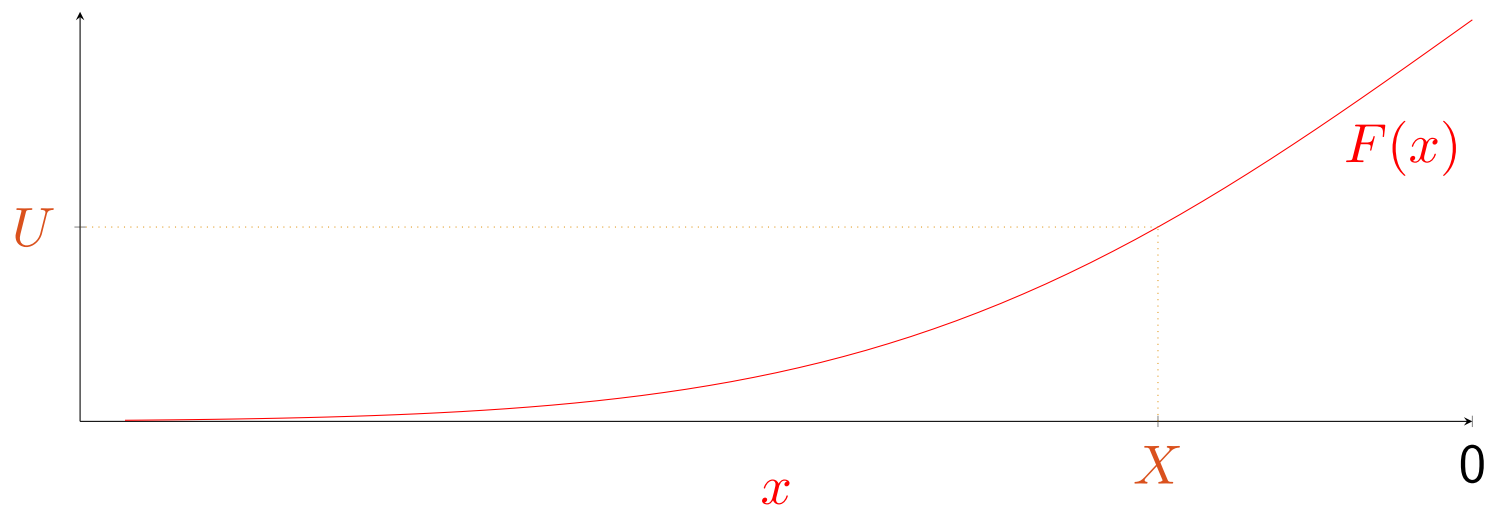


On a en effet $\mathbb{P}[X \leq x] = \mathbb{P}[F^{-1}(U) \leq x] = \mathbb{P}[U \leq F(x)]$

V.A. Non Uniformes: Inversion

Soit F la fonction de répartition de X , i.e., $F(x) = \mathbb{P}[X \leq x]$ pour tout $x \in \mathbb{R}$.
Pour générer X , générer $U \sim U(0, 1)$ et poser:

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

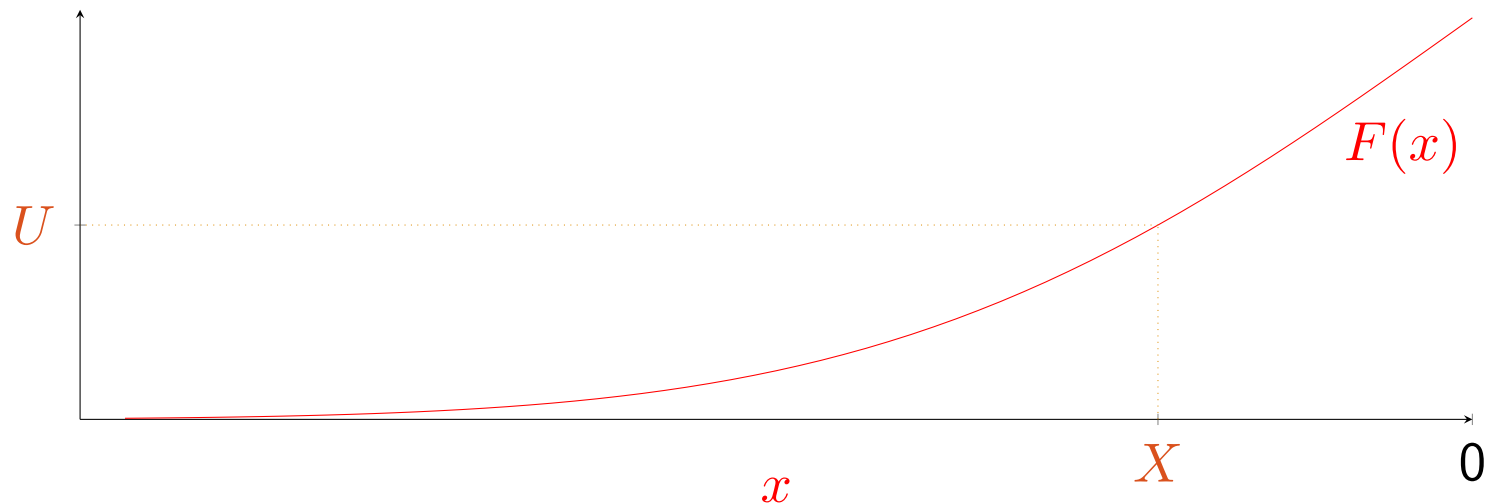


On a en effet $\mathbb{P}[X \leq x] = \mathbb{P}[F^{-1}(U) \leq x] = \mathbb{P}[U \leq F(x)] = F(x)$.

V.A. Non Uniformes: Inversion

Soit F la fonction de répartition de X , i.e., $F(x) = \mathbb{P}[X \leq x]$ pour tout $x \in \mathbb{R}$.
Pour générer X , générer $U \sim U(0, 1)$ et poser:

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



On a en effet $\mathbb{P}[X \leq x] = \mathbb{P}[F^{-1}(U) \leq x] = \mathbb{P}[U \leq F(x)] = F(x)$.

Contrainte: il faut une formule ou une bonne approximation pour F^{-1} .

Exemple: $X \sim \text{Weibull}(\alpha, \lambda)$.

$$F(x) = 1 - \exp[-(\lambda x)^\alpha] \quad \text{pour } x > 0.$$

Exemple: $X \sim \text{Weibull}(\alpha, \lambda)$.

$$F(x) = 1 - \exp[-(\lambda x)^\alpha] \quad \text{pour } x > 0.$$

Inversion:

$$U = 1 - \exp[-(\lambda X)^\alpha]$$

Exemple: $X \sim \text{Weibull}(\alpha, \lambda)$.

$$F(x) = 1 - \exp[-(\lambda x)^\alpha] \quad \text{pour } x > 0.$$

Inversion:

$$\begin{aligned} U &= 1 - \exp[-(\lambda X)^\alpha] \\ \exp[-(\lambda X)^\alpha] &= 1 - U \end{aligned}$$

Exemple: $X \sim \text{Weibull}(\alpha, \lambda)$.

$$F(x) = 1 - \exp[-(\lambda x)^\alpha] \quad \text{pour } x > 0.$$

Inversion:

$$\begin{aligned} U &= 1 - \exp[-(\lambda X)^\alpha] \\ \exp[-(\lambda X)^\alpha] &= 1 - U \\ (\lambda X)^\alpha &= -\ln(1 - U) \end{aligned}$$

Exemple: $X \sim \text{Weibull}(\alpha, \lambda)$.

$$F(x) = 1 - \exp[-(\lambda x)^\alpha] \quad \text{pour } x > 0.$$

Inversion:

$$\begin{aligned} U &= 1 - \exp[-(\lambda X)^\alpha] \\ \exp[-(\lambda X)^\alpha] &= 1 - U \\ (\lambda X)^\alpha &= -\ln(1 - U) \\ \lambda X &= [-\ln(1 - U)]^{1/\alpha} \end{aligned}$$

Exemple: $X \sim \text{Weibull}(\alpha, \lambda)$.

$$F(x) = 1 - \exp[-(\lambda x)^\alpha] \quad \text{pour } x > 0.$$

Inversion:

$$\begin{aligned} U &= 1 - \exp[-(\lambda X)^\alpha] \\ \exp[-(\lambda X)^\alpha] &= 1 - U \\ (\lambda X)^\alpha &= -\ln(1 - U) \\ \lambda X &= [-\ln(1 - U)]^{1/\alpha} \\ X &= [-\ln(1 - U)]^{1/\alpha} / \lambda = F^{-1}(U). \end{aligned}$$

Exemple: $X \sim \text{Weibull}(\alpha, \lambda)$.

$$F(x) = 1 - \exp[-(\lambda x)^\alpha] \quad \text{pour } x > 0.$$

Inversion:

