

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ MOHAMED KHIDER, BISKRA
FACULTÉ des SCIENCES EXACTES et des SCIENCES de la NATURE et de la VIE
DEPARTEMENT DE MATHEMATIQUE



Polycopié du Cours

Modélisation et Simulation

Première Année Master
Option : Proba & Stat

Préparé par :
Pr. CHERFAOUI Mouloud

Université de Biskra, 2023/2024

Table des matières

Simulation et génération des nombres pseudo-aléatoires

Introduction

Lorsqu'une résolution mathématique exactes d'un problème n'est pas possible, on fait appel aux méthodes d'approximation. Entre autres, figure la méthode de la simulation qui est un outil puissant et idéal dans la majorité des sciences actuelles (Informatique, Physique, Chimie, Télécommunications,...) en plus est un outil de résolution numérique et une discipline de modélisation.

1.1 Simulation

Le terme "Simulation" est dérivé du mot latin *SIMULARE* qui veut dire : copier, feindre, faire paraître comme réelle une chose qui ne l'est pas. Nous conduit aussi vers le chemin qui nous permet de déduire les caractéristiques du fonctionnement d'un système réel. En d'autre terme, la simulation est l'étude du comportement d'un système dynamique à travers un modèle (grâce à un modèle que l'on fait évoluer dans le temps), elle désigne un procédé selon lequel on exécute un programme informatique sur ordinateur en vue de simuler un phénomène ou une organisation particulière : Chaîne de montage en industrie, trafic routier dans une agglomération, bloc opératoire dans un hôpital, etc.

La simulation ne résout pas le problème posé en trouvant la solution, mais elle nous aide à prendre parmi plusieurs solutions la meilleure d'entre elles.

La simulation a pour rôle d'imiter l'évolution d'un système dans le temps et calculer ensuite ses caractéristiques. On a souvent recours à la simulation lorsque :

1. L'expérience sur le système est impossible ou dangereuse (exemple de réacteurs nucléaires).
2. Le système évolue rapidement de telle sorte qu'on ne peut pas l'observer avec précision (exemple de réactions chimiques).
3. Le système évolue très lentement (exemple de simulation du climat à long terme).
4. L'expérience est coûteuse financièrement.
5. Le système est en phase de conception.
6. On veut valider des résultats analytiques.

1.1.1 Terminologies

Définition 1.1 *Un système est un ensemble doté d'une structure, d'un groupe d'éléments et de relations entre ceux-ci dans un environnement fermé. Il est affecté par les éventuels changements de son environnement.*

Dans le domaine de la simulation, plusieurs définitions ont été attribuées au concept "Modèle", en voici celle donnée par l'AFCET (Association Française pour la Cybernétique économique et Technique)

Définition 1.2 *"Un modèle est un schéma, c'est-à-dire une description mentale (intériorisée), ou figurée (diagramme, formules mathématiques, etc.) qui pour un champ de questions est pris comme représentation abstraite d'une classe de phénomènes, plus ou moins habilement dégagés de leur contextes par un observateur pour servir de support à l'investigation, et/ou la communication".*

Le modèle est une représentation simplifiée d'un système réel ou imaginé exprimer sous forme verbale, graphique ou mathématique dans le but de l'étudier. Toutefois, il doit contenir assez de détails pour tirer des conclusions valables sur le système réel.

Définition 1.3 *Les variables d'état sont les informations nécessaires qui ont pour but de définir ce qui est en train de se passer dans un système.*

Définition 1.4 *Les entités sont les éléments ou les objets du système qui subissent des opérations et se déplacent en général dans le système (message dans un réseau,...). Une entité peut être active ou peut se résoudre d'une manière pacifique (passive), permanente ou temporaire.*

Définition 1.5 *Les ressources sont des éléments qui exécutent des opérations, et généralement ne se déplacent pas dans le système (Machine, Unité Centrale,...). Mais on peut trouver des objets qui peuvent exécuter une opération tout en se déplaçant à l'intérieur du système (une machine transportant des palettes dans une usine, chariot transportant une pièce dans un atelier).*

Définition 1.6 *Les attributs sont des variables identification qui caractérisent les entités ou les ressources. On distingue cependant deux types d'attributs :*

1. **Fixes** : Contiennent les caractéristiques constantes de l'objet (durée de service, date d'arrivée dans le système, ...).
2. **Variables** : Contiennent les caractéristiques changeantes de l'objet (état d'une ressource, longueur de la file associée à une ressource,...).

