Simulation et pratique du logiciel

Khawla Boudjerda.

Departement of mathematics, Mohammed Seddik Ben Yahia university, Jijel, Algeria

La simulation est un processus qui consiste à imiter le comportement d'un système réel à l'aide d'un modèle mathématique, physique ou informatique. Elle permet d'analyser, de prédire ou d'optimiser le fonctionnement d'un système dans avoir à l'expérimenter directement dans le modèle réel.

Elleutilisée dans divers domaaines comme :

- **L'ingénierie** (test de structures, aérodynamique, circuits éléctronique).
- **L'informatique** (simulation des réseaux, intélligence artificielle).
- **La physiqueet les sciences naturelles** (modélisation de climat,réactions chimiques).
- **L'économie et la géstion** (prévisions financières, gestion des risques).
- **La santé** (formation médicale, essais cliniques vertuels).

Importance de la simulation

La simulation est outil puissant qui améliore l'éfficacité, la sécurité et la précision dans de nombreux secteurs :

- **Réduction des coûts** : Elle permet de tester des scénarios sans engager de ressources matérielles coûteuses, comme l'aéronotique ou l'industrie automobile.
- **sécurité** Elle offre un environnement sans danger pour tester des situations risquées, comme la formations des pilotes, la médecine ou la gestion des catastrophes.
- **Optimisation et prise de décision** en testant différents paramètres,la simulation aide à identifier les meilleurs solutions dans les domaines comme la logestique,la finance ou la géstion d'énergie.

- **Anticipation et prévision** Elle permet d'analyser des phénomènes complexes et de prévoir leurs évolutions, comme en météorologie, en épidémiologie ou en économie.
- **Innivation et conception** Dans l'ingénierie et la recherche, la simulation accélère le développement de nouveaux produits en testant virtuellement leurs performances.
- **Accessibilité et formation** Elle permet de former des personnes (pilotes, médecins, ingé,ieurs) à des situations réalistes sans danger ni coût élevé.

Pourquoi utiliser la simulation pour les variables aléatoires

- **Approximation numériques** : Certaines distribution ou intégrales sont trop compliquées à résoudre analytiquement, la simulation permet d'en obtenir des estimations.
- **Modélisation de systèmes réels** : Beaucoup de phyénomènes naturels ou industriels so,t inftuancé par l'aléatoire.

Génaration des variables aléatoires

Lorsque la fonction de distribution est une fonction inverse cunnue, la méthode de transformation inverse est le choix. Autrement, des méthodes spéciales de génération des variables aléatoires peuvent être utilisée.

Transformation inverse- Variables aléatoires continus

Theorem

- *Soit* X *une variable aléatoire de fonction de répartition* F*, donc* $F(x) = P(X \leq x)$.
- *Soit* F^{-1} *la fonction réciproque (inverse) de* F *:* $F(x) = y \iff F^{-1}(y) = x$.
- *Soit* $u \sim U[0,1]$*, alors la variable aléatoire* $F^{-1}(u)$ *a pour fonction de répartition* F*. Autrement dit* $F^{-1}(u)$ *suit la même loi de* X*.*

Démonstration.

Soit G la fonction de répartition de $F^{-1}(u)$

$$G(x) = P(F^{-1}(u) \leq x)$$

F estune fonction de répartitio, don F est strictement croissante, donc

$$G(x) = P(F(F^{-1}(u)) \leq F(x))$$

$F(F^{-1}(u)) = u$, donc : $G(x) = P(u \leq F(x))$
$u \sim U[0, 1]$, donc : $G(x) = P(u \leq F(x)) = F(x)$ $\square$

- La méthode de transformation inversée consiste générer des valeurs d'une variable aléatoire X continue de fonction de répartition F, pour laquelle F^{-1} est connue, à partir des échantillons de la loi uniforme $u \sim U[0,1]$.
- L'algorithme de transformation inversée peut être résumé comme suit :

  1- Générer une valeur de u tels que $u \sim U[0,1]$

  2- Calculer $x = F^{-1}(u)$

  3- Retourner x.

Example (La loi exponentielle)

Soit X une varaible aléatoire qui suit une loi exponentielle de paramètre λ : $X \sim Exp(\lambda)$.

La fonction de répartition de X : $F(x) = P(X \leq x) = 1 - e^{-\lambda x}$.

Donc $F^{-1}(x) = -\frac{ln(1-x)}{\lambda}$.

Si $X \sim Exp(\lambda)$, et $U \sim U[0,1]$, nous avons : $-\frac{ln(1-u)}{\lambda} \sim Exp(\lambda)$

Algorithme pour générer des valeurs d'une variable aléatoire $X \sim Exp(\lambda)$:

- Générer une valeur u_i de $u \sim U[0,1]$.
- Calculer $x_i = -\frac{ln(u_i)}{\lambda}$.
- Retourner x_i.