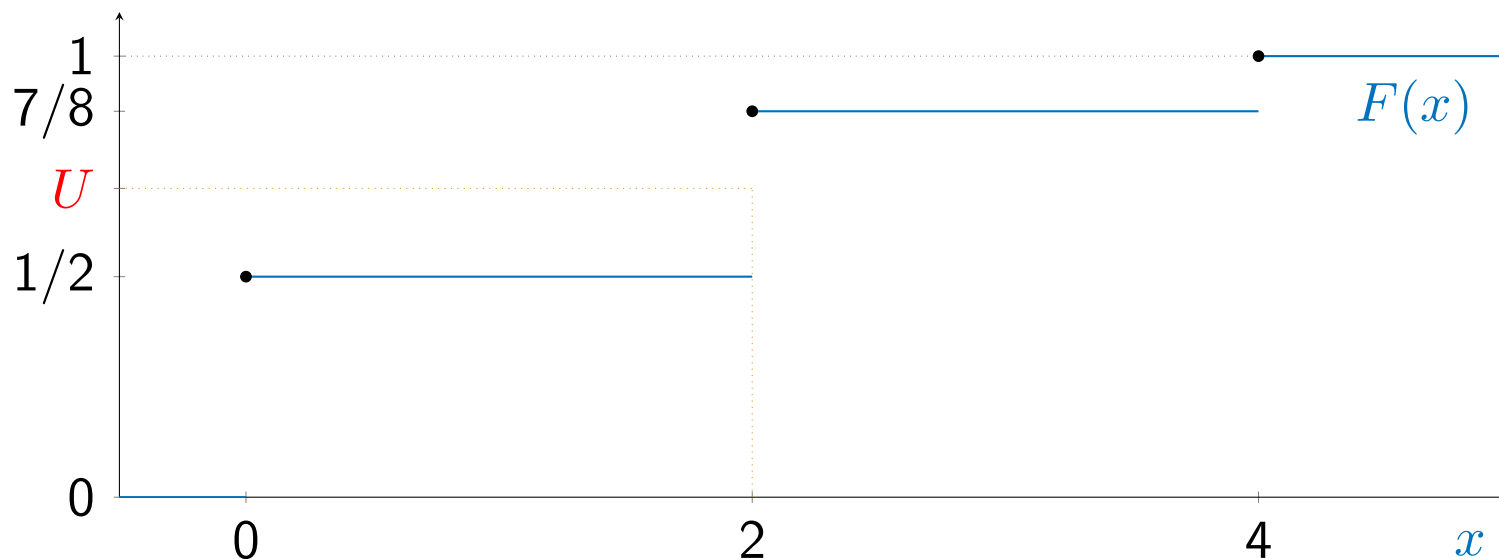
$$\begin{aligned} U &= 1 - \exp[-(\lambda X)^\alpha] \\ \exp[-(\lambda X)^\alpha] &= 1 - U \\ (\lambda X)^\alpha &= -\ln(1 - U) \\ \lambda X &= [-\ln(1 - U)]^{1/\alpha} \\ X &= [-\ln(1 - U)]^{1/\alpha} / \lambda = F^{-1}(U). \end{aligned}$$

Cas particulier: si $\alpha = 1$, $X \sim \text{Exponentielle}(\lambda)$.

Pour générer: $X = F^{-1}(U) = -\ln(1 - U)/\lambda$.

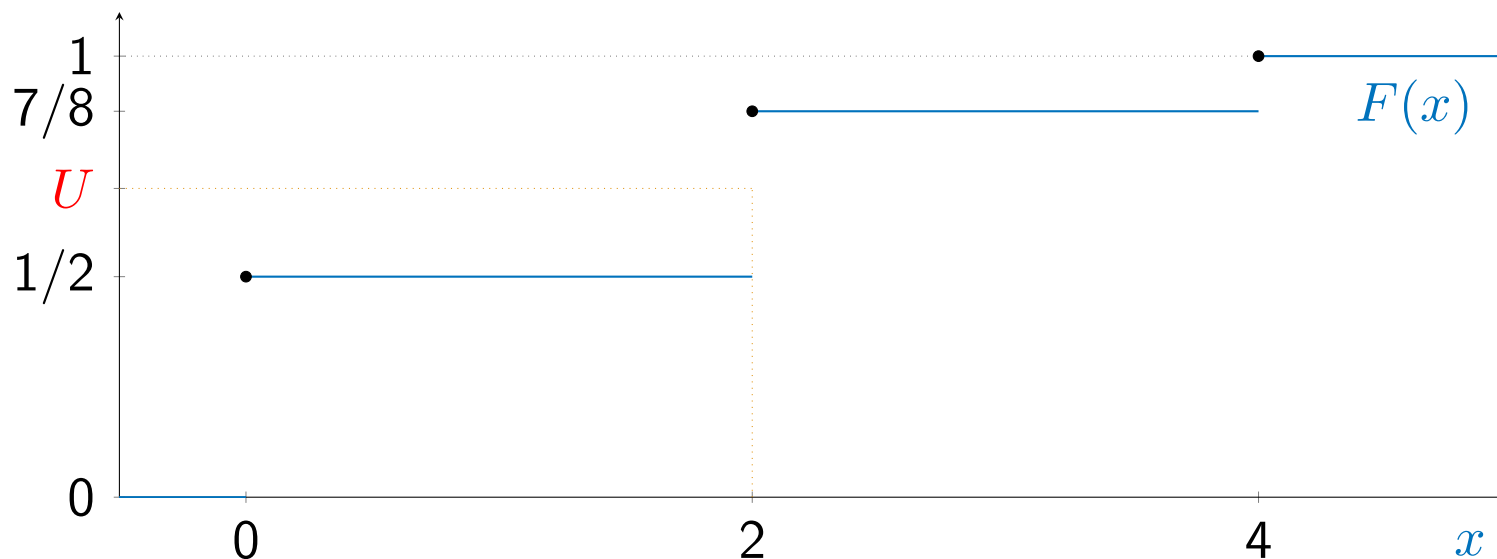
Exemple: Soit $\mathbb{P}[X = i] = p_i$ où $p_0 = 1/2$, $p_2 = 3/8$, $p_4 = 1/8$, et $p_i = 0$ ailleurs.

Inversion: retourner **0** si $U < 1/2$, **2** si $1/2 \leq U < 7/8$, et **4** si $U \geq 7/8$.



Exemple: Soit $\mathbb{P}[X = i] = p_i$ où $p_0 = 1/2$, $p_2 = 3/8$, $p_4 = 1/8$, et $p_i = 0$ ailleurs.

Inversion: retourner **0** si $U < 1/2$, **2** si $1/2 \leq U < 7/8$, et **4** si $U \geq 7/8$.

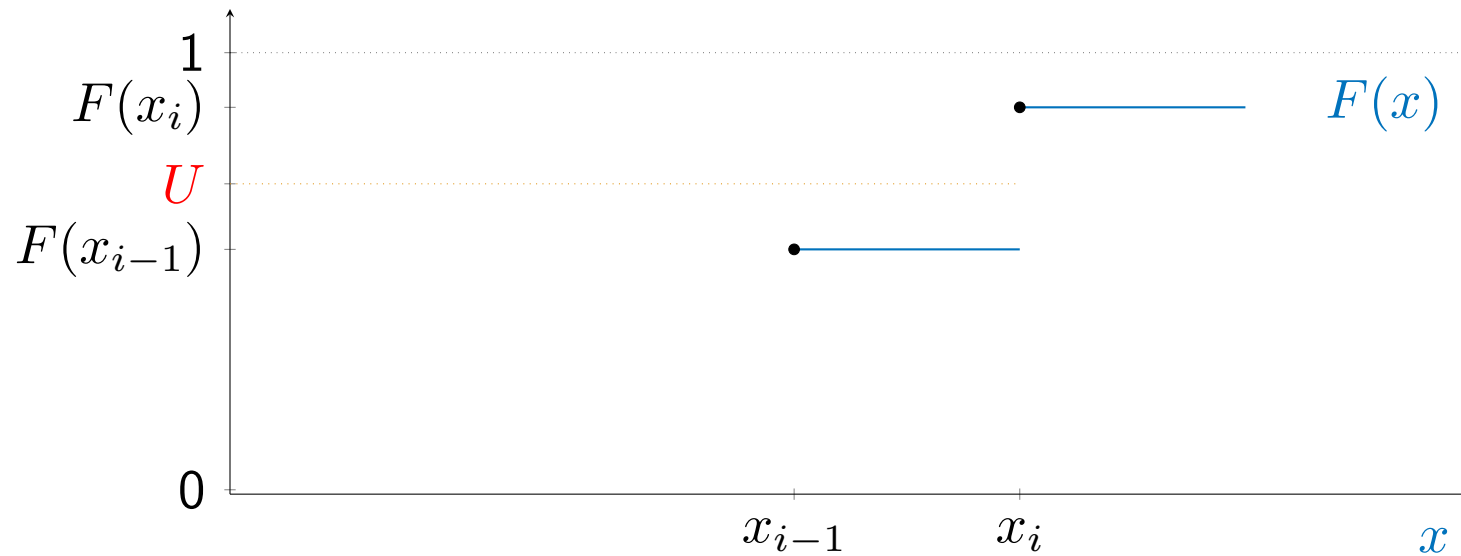


Loi discrète: $\mathbb{P}[X = x_i] = p_i$ pour $i = 0, 1, \dots$

On a $F(x_i) = p_0 + \dots + p_i$ et (disons) $F(x_{-1}) = 0$.