Définition 1.7 *L'activité est un processus qui provoque un changement dans l'état du système, ce changement d'état est appelé événement. Les objets exécutent quelques opérations et dès que ces dernières seront initiées (ou terminées) à chaque événement, elles seront appelées des activités. Les activités utilisées dans la simulation à événement discret possèdent des durées.*

Définition 1.8 *L'événement est la réalisation instantanée qui peut changer l'état du système. On distingue des événements internes au système (endogènes) et des événements externes (exogènes).*

Définition 1.9 *Un processus est le rassemblement d'un certain nombre d'événements dans lequel ces derniers sont produits.*

Définition 1.10 *Un simulateur est un programme contenant l'algorithme utilisé pour simuler le système étudié. Il est constitué d'un ensemble d'entités qui décrivent une composante du système réel.*

1.1.2 Etapes de la simulation

Pour concevoir un simulateur, on doit respecter les étapes suivantes :

- 1) **Formulation du problème** : Cette étape consiste à identifier et analyser le problème tout en déterminant ses composantes, ses relations et les frontières entre le système et son environnement.
- 2) **Fixation des objectifs** : Dans n'importe quel problème à simuler, il faut bien définir les objectifs visés par le projet de simulation. Ceci comprend :
 - les questions auxquelles on devra répondre.
 - Les sorties attendues.
 - Le temps requis ainsi que les coûts de l'étude, etc.
- 3) **Elaboration du modèle** : Il s'agit d'extraire un modèle qui est une abstraction du système réel, c'est-à-dire aussi fidèle que possible de ce dernier.
- 4) **Collecte de données** : Après la formulation d'un problème quelconque et après avoir identifié les objectifs, on fera appel à une collection de données qu'on adapte à des distributions de probabilité tout en faisant appel à des tests statistiques.
- 5) **Validation du modèle** : Il s'agit de traduire le modèle obtenu sous une forme acceptable par ordinateur et de s'assurer que le modèle s'exécute sans erreurs. Le meilleur moyen pour valider un modèle conceptuel est de comparer ses sorties avec celles du système réel. C'est dans cette étape qu'intervient les tests statistiques, c'est-à-dire pour pouvoir valider un modèle de simulation, on fait appel par exemple aux tests d'ajustements (*Khi-Deux ou Kolmogorov-Smirnov*), aux tests de comparaison (*test de Fisher ou test de Student*), ...etc. Le processus de validation d'un modèle peut être illustré par la figure ??.
- 6) **Exécution de la simulation** : L'exécution et le rassemblement des résultats d'un modèle permet de pouvoir le mettre à l'épreuve.
- 7) **Analyse et interprétation des résultats** : Une fois les résultats sont obtenus, le concepteur passe à l'interprétation de ces dernières dans le but de donner des propositions (solutions). Le choix de la meilleure solution devra être fait par l'analyse qui la justifiera dans une documentation qui est nécessaire pour différentes raisons :
 - Aide à comprendre le fonctionnement du modèle.
 - Facilite toute modification ou mise à jour du modèle.
 - Permet de revoir toutes les alternatives prises en considération, des critères de comparaisons qui ont été utilisés. Ceci nous aidera dans la prise de décision qui sera principalement basée sur les résultats fournis par la simulation et rapportés dans la documentation.

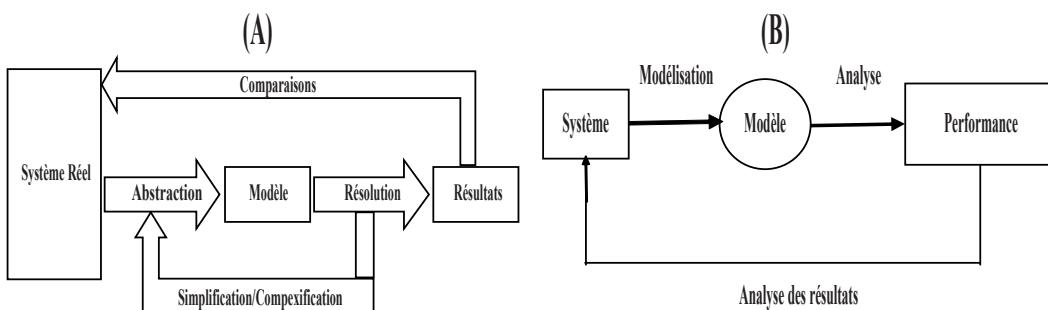


FIGURE 1.1: (A) : Processus de validation et (B) : Processus d'analyse et interprétation des résultats