Méthode de transformation- Variables aléatoires discètes

Considérons une variable aléatoire discète non négative X, qui peut prendre les valeurs $x_0, x_1, x_2, ..., x_i, ..., x_n$

- la fonction de masse de X est $f(x) = P(X = x_i) = P_i,\ x_i \geq 0$.
- la fonction cumulative (de répartition) $F(x) = P(X \leq x_i) = \sum_{i=0}^{j} P_i$

La fonction $F^{-1}(u),\ \ u \sim U[0,1]$, est explicitement donnée par :

$$* * *$$

Example

soit une variable aléatoire X telle que

$$P(X = 1) = P_1 = 0.20$$

$$P(X = 2) = P_2 = 0.15$$

$$P(X = 3) = P_3 = 0.25$$

$$P(X = 4) = P_4 = 0.40$$

Example

Algorithme pour générer la distribution de X

générer $U \sim U[0,1]$ et faire ce qui suit :

Si $(U \leq 0.20)$ alors

$X = 1$

Sinon

Si $(U \leq 0.35)$ alors

$X = 2$ Sinon

Si $(U \leq 0.60)$ alors

$X = 3$

Sinon

$X = 4$

FinSi

FinSi

FinSi.

Méthode de convolution

Pour certaines distributions théoriques, la variable aléatoire X peut être expromée comme une somme de deux variables aléatoires quisont identiquement indépendants distribuées (IID).
Nous supposons qu'il existe des variables aléatoires IID Y_1, Y_2,...,Y_m, tel que la somme $Y_1 + Y_2 + ... + Y_m$ a la même distribution que X, nous écrivons do,c $X = Y_1 + Y_2 + ... + Y_m$.
Soit F la fonction de répartition de X et G la fonction de répartition de Y_i. L'algorithme de génération de la variable X est

- Générer Y_1, Y_2,...,Y_m IID chacun avec sa fonction de distribution de G
- Retour $X = Y_1 + Y_2 + ... + Y_m$

Méthode d'accéptation-r ejet

La méthode d'accéptation-rejet utilise une fonction majorante $g(x)$ de la fonction de densité $f(x)$ pour laquelle nous souhaitons générer des varaibles aléatoires.

La fonction majorante $g(x)$ doit avoir les propriétés suivantes

- $g(x) \geq f(x)$
- $m = \int g(x)dx < \infty$
- La variable aléatoire $Y \sim \frac{g(x)}{m}$ est facilement générée.

La méthode d'accéptation-rejet pour générer une variable aléatoire $X \sim f(x)$ peut être résumé comme suit

- Générer Y ayant une fonction de densité $g(x)/m$
- Générer $U \sim U[0, 1]$ indépendant de Y.
- Si $U \geq \frac{f(Y)}{g(Y)}$ alors retournez $X = Y$, sinon revenez à l'étape (1)

Introduction

Quand on parle d'aléatoire en informatique, il ne s'agit presque jamais de véritable hasard. Ce que les ordinateurs produisent, ce sont des nombres pseudo-aléatoires : des nombres qui semblent aléatoires, mais qui sont générés par des algorithmes déterministes.

Definition

Une variable pseudo aléatoire est une variable dont la valeur est générée par un algorithme, de manière à imiter l'aléatoire.

Pourquoi "pseudo" ?

- En informatique, tout est calculé
- Un ordinateur suit des règles strictes : il ne peut pas produire de "vrai hasard" tout seul.
- Les nombres qu'il génère sont donc prévisibles si l'on connaît l'algorithme et les données de départ (la semence ou seed en anglais)

La différence entre les variables aléatoires et pseudo aléatoires

Les variables aléatoires provenant du monde réel (exemple : lancer un vrai dé, mesurer le bruit d'un capteur...) par contre les variables pseudo aléatoire sont calculées dans un programme (exemple : un ordinateur qui génère un nombre au hasard)

Génération des nombres pseudo-aléatoires

- Les méthodes de génération des variables, tronsforment les nombres aléatoires de la distribution uniforme **U[0,1]** pour génerer des théoriques ou des empiriques.
- L'approche moderne consiste à utiliser un ordinateur pour générer successivement des nombres pseudo-aléatoires de distribution Uniforme.
- Les nombres pseudo aléatoires constituent une séquence de valeurs qui, bien qu'elles soient générées de manière déterministe, ont toutes les apparences d'être des variables aléatoires indépendantes uniformes $X \sim U(0,1)$
- La plupart des langages de programmation permettent de générer des valeurs pseudo-aléatoires distribuées selon la loi uniforme
- Différentes méthodes sont utilisées pour générer des nombres pseudo aléatoires : **Linear Congruential Generator ,**