L'inversion retourne $X = x_i$ ssi

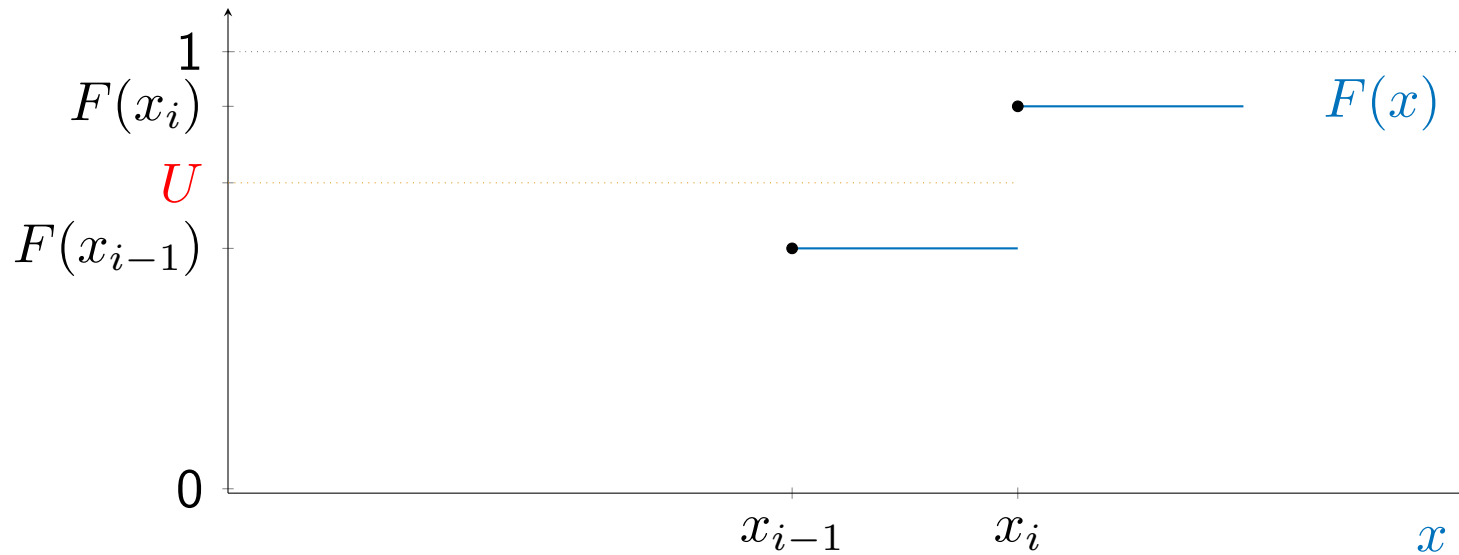
$$F(x_i) \geq U > F(x_{i-1}).$$



Exemple: $X \sim \text{Géométrique}(p)$, où $0 < p < 1$.

$\mathbb{P}[X = x] = p(1 - p)^x$ pour $x = 0, 1, 2, \dots$ et

$F(x) = 1 - (1 - p)^{\lfloor x+1 \rfloor}$ pour $x \geq 0$.



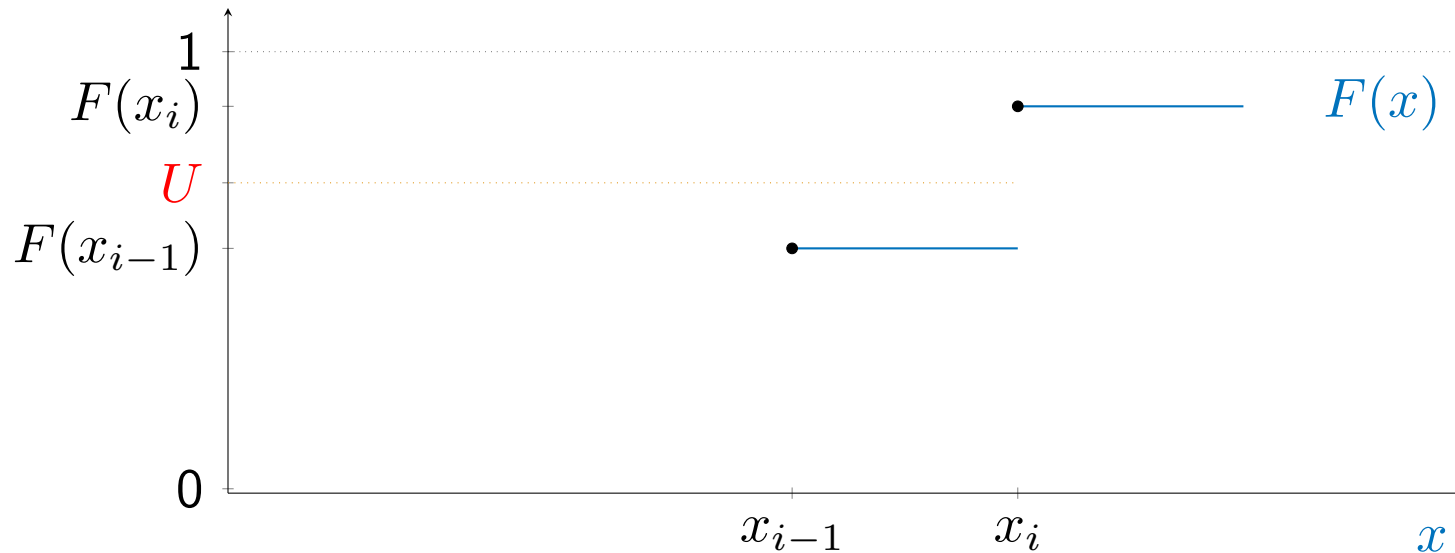
Exemple: $X \sim \text{Géométrique}(p)$, où $0 < p < 1$.

$\mathbb{P}[X = x] = p(1 - p)^x$ pour $x = 0, 1, 2, \dots$ et

$F(x) = 1 - (1 - p)^{\lfloor x+1 \rfloor}$ pour $x \geq 0$.

Pour $x \geq 0$ entier, $F(x) = 1 - (1 - p)^{x+1}$.

L'inversion doit retourner $X = x$ ssi $F(x) \geq U > F(x - 1)$,



Exemple: $X \sim \text{Géométrique}(p)$, où $0 < p < 1$.

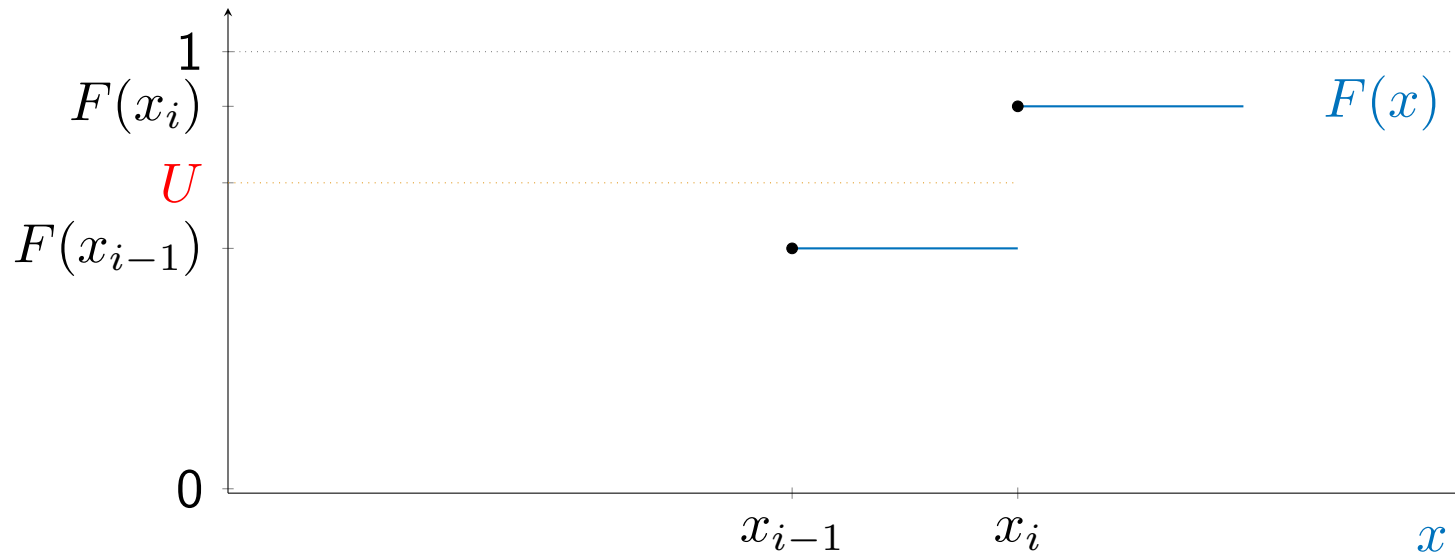
$\mathbb{P}[X = x] = p(1 - p)^x$ pour $x = 0, 1, 2, \dots$ et

$F(x) = 1 - (1 - p)^{\lfloor x+1 \rfloor}$ pour $x \geq 0$.