1.1.3 Simulation à événements discrets

La simulation à événements discrets est la représentation abstraite du système réel tel qu'il évolue dans le temps, c'est-à-dire se rapporte à la modélisation d'un système, dont l'état évolue au cours du temps selon une représentation dans laquelle les variables d'état changent instantanément à certaines dates précises, dans le but d'expliquer certains aspects du comportement de ce système.

1.1.3.1 Principales approches de la simulation à événements Discrets

Il existe quatre approches de la simulation à événements discrets et chaque approche est caractérisée par une méthode différente pour faire améliorer le système dans le temps :

1. **Approche activité (méthode a pas fixe)** : Lorsqu'on utilise cette méthode l'horloge est avancée par intervalle de temps discret et uniforme, et le système est examiné à chaque unité de temps, c'est-à-dire à chaque incrémentation, on vérifie et on cherche s'il y a des événements qui peuvent se produire ou des activités qui peuvent débuter pour pouvoir les traiter.
2. **Approche événements** : Cette méthode est économique en terme de temps de calcul, lorsqu'on l'utilise, on s'intéressera particulièrement à un événement, et l'horloge du temps sera avancé à chaque fois que ce dernier se produit . Ce qui veut dire qu'avant chaque déplacement de l'horloge, il est nécessaire de faire une vérification sur les événements qui nous sont réalisés juste avant, et s'il y a lieu on peut pas exécuter d'autres jusqu'à ce qu'on termine les précédents.
3. **Méthode des trois phases** : le principe de cette méthode consiste à :
 - (a) Déterminer le plus proche événement et de faire avancer l'horloge à ce point.
 - (b) Exécuter toute les activités B (Bound).
 - (c) Exécuter toute les activités C (Conditionnelle), dont les conditions sont satisfaites.
4. **Approche par processus** : La présence des séquences d'événements ou des activités similaires pour un type d'objets, défini sous forme de processus :
 - La description du fonctionnement du système complet par macro-représentation.
 - La gestion des conflits et la synchronisation entre processus par des règles d'interruption et de reprise.

1.1.4 Simulation de Monte Carlo

Cette méthode a été développée durant les années 40 par Von Neumann, Vlan et Fermi pour résoudre certains problèmes de désigne d'écrans anti-rayonnement. Elle est conçue à l'origine pour résoudre des problèmes déterministes et consiste à représenter un problème déterministe par un processus stochastique dont les distributions de probabilité satisfont les relations mathématiques du problème déterministe complexe [?].

La première étape consiste à construire un modèle analytique qui représente l'investissement considéré (*exemple* : une équation pour la valeur présente). Dans la deuxième étape, l'analyste développe une distribution de probabilité pour chaque facteur qui est sujet à l'incertitude dans le modèle. Il peut utiliser des données historiques ou une approche subjective. À partir des distributions de probabilités pour chaque facteur dans le modèle, il génère au hasard une réponse tentative. La répétition de ce mode d'échantillonnage varie entre les différentes histoires bien que le modèle reste le même. Une question importante à connaître dans la simulation de Monté Carlo est le nombre d'essais (ou d'exécution) nécessaire pour avoir une réponse satisfaisante. La réponse est que le nombre d'essais doit être assez élevé pour atteindre le régime permanent.

On peut définir les conditions de régime permanent comme une situation dans laquelle le résultat d'essais consécutifs ne varie pas significativement. Les modèles analytiques et la simulation de Monté Carlo ont les mêmes problèmes, à savoir, la validité du modèle. Cependant, dans le cas de la simulation de Monté Carlo, la taille de l'échantillon doit être assez grande pour faire décroître la validité d'échantillonnage à un niveau acceptable. Les probabilités des événements doivent être basées sur les jugements de personnes impliquées dans le projet, utilisées avec le jugement du décideur.