Générateur congruentiel linéaire

- Génèrteur congruentiel linéaire (Linear Congruential Generator) est l'une des méthodes les plus appliquées pour générer les nombres pseudo-aléatoires.
- Le générateur congruentiel linéaire commence par une valeur initiale x_0 appelée semence (seed en anglais) puis calcule de manière récursive les valeurs successives $x_n, n \geq 1$ par la formule

$$x_n = (ax_{n-1} + c) mod(m)$$

  - m, $m > 0$ est le module.
  - a, $0 < a \leq m$ est le multiplicateur.
  - c, $0 < c \leq m$ est l'incrément
  - x_0, $0 < x_0 \leq m$ est la semence ou valeur initiale (seed en anglais)

- Les entiers x_n générés prennent des valeurs de 0 à $m-1$, alor, pour calculer le nombre uniforme pseudo-aléatoire u_n, pris comme une approximation de la valeur d'une variable aléatoire uniforme $U \sim Unif[0,1]$, nous utilisons la formule

$$u_n = \frac{x_n}{m}$$

Example

Considérons un LCG avec des paramètres

$(m = 8,\ a = 5,\ c = 1,\ x_0 = 5)$

Calculez les neuf premières valeurs des x_i et u_i.

Formule du générateur congruetiel linéaire : $x_n = (5x_{n-1} + 1) mod 8$

$x_1 = (5 * 5 + 1) mod 8 = 2;\ \ u_1 = \frac{x_1}{m} = \frac{2}{8} = 0.25$

$x_2 = (5 * 2 + 1) mod 8 = 3;\ \ u_2 = \frac{x_2}{m} = \frac{3}{8} = 0.375$

$x_3 = (5 * 3 + 1) mod 8 = 0;\ \ u_3 = \frac{x_3}{m} = \frac{0}{8} = 0$

$x_4 = (5 * 0 + 1) mod 8 = 1;\ \ u_4 = \frac{x_4}{m} = \frac{1}{8} = 0.125$

$x_5 = (5 * 1 + 1) mod 8 = 6;\ \ u_5 = \frac{x_5}{m} = \frac{6}{8} = 0.75$

$x_6 = (5 * 6 + 1) mod 8 = 7;\ \ u_6 = \frac{x_6}{m} = \frac{7}{8} = 0.875$

$x_7 = (5 * 7 + 1) mod 8 = 4;\ \ u_7 = \frac{x_7}{m} = \frac{4}{8} = 0.5$

$x_8 = (5 * 7 + 1) mod 8 = 5;\ \ u_8 = \frac{x_8}{m} = \frac{5}{8} = 0.625$

$x_9 = (5 * 5 + 1) mod 8 = 2;\ \ u_9 = \frac{x_9}{m} = \frac{2}{8} = 0.25$

Période de LCG

- Soient $x_0, x_1, x_2...$ la séquence générée par un LCG de paramètres (m, a, c, x_0). La période d'un générateur congruentiel linéaire (LCG) est le plus petite entier positive n tel que : $x_0 = x_n$
- La période correspond à la longueur de la séquence avant qu'elle ne commence à se répéter : si $x_0 = x_n$ alors $x_1 = x_{n+1}$, $x_2 = x_{n+2}$, etc
- Une propriété importante d'u LCG est qu'il a une période langue, aussi proche que possible du module m. Si la période du LCG est égale à m, On dit que le LCG a une période complète (full period)

Theorem

Un LCG de paramètres m, a, c, x_0 *a une période complète si et seulement si les trois conditions suivantes sont remplies :*

- m *et* c *doivent être premiers entre eux (Les seuls entiers positifs qui divise à la fois* m, c *et* 1*)*
- $a - 1$ *doit être divisible par tous les facteurs premiers de* m*. Autrement dit, si* q *est un nombre premier qui divise* m*, alors* q *doit diviser* $(a - 1)$
- *Si* 4 *divise* m*, alors* 4 *devrait divise* $(a - 1)$*. Autrement dit, si* m *est multiple de* 4 *alors* $(a - 1)$ *doit également être un multiple de* 4

Example

- Soit le LCG de paramètres $m = 8$, $a = 5$, $c = 1$, $x_0 = 5$. Vérifier s'il doit on non une période complète.
- Pour appliquer le théorème, vous devez vérifier si chacune des trois conditions est vraie pour le générateur.
  Condition 1 : c et m n'ont pas de facteurs communes autre que 1
  Les facteurs de m=8 sont (1,2,4,8), puisque c=1 (de facteur 1), la condition 1 est vérifiée.
  Condition 2 : (a-1) est multiple de tout nombre premier qui divise m
  - Les nombres premiers q, qui divisent m=8, sont q=1,2
  - (a-1)=4, q=1 divise 4 et q=2 divise 4. Alors la condition 2 est vérifiée

Example

Condition 3 : Si 4 divise m, alors 4 de vrait diviser (a-1)

4 divise m=8. De plus, 4 divise (a-1)=4. Ainsi la condition est vérifiée.

Le LCG de paramètres $m = 8$, $a = 5$, $c = 1$, $x_0 = 5$ a une période complète.