Pour $x \geq 0$ entier, $F(x) = 1 - (1 - p)^{x+1}$.

L'inversion doit retourner $X = x$ ssi $F(x) \geq U > F(x - 1)$, que l'on peut réécrire

$$1 - (1 - p)^{x+1} \geq U > 1 - (1 - p)^x,$$



Exemple: $X \sim \text{Géométrique}(p)$, où $0 < p < 1$.

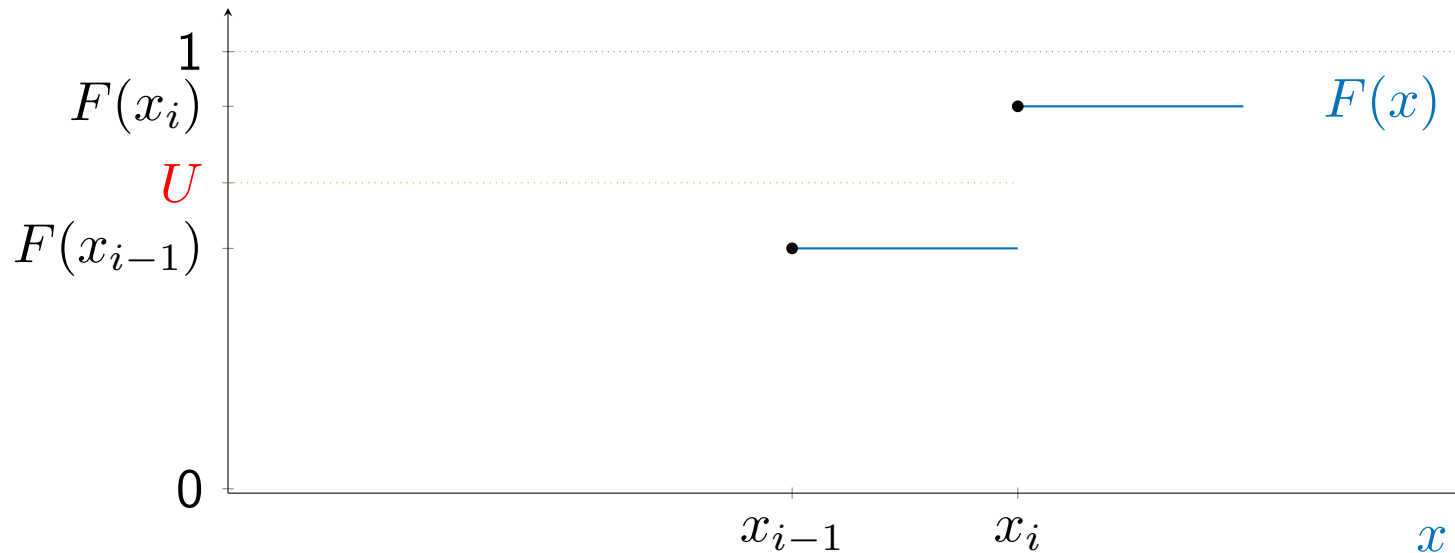
$\mathbb{P}[X = x] = p(1 - p)^x$ pour $x = 0, 1, 2, \dots$ et

$F(x) = 1 - (1 - p)^{\lfloor x+1 \rfloor}$ pour $x \geq 0$.

Pour $x \geq 0$ entier, $F(x) = 1 - (1 - p)^{x+1}$.

L'inversion doit retourner $X = x$ ssi $F(x) \geq U > F(x - 1)$, que l'on peut réécrire

$$\begin{aligned} 1 - (1 - p)^{x+1} &\geq U > 1 - (1 - p)^x, & \text{ou} \\ -(x + 1) \ln(1 - p) &\geq -\ln(1 - U) > -x \ln(1 - p), \end{aligned}$$



Exemple: $X \sim \text{Géométrique}(p)$, où $0 < p < 1$.

$\mathbb{P}[X = x] = p(1 - p)^x$ pour $x = 0, 1, 2, \dots$ et

$F(x) = 1 - (1 - p)^{\lfloor x+1 \rfloor}$ pour $x \geq 0$.

Pour $x \geq 0$ entier, $F(x) = 1 - (1 - p)^{x+1}$.

L'inversion doit retourner $X = x$ ssi $F(x) \geq U > F(x - 1)$, que l'on peut réécrire

$$\begin{aligned} 1 - (1 - p)^{x+1} &\geq U > 1 - (1 - p)^x, && \text{ou} \\ -(x + 1) \ln(1 - p) &\geq -\ln(1 - U) > -x \ln(1 - p), && \text{ou} \\ (x + 1) &\geq \ln(1 - U) / \ln(1 - p) > x, \end{aligned}$$

ce qui donne $x = \lceil \ln(1 - U) / \ln(1 - p) \rceil - 1$.

Avec probabilité 1, c'est la même chose que retourner $X = \lfloor \ln(1 - U) / \ln(1 - p) \rfloor$.

- Dans plusieurs cas (normale, Student, chi-deux, etc.), pas de formule pour F^{-1} ,⁵² mais approximation numérique.

- Dans plusieurs cas (normale, Student, chi-deux, etc.), pas de formule pour F^{-1} , mais approximation numérique.
- Plus difficile lorsque la **forme** de F dépend des paramètres (beta, gamma, par exemple).

- Dans plusieurs cas (normale, Student, chi-deux, etc.), pas de formule pour F^{-1} , mais approximation numérique.
- Plus difficile lorsque la **forme** de F dépend des paramètres (beta, gamma, par exemple).
- Inversion préférable car monotone (on verra pourquoi plus tard).

- Dans plusieurs cas (normale, Student, chi-deux, etc.), pas de formule pour F^{-1} , mais approximation numérique.
- Plus difficile lorsque la **forme** de F dépend des paramètres (beta, gamma, par exemple).
- Inversion préférable car monotone (on verra pourquoi plus tard).
- Mais autres méthodes parfois beaucoup plus rapides.

Intégration par Monte Carlo

Les problèmes de simulation dont nous avons discuté à date se ramènent souvent à estimer une intégrale de la forme

$$\mu = \mathbb{E}[X] = \int_0^1 \cdots \int_0^1 f(u_1, \dots, u_s) du_1 \cdots du_s = \int_{[0,1)^s} f(\mathbf{u}) d\mathbf{u} = \mathbb{E}[f(\mathbf{U})],$$

où $f : [0, 1)^s \rightarrow \mathbb{R}$ et \mathbf{u} est dans $[0, 1)^s$.

La dimension s peut parfois être aléatoire ou infinie.

La fonction f est souvent très compliquée.

μ est la valeur moyenne de f sur $[0, 1)^s$.

Intégration par Monte Carlo

Les problèmes de simulation dont nous avons discuté à date se ramènent souvent à estimer une intégrale de la forme

$$\mu = \mathbb{E}[X] = \int_0^1 \cdots \int_0^1 f(u_1, \dots, u_s) du_1 \cdots du_s = \int_{[0,1)^s} f(\mathbf{u}) d\mathbf{u} = \mathbb{E}[f(\mathbf{U})],$$

où $f : [0, 1)^s \rightarrow \mathbb{R}$ et \mathbf{u} est dans $[0, 1)^s$.

La dimension s peut parfois être aléatoire ou infinie.

La fonction f est souvent très compliquée.

μ est la valeur moyenne de f sur $[0, 1)^s$.

Exemple: réseau d'activités stochastique à 13 activités.

On a $s = 13$. À un point $(u_1, \dots, u_{13}) \in [0, 1)^{13}$ correspond un vecteur $(V_1, \dots, V_{13}) = (F_1^{-1}(u_1), \dots, F_{13}^{-1}(u_{13}))$ de durées des activités, et une durée du projet T .

Intégration par Monte Carlo

Les problèmes de simulation dont nous avons discuté à date se ramènent souvent à estimer une intégrale de la forme

$$\mu = \mathbb{E}[X] = \int_0^1 \cdots \int_0^1 f(u_1, \dots, u_s) du_1 \cdots du_s = \int_{[0,1]^s} f(\mathbf{u}) d\mathbf{u} = \mathbb{E}[f(\mathbf{U})],$$

où $f : [0, 1]^s \rightarrow \mathbb{R}$ et \mathbf{u} est dans $[0, 1]^s$.

La dimension s peut parfois être aléatoire ou infinie.

La fonction f est souvent très compliquée.

μ est la valeur moyenne de f sur $[0, 1]^s$.

Exemple: réseau d'activités stochastique à 13 activités.

On a $s = 13$. À un point $(u_1, \dots, u_{13}) \in [0, 1]^{13}$ correspond un vecteur $(V_1, \dots, V_{13}) = (F_1^{-1}(u_1), \dots, F_{13}^{-1}(u_{13}))$ de durées des activités, et une durée du projet T . On a $f(u_1, \dots, u_{13}) = 1$ si $T > x$ et $f(u_1, \dots, u_{13}) = 0$ sinon.

Exemple: option asiatique.

Supposons que $Z_j = \Phi^{-1}(U_j)$ où les U_j sont i.i.d. $U(0, 1)$. On a ici $s = d$ et

$$v(s, T) = e^{-rT} \int_{[0,1]^s} \max \left(0, \frac{1}{s} \sum_{i=1}^s S(0) \cdot \exp \left[(r - \sigma^2/2)t_i + \sigma \sum_{j=1}^i \sqrt{t_j - t_{j-1}} \Phi^{-1}(u_j) \right] - K \right) du_1 \dots du_s.$$

Si s ne dépasse pas 3 ou 4: méthodes d'intégration numériques.

Si s ne dépasse pas 3 ou 4: méthodes d'intégration numériques.

Si s est grand, estimateur de Monte Carlo:

$$\hat{\mu}_n = \frac{1}{n} \sum_{i=1}^n f(\mathbf{U}_i),$$

où n est la taille de l'échantillon et $\mathbf{U}_1, \dots, \mathbf{U}_n$ sont i.i.d. uniformes sur $[0, 1)^s$.

Si s ne dépasse pas 3 ou 4: méthodes d'intégration numériques.

Si s est grand, estimateur de **Monte Carlo**:

$$\hat{\mu}_n = \frac{1}{n} \sum_{i=1}^n f(\mathbf{U}_i),$$

où n est la taille de l'échantillon et $\mathbf{U}_1, \dots, \mathbf{U}_n$ sont i.i.d. uniformes sur $[0, 1)^s$.

On a

$$\mathbb{E}[\hat{\mu}_n] = \mathbb{E}[f(\mathbf{U}_i)] = \int_{[0,1)^s} f(\mathbf{u}) d\mathbf{u} = \mu$$

$$\text{Var}[\hat{\mu}_n] = \frac{1}{n} [\mathbb{E}[f^2(\mathbf{U}_i)] - \mu^2] = \frac{1}{n} \text{Var}[X] = \frac{\sigma^2}{n}$$

$$\text{où } \sigma^2 = \int_{[0,1)^s} f^2(\mathbf{u}) d\mathbf{u} - \mu^2 = \text{Var}[f(\mathbf{U}_i)].$$

La variance peut être infinie. Exemple: $s = 1$ et $f(u) = u^{-1/2}$, on a

$$\int_0^1 f^2(u) du = \int_0^1 (1/u) du = \infty.$$

Convergence

Théorème. Supposons que $\sigma^2 < \infty$. Lorsque $n \rightarrow \infty$:

(i) **Loi forte des grands nombres:** $\lim_{n \rightarrow \infty} \hat{\mu}_n = \mu$ avec probabilité 1.

Convergence

Théorème. Supposons que $\sigma^2 < \infty$. Lorsque $n \rightarrow \infty$:

- (i) Loi forte des grands nombres: $\lim_{n \rightarrow \infty} \hat{\mu}_n = \mu$ avec probabilité 1.
- (ii) Théorème de la limite centrale:

$$\frac{\sqrt{n}(\hat{\mu}_n - \mu)}{\sigma} \Rightarrow N(0, 1), \text{ i.e.,}$$
$$\lim_{n \rightarrow \infty} \mathbb{P} \left[\frac{\sqrt{n}(\hat{\mu}_n - \mu)}{\sigma} \leq x \right] = \Phi(x) = \mathbb{P}[Z \leq x]$$

pour tout $x \in \mathbb{R}$, où $Z \sim N(0, 1)$ et $\Phi(\cdot)$ sa fonction de répartition.

Convergence

Théorème. Supposons que $\sigma^2 < \infty$. Lorsque $n \rightarrow \infty$:

- (i) Loi forte des grands nombres: $\lim_{n \rightarrow \infty} \hat{\mu}_n = \mu$ avec probabilité 1.
- (ii) Théorème de la limite centrale:

$$\frac{\sqrt{n}(\hat{\mu}_n - \mu)}{\sigma} \Rightarrow N(0, 1), \text{ i.e.,}$$

$$\lim_{n \rightarrow \infty} \mathbb{P} \left[\frac{\sqrt{n}(\hat{\mu}_n - \mu)}{\sigma} \leq x \right] = \Phi(x) = \mathbb{P}[Z \leq x]$$

pour tout $x \in \mathbb{R}$, où $Z \sim N(0, 1)$ et $\Phi(\cdot)$ sa fonction de répartition.

La propriété (ii) tient aussi si on remplace σ^2 par son estimateur sans biais

$$S_n^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2 = \frac{1}{n-1} \left(\sum_{i=1}^n X_i^2 - n(\bar{X}_n)^2 \right),$$

où $X_i = f(\mathbf{U}_i)$.

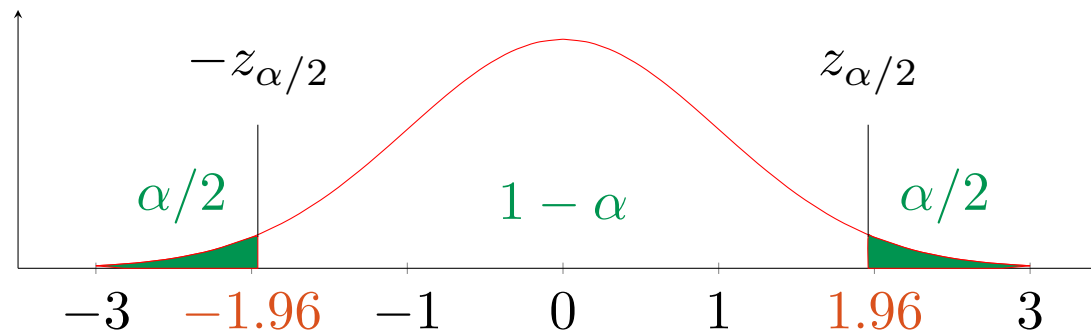
Ainsi, pour n grand et un niveau de confiance $1 - \alpha$, on veut

$$\mathbb{P}[\hat{\mu}_n - \mu \leq xS_n/\sqrt{n}] = \mathbb{P}[\sqrt{n}(\hat{\mu}_n - \mu)/S_n \leq x] \approx \Phi(x) = 1 - \alpha/2.$$

Intervalle de confiance au niveau α :

$$(\hat{\mu}_n \pm z_{\alpha/2}S_n/\sqrt{n}), \text{ où } z_{\alpha/2} = \Phi^{-1}(1 - \alpha/2).$$

Exemple: $z_{\alpha/2} \approx 1.96$ pour $\alpha = 0.05$.



La largeur de l'intervalle de confiance est asymptotiquement proportionnelle à σ/\sqrt{n} , donc converge en $O(n^{-1/2})$.

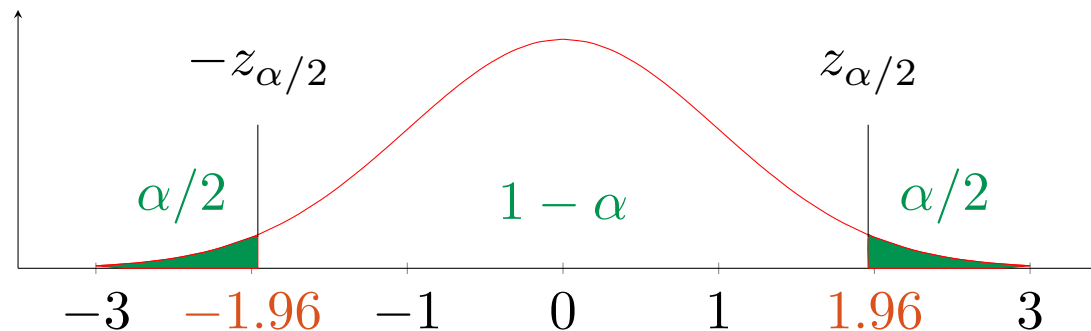
Ainsi, pour n grand et un niveau de confiance $1 - \alpha$, on veut

$$\mathbb{P}[\hat{\mu}_n - \mu \leq xS_n/\sqrt{n}] = \mathbb{P}[\sqrt{n}(\hat{\mu}_n - \mu)/S_n \leq x] \approx \Phi(x) = 1 - \alpha/2.$$

Intervalle de confiance au niveau α :

$$(\hat{\mu}_n \pm z_{\alpha/2}S_n/\sqrt{n}), \text{ où } z_{\alpha/2} = \Phi^{-1}(1 - \alpha/2).$$

Exemple: $z_{\alpha/2} \approx 1.96$ pour $\alpha = 0.05$.



La largeur de l'intervalle de confiance est asymptotiquement proportionnelle à σ/\sqrt{n} , donc converge en $O(n^{-1/2})$.

Si n est petit et les X_i suivent la loi normale, alors

$$\sqrt{n}(\hat{\mu}_n - \mu)/S_n \sim \text{Student}(n - 1).$$

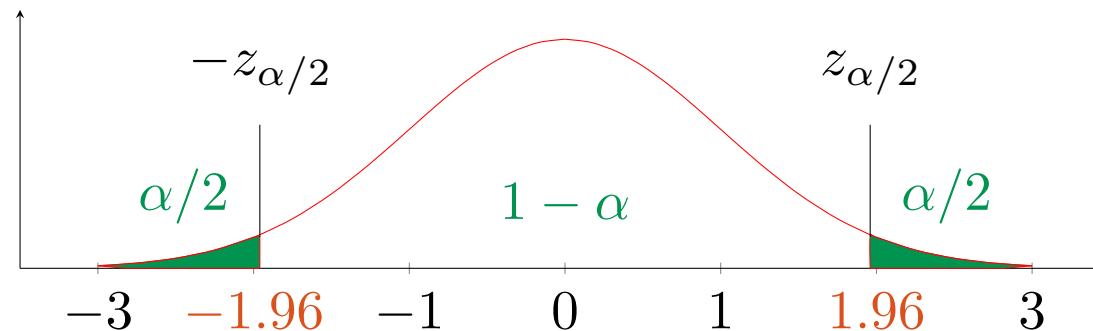
Ainsi, pour n grand et un niveau de confiance $1 - \alpha$, on veut

$$\mathbb{P}[\hat{\mu}_n - \mu \leq xS_n/\sqrt{n}] = \mathbb{P}[\sqrt{n}(\hat{\mu}_n - \mu)/S_n \leq x] \approx \Phi(x) = 1 - \alpha/2.$$

Intervalle de confiance au niveau α :

$$(\hat{\mu}_n \pm z_{\alpha/2}S_n/\sqrt{n}), \text{ où } z_{\alpha/2} = \Phi^{-1}(1 - \alpha/2).$$

Exemple: $z_{\alpha/2} \approx 1.96$ pour $\alpha = 0.05$.



La largeur de l'intervalle de confiance est asymptotiquement proportionnelle à σ/\sqrt{n} , donc converge en $O(n^{-1/2})$.

Si n est petit et les X_i suivent la loi normale, alors

$$\sqrt{n}(\hat{\mu}_n - \mu)/S_n \sim \text{Student}(n - 1).$$

Mais si la loi des X_i est très asymétrique, ces intervalles ne sont plus valides même comme approximations.

Example: On veut estimer $\mu = \mathbb{P}[0 \leq Z \leq 2]$ où $Z \sim N(0, 1)$.

Pour $i = 1, \dots, n$,

Générer $Z_i \sim N(0, 1)$;

Poser $X_i = \mathbb{I}[0 \leq Z_i \leq 2]$;

Calculer $\hat{\mu}_n = \bar{X}_n$ et S_n^2 , puis un intervalle de confiance pour μ .

Example: On veut estimer $\mu = \mathbb{P}[0 \leq Z \leq 2]$ où $Z \sim N(0, 1)$.

Pour $i = 1, \dots, n$,

Générer $Z_i \sim N(0, 1)$;

Poser $X_i = \mathbb{I}[0 \leq Z_i \leq 2]$;

Calculer $\hat{\mu}_n = \bar{X}_n$ et S_n^2 , puis un intervalle de confiance pour μ .

Autre algorithme MC pour estimer μ : Si ϕ est la densité normale standard, on a

$$\mu = \int_0^2 \phi(y) dy = \int_0^1 2\phi(2u) du = \int_0^1 f(u) du$$

via le changement de variable $y = 2u$ et $f(u) = 2\phi(2u)$.

Example: On veut estimer $\mu = \mathbb{P}[0 \leq Z \leq 2]$ où $Z \sim N(0, 1)$.

Pour $i = 1, \dots, n$,

Générer $Z_i \sim N(0, 1)$;

Poser $X_i = \mathbb{I}[0 \leq Z_i \leq 2]$;

Calculer $\hat{\mu}_n = \bar{X}_n$ et S_n^2 , puis un intervalle de confiance pour μ .

Autre algorithme MC pour estimer μ : Si ϕ est la densité normale standard, on a

$$\mu = \int_0^2 \phi(y) dy = \int_0^1 2\phi(2u) du = \int_0^1 f(u) du$$

via le changement de variable $y = 2u$ et $f(u) = 2\phi(2u)$.

Pour $i = 1, \dots, n$,

Générer $U_i \sim U(0, 1)$;

Poser $X_i = 2\phi(2U_i)$;

Calculer $\hat{\mu}_n = \bar{X}_n$ et S_n^2 , puis un intervalle de confiance pour μ .

Exemple: Réseau d'activités stochastique. On a

$$X_i = \mathbb{I}[T_i > x],$$

$$\hat{\mu}_n = \bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i = \frac{Y}{n},$$

$$S_n^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - Y/n)^2 = \frac{Y(1 - Y/n)}{n-1}.$$

Intervalle de confiance à 95%: $(Y/n \pm 1.96S_n/\sqrt{n})$.

Raisonné si $\mathbb{E}[Y] = n\mu$ n'est pas trop proche de 0 ou 1.

Exemple: Réseau d'activités stochastique. On a

$$X_i = \mathbb{I}[T_i > x],$$

$$\hat{\mu}_n = \bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i = \frac{Y}{n},$$

$$S_n^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - Y/n)^2 = \frac{Y(1 - Y/n)}{n-1}.$$

Intervalle de confiance à 95%: $(Y/n \pm 1.96S_n/\sqrt{n})$.

Raisonnable si $\mathbb{E}[Y] = n\mu$ n'est pas trop proche de 0 ou 1.

En fait, $Y \sim \text{Binomiale}(n, \mu)$.

Supposons par exemple que l'on observe $Y = 882$. On a alors:

$$\bar{X}_n = 882/1000 = 0.882;$$

$$S_n^2 = \bar{X}_n(1 - \bar{X}_n)n/(n - 1) \approx 0.1042.$$

Supposons par exemple que l'on observe $Y = 882$. On a alors:

$$\bar{X}_n = 882/1000 = 0.882;$$

$$S_n^2 = \bar{X}_n(1 - \bar{X}_n)n/(n - 1) \approx 0.1042.$$

On obtient l'intervalle de confiance à 95%:

$$(\bar{X}_n \pm 1.96S_n/\sqrt{n}) \approx (0.882 \pm 0.020) = (0.862, 0.902).$$

Supposons par exemple que l'on observe $Y = 882$. On a alors:

$$\bar{X}_n = 882/1000 = 0.882;$$

$$S_n^2 = \bar{X}_n(1 - \bar{X}_n)n/(n - 1) \approx 0.1042.$$

On obtient l'intervalle de confiance à 95%:

$$(\bar{X}_n \pm 1.96S_n/\sqrt{n}) \approx (0.882 \pm 0.020) = (0.862, 0.902).$$

Notre estimateur de μ a donc deux chiffres significatifs: $\mu \approx 0.88$.

Le "2" dans 0.882 n'est pas significatif.

Supposons par exemple que l'on observe $Y = 882$. On a alors:

$$\bar{X}_n = 882/1000 = 0.882;$$

$$S_n^2 = \bar{X}_n(1 - \bar{X}_n)n/(n - 1) \approx 0.1042.$$

On obtient l'intervalle de confiance à 95%:

$$(\bar{X}_n \pm 1.96S_n/\sqrt{n}) \approx (0.882 \pm 0.020) = (0.862, 0.902).$$

Notre estimateur de μ a donc deux chiffres significatifs: $\mu \approx 0.88$.

Le "2" dans 0.882 n'est pas significatif.

Il faut éviter de donner des chiffres non significatifs dans les résultats, car cela peut induire en erreur.

Avantages de Monte Carlo pour l'intégration:

- Ne requiert qu'une hypothèse très faible sur f .
- Le taux de convergence de l'erreur ne dépend pas de la dimension s , contrairement aux méthodes d'intégration numérique classiques.
- On peut estimer l'erreur de manière probabiliste (intervalles de confiance). Les méthodes d'intégration numériques donnent des bornes sur l'erreur, mais peu pratiques.

Efficacité des estimateurs

Soit X estimateur de μ .

$$\beta = \mathbb{E}[X] - \mu \quad \text{biais}$$

$$\sigma^2 = \text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] \quad \text{variance}$$

$$\text{MSE}[X] = \mathbb{E}[(X - \mu)^2] = \beta^2 + \sigma^2 \quad \text{erreur quadratique moyenne}$$

$$\sqrt{\text{MSE}[X]} \quad \text{erreur absolue}$$

$$\text{RE}[X] = \sqrt{\text{MSE}[X]} / |\mu|, \quad \text{pour } \mu \neq 0 \quad \text{erreur relative}$$

Soit $C(X)$ l'espérance mathématique du temps de calcul de X .
On définit l'efficacité de l'estimateur X par

$$\text{Eff}(X) = \frac{1}{C(X) \cdot \text{MSE}(X)}.$$

On dit que X est plus efficace que Y si $\text{Eff}(X) > \text{Eff}(Y)$.

L'amélioration de l'efficacité consiste à trouver des estimateurs plus efficaces, en ce sens.

Peut être obtenu en diminuant la variance, ou le biais, ou le temps de calcul.

D'où vient cette définition?

Soit X_1, \dots, X_n i.i.d. et $C(\bar{X}_n) = \kappa n$.

Soit X_1, \dots, X_n i.i.d. et $C(\bar{X}_n) = \kappa n$.
On a $\text{Var}[\bar{X}_n] = \text{Var}[X_i]/n = \sigma^2/n$.

Soit X_1, \dots, X_n i.i.d. et $C(\bar{X}_n) = \kappa n$.

On a $\text{Var}[\bar{X}_n] = \text{Var}[X_i]/n = \sigma^2/n$.

Si $\mathbb{E}[X_i] = \mu$, alors la mesure de l'efficacité est

$$\text{Eff}[\bar{X}_n] = \frac{1}{C(\bar{X}_n) \cdot \text{MSE}(\bar{X}_n)} = \frac{1}{\kappa n \sigma^2 / n} = \frac{1}{\kappa \sigma^2}.$$

Elle ne dépend pas de n .

Soit X_1, \dots, X_n i.i.d. et $C(\bar{X}_n) = \kappa n$.

On a $\text{Var}[\bar{X}_n] = \text{Var}[X_i]/n = \sigma^2/n$.

Si $\mathbb{E}[X_i] = \mu$, alors la mesure de l'efficacité est

$$\text{Eff}[\bar{X}_n] = \frac{1}{C(\bar{X}_n) \cdot \text{MSE}(\bar{X}_n)} = \frac{1}{\kappa n \sigma^2 / n} = \frac{1}{\kappa \sigma^2}.$$

Elle ne dépend pas de n .

Par contre, si $\mathbb{E}[X_i] - \mu = \beta \neq 0$,

$$\text{Eff}[\bar{X}_n] = \frac{1}{\kappa n (\beta^2 + \sigma^2/n)} = \frac{1}{\kappa (n\beta^2 + \sigma^2)}$$

diminue avec n , et on ne peut généralement pas estimer β .

On pourra s'intéresser à étudier le comportement asymptotique de l'efficacité.

Soit X_1, \dots, X_n i.i.d. et $C(\bar{X}_n) = \kappa n$.

On a $\text{Var}[\bar{X}_n] = \text{Var}[X_i]/n = \sigma^2/n$.

Si $\mathbb{E}[X_i] = \mu$, alors la mesure de l'efficacité est

$$\text{Eff}[\bar{X}_n] = \frac{1}{C(\bar{X}_n) \cdot \text{MSE}(\bar{X}_n)} = \frac{1}{\kappa n \sigma^2 / n} = \frac{1}{\kappa \sigma^2}.$$

Elle ne dépend pas de n .

Par contre, si $\mathbb{E}[X_i] - \mu = \beta \neq 0$,

$$\text{Eff}[\bar{X}_n] = \frac{1}{\kappa n (\beta^2 + \sigma^2/n)} = \frac{1}{\kappa (n\beta^2 + \sigma^2)}$$

diminue avec n , et on ne peut généralement pas estimer β .

On pourra s'intéresser à étudier le comportement asymptotique de l'efficacité.

En principe on peut estimer σ^2 par S_n^2 , mais $\text{RE}[S_n^2]$ est parfois énorme.

Ainsi, le coût de calcul, la variance et le biais ne disent pas tout.

Ils ne tiennent pas compte, par exemple, des moments d'ordre plus élevé de l'estimateur, ou encore de la qualité de l'estimateur de variance.

Notre définition d'efficacité est un compromis.

Exemple: événements rares.

On veut estimer $p = \mathbb{P}\{A\}$ où A est un événement rare (p est proche de 0).
La variable binaire $X = \mathbb{I}[A]$ est un estimateur sans biais de p ,
de variance (et MSE) $\text{Var}[X] = \text{MSE}[X] = p(1 - p)$.

Exemple: événements rares.

On veut estimer $p = \mathbb{P}\{A\}$ où A est un événement rare (p est proche de 0).

La variable binaire $X = \mathbb{I}[A]$ est un estimateur sans biais de p ,
de variance (et MSE) $\text{Var}[X] = \text{MSE}[X] = p(1 - p)$.

Si p est petit, $\text{Var}[X] \approx p$ est petite.

Exemple: événements rares.

On veut estimer $p = \mathbb{P}\{A\}$ où A est un événement rare (p est proche de 0).

La variable binaire $X = \mathbb{I}[A]$ est un estimateur sans biais de p ,
de variance (et MSE) $\text{Var}[X] = \text{MSE}[X] = p(1 - p)$.

Si p est petit, $\text{Var}[X] \approx p$ est petite.

Mais on veut $\text{Var}[X] \ll p$.

En effet, l'estimateur trivial $Y = 0$ donne $\text{Var}[Y] = 0$ et $\text{MSE}[Y] = p^2$.

Exemple: événements rares.

On veut estimer $p = \mathbb{P}\{A\}$ où A est un événement rare (p est proche de 0).

La variable binaire $X = \mathbb{I}[A]$ est un estimateur sans biais de p ,
de variance (et MSE) $\text{Var}[X] = \text{MSE}[X] = p(1 - p)$.

Si p est petit, $\text{Var}[X] \approx p$ est petite.

Mais on veut $\text{Var}[X] \ll p$.

En effet, l'estimateur trivial $Y = 0$ donne $\text{Var}[Y] = 0$ et $\text{MSE}[Y] = p^2$.

Si on prend \bar{X}_n , alors $\text{Var}[\bar{X}_n] = p(1 - p)/n$, de sorte que $\text{MSE}[\bar{X}_n] < \text{MSE}[Y]$
ssi $n > (1 - p)/p$.

Exemple: événements rares.

On veut estimer $p = \mathbb{P}\{A\}$ où A est un événement rare (p est proche de 0).

La variable binaire $X = \mathbb{I}[A]$ est un estimateur sans biais de p ,
de variance (et MSE) $\text{Var}[X] = \text{MSE}[X] = p(1 - p)$.

Si p est petit, $\text{Var}[X] \approx p$ est petite.

Mais on veut $\text{Var}[X] \ll p$.

En effet, l'estimateur trivial $Y = 0$ donne $\text{Var}[Y] = 0$ et $\text{MSE}[Y] = p^2$.

Si on prend \bar{X}_n , alors $\text{Var}[\bar{X}_n] = p(1 - p)/n$, de sorte que $\text{MSE}[\bar{X}_n] < \text{MSE}[Y]$
ssi $n > (1 - p)/p$.

Si $|p|$ est petit, il est plus approprié de considérer le **MSE relatif** $\text{MSE}[X]/p^2$, ou
l'**erreur relative** $\text{RE}[X]$, car la largeur relative d'un intervalle de confiance sur p est
à peu près proportionnelle à $\text{RE}[X]/\sqrt{n}$.

On a ici $\text{RE}[X] = \sqrt{(1 - p)/p} \rightarrow \infty$ lorsque $p \rightarrow 0$.

Exemple: événements rares.

On veut estimer $p = \mathbb{P}\{A\}$ où A est un événement rare (p est proche de 0).

La variable binaire $X = \mathbb{I}[A]$ est un estimateur sans biais de p ,
de variance (et MSE) $\text{Var}[X] = \text{MSE}[X] = p(1 - p)$.

Si p est petit, $\text{Var}[X] \approx p$ est petite.

Mais on veut $\text{Var}[X] \ll p$.

En effet, l'estimateur trivial $Y = 0$ donne $\text{Var}[Y] = 0$ et $\text{MSE}[Y] = p^2$.

Si on prend \bar{X}_n , alors $\text{Var}[\bar{X}_n] = p(1 - p)/n$, de sorte que $\text{MSE}[\bar{X}_n] < \text{MSE}[Y]$
ssi $n > (1 - p)/p$.

Si $|p|$ est petit, il est plus approprié de considérer le **MSE relatif** $\text{MSE}[X]/p^2$, ou
l'**erreur relative** $\text{RE}[X]$, car la largeur relative d'un intervalle de confiance sur p est
à peu près proportionnelle à $\text{RE}[X]/\sqrt{n}$.

On a ici $\text{RE}[X] = \sqrt{(1 - p)/p} \rightarrow \infty$ lorsque $p \rightarrow 0$.

Par exemple, si $p \approx 10^{-10}$, il faut $n \approx 10^{12}$ pour une erreur relative de 10%.

Autre point de vue du même problème: Si p est très petit (np est petit), on a de fortes chances d'avoir $X_1 = \dots = X_n = 0$, ce qui donne $\bar{X}_n = S_n^2 = 0$ et un intervalle de confiance de largeur 0.

Autre point de vue du même problème: Si p est très petit (np est petit), on a de fortes chances d'avoir $X_1 = \dots = X_n = 0$, ce qui donne $\bar{X}_n = S_n^2 = 0$ et un intervalle de confiance de largeur 0.

La probabilité d'un échec étant $1 - p$, on a $\mathbb{P}[\bar{X}_n = S_n^2 = 0] = (1 - p)^n \approx 1 - np$.

Autre point de vue du même problème: Si p est très petit (np est petit), on a de fortes chances d'avoir $X_1 = \dots = X_n = 0$, ce qui donne $\bar{X}_n = S_n^2 = 0$ et un intervalle de confiance de largeur 0.

La probabilité d'un échec étant $1 - p$, on a $\mathbb{P}[\bar{X}_n = S_n^2 = 0] = (1 - p)^n \approx 1 - np$.

Ici, $n\bar{X}_n$ suit la loi binomiale de paramètres (n, p) et l'approximation normale est bonne seulement si n et np sont grands.

Autre point de vue du même problème: Si p est très petit (np est petit), on a de fortes chances d'avoir $X_1 = \dots = X_n = 0$, ce qui donne $\bar{X}_n = S_n^2 = 0$ et un intervalle de confiance de largeur 0.

La probabilité d'un échec étant $1 - p$, on a $\mathbb{P}[\bar{X}_n = S_n^2 = 0] = (1 - p)^n \approx 1 - np$.

Ici, $n\bar{X}_n$ suit la loi binomiale de paramètres (n, p) et l'approximation normale est bonne seulement si n et np sont grands.

Si n est grand et np est petit: approx. par la loi de Poisson.

Autre point de vue du même problème: Si p est très petit (np est petit), on a de fortes chances d'avoir $X_1 = \dots = X_n = 0$, ce qui donne $\bar{X}_n = S_n^2 = 0$ et un intervalle de confiance de largeur 0.

La probabilité d'un échec étant $1 - p$, on a $\mathbb{P}[\bar{X}_n = S_n^2 = 0] = (1 - p)^n \approx 1 - np$.

Ici, $n\bar{X}_n$ suit la loi binomiale de paramètres (n, p) et l'approximation normale est bonne seulement si n et np sont grands.

Si n est grand et np est petit: approx. par la loi de Poisson.

On verra plus loin des techniques adaptées à ce contexte d'événements rares.

Estimateur succès/échec (“hit-or-miss”)

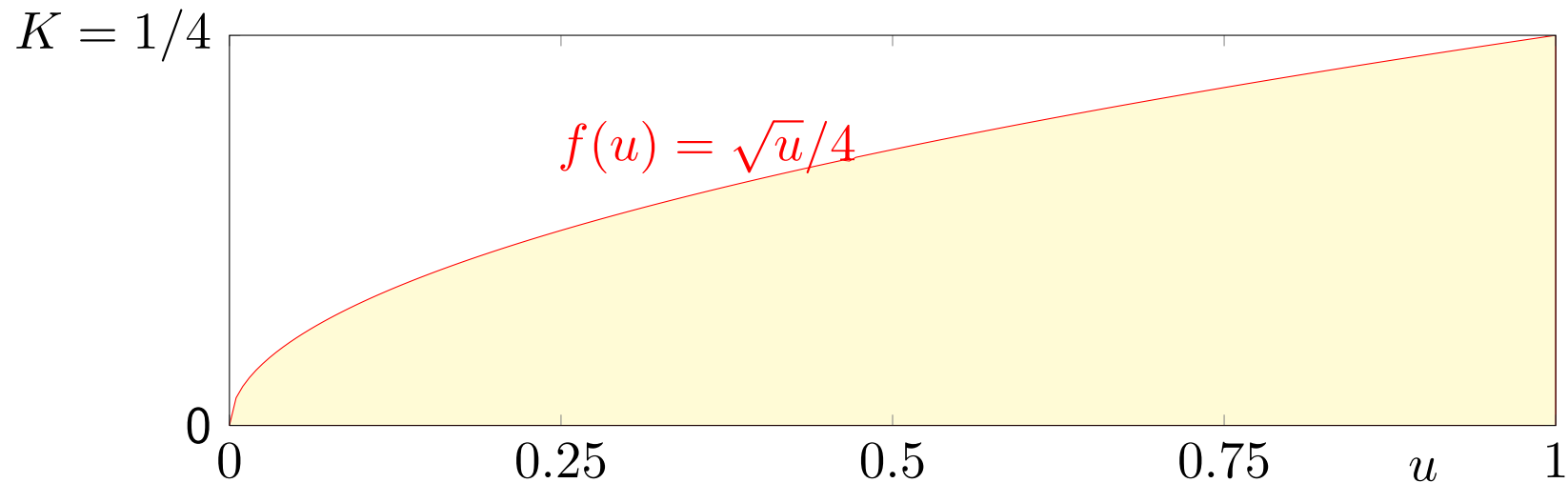
On suppose $0 \leq f(\mathbf{u}) \leq K$.

Soient $\mathbf{U}_i \sim U[0, 1)^s$ et $V_i \sim U(0, 1)$, $i = 1, \dots, n$, mutuellement indépendantes,

$$B_i = \mathbb{I}[V_i K \leq f(\mathbf{U}_i)] = \begin{cases} 1 & \text{si } V_i K \leq f(\mathbf{U}_i); \\ 0 & \text{sinon,} \end{cases}$$

$(V_i K \sim U[0, K])$ et

$$\tilde{\mu}_n = \frac{K}{n} \sum_{i=1}^n B_i.$$



Les B_i sont i.i.d. Bernoulli de paramètre $p = \mu/K$. Ainsi $\mathbb{E}[\tilde{\mu}_n] = Kp = \mu$ et

$$\text{Var}[\tilde{\mu}_n] = \frac{1}{n} [\mathbb{E}[(KB_i)^2] - \mu^2] = \frac{1}{n} [K^2p - \mu^2] = \mu(K - \mu)/n.$$

Les B_i sont i.i.d. Bernoulli de paramètre $p = \mu/K$. Ainsi $\mathbb{E}[\tilde{\mu}_n] = Kp = \mu$ et

$$\text{Var}[\tilde{\mu}_n] = \frac{1}{n} [\mathbb{E}[(KB_i)^2] - \mu^2] = \frac{1}{n} [K^2p - \mu^2] = \mu(K - \mu)/n.$$

On a $\int_{[0,1)^s} f^2(\mathbf{u})d\mathbf{u} \leq \int_{[0,1)^s} Kf(\mathbf{u})d\mathbf{u} = K\mu$, et donc

$$\text{Var}[\hat{\mu}_n] \leq (K\mu - \mu^2)/n = \text{Var}[\tilde{\mu}_n].$$

Les B_i sont i.i.d. Bernoulli de paramètre $p = \mu/K$. Ainsi $\mathbb{E}[\tilde{\mu}_n] = Kp = \mu$ et

$$\text{Var}[\tilde{\mu}_n] = \frac{1}{n} [\mathbb{E}[(KB_i)^2] - \mu^2] = \frac{1}{n} [K^2p - \mu^2] = \mu(K - \mu)/n.$$

On a $\int_{[0,1)^s} f^2(\mathbf{u})d\mathbf{u} \leq \int_{[0,1)^s} Kf(\mathbf{u})d\mathbf{u} = K\mu$, et donc

$$\text{Var}[\hat{\mu}_n] \leq (K\mu - \mu^2)/n = \text{Var}[\tilde{\mu}_n].$$

Mais il se peut que $\tilde{\mu}_n$ soit moins coûteux à calculer.