1.2 Génération des nombres pseudo-aléatoires

Toute simulation des phénomènes stochastique se fait par un moyen indirect, qui est la génération de nombres aléatoires, cette dernière consiste à obtenir des observations des variables aléatoires à partir de la distribution désirée, pour que ces observations puissent interpréter fidèlement la réalité du phénomène, il est nécessaire qu'elles aient la même loi de probabilité. L'objectif de la génération des nombres aléatoires est la construction d'une suite de nombres statistiquement indépendantes et réparties uniformément sur $[0, 1]$:

$$J_{n+1} = f(J_n), \quad (1.1)$$

où f est une fonction qui doit être choisie minutieusement et précisément, pour que la répartition des nombres J_n se rapproche de ce que donnerait le hasard. C'est à partir de là que la notion de nombres **pseudo-aléatoires** est apparue qui n'est autre qu'une suite de nombres aléatoires générée par un algorithme déterministe.

1.2.1 Méthodes de génération des nombres pseudo-aléatoires

1.2.1.1 Méthode congruentielle générale

Cette méthode, introduite par Lehmer en 1951, est une génération d'une suite de nombres par une formule tout à fait déterministe de manière à obtenir une suite qui semble aléatoire (indépendante et distribution uniforme dans l'intervalle de variation).

Formule du générateur congruentiel

Soit X_0 un nombre entier positif. La suite de nombres est définie par :

$$x_{n+1} = (ax_n + c) \bmod(m). \quad (1.2)$$

Avec :

- m : module, $m > 0$.
- a : multiplicateur, $0 \leq a < m$
- c : l'incrément, $0 \leq c < m$.

Dans le cas où :

1. $c=0$: le générateur congruentiel est dit linéaire.
2. $c \neq 0$: le générateur congruentiel est dit mixte.

Propriétés et vocabulaire de la méthode congruentiel

- $X_i < m$.
- Le nombre maximum de valeurs possibles de la suite est m .
- Dès qu'un nombre est répété, toute la séquence recommence.

- p dénote la période et le nombre de valeurs possibles de la suite.
- Si $p = m$, le générateur est dit de période maximum.

Théorème 1.1 *Un générateur est de période maximum $p = m$ si et seulement si les trois conditions suivantes sont vérifiées :*

- c et m n'ont pas de diviseur commun,
- $a = 1$ modulo g pour tous les nombres premiers g diviseurs de m (Pour rappel, $a = i$ modulo j si a peut s'exprimer comme $a = i + kj$ où k est un entier positif ou nul).

Remarque 1.1 Pour obtenir des nombres pseudo aléatoires uniformément répartis sur $[0,1]$, il suffit de prendre :

$$U_n = \frac{x_n}{m}. \quad (1.3)$$

Il s'agit de procréer (engendrer) une variable aléatoire X de fonction de densité $f(x)$, et de fonction de répartition $\mathbb{F}(x)$, suivant une loi uniforme sur $[0, 1]$ en se basant sur des techniques connues dont certaines s'appliquent à la génération de variables pseudo-aléatoires de distributions quelconques, tant dit que d'autres ne s'appliquent qu'aux distributions continues ou discrètes. Parmi les principales techniques de génération de nombres pseudo-aléatoires, on distingue :

1. **Méthode d'inversion** Si la fonction de répartition est explicitement connue, et la fonction de répartition inverse \mathbb{F}^{-1} existe. Dans ce cas, la méthode de la transformation inverse peut être utilisée comme suit :
 - Générer des variables $Y \rightsquigarrow$ Uniforme $[0, 1]$;
 - Écrire Y en fonction de $\mathbb{F}(X)$;
 - Déduire : $x = \mathbb{F}^{-1}(y)$.

Remarque 1.2 La technique de transformation inverse est utilisée pour des distributions dont $\mathbb{F}(x)$ est simple.

Implémentation de la méthode d'inversion :

- i) Cas discret : Générer n observations d'une variable aléatoire X qui suit une loi discrète.

x_i	x_0	x_1	\dots	x_N
p_i	p_0	p_1	\dots	p_N
$\mathbb{F}(x)$	p_0	$p_0 + p_1$	\dots	1

Algorithmme

Début

Pour i allant de 1 à N faire :

Générer $Y \rightsquigarrow U[0, 1]$;

Trouver n tel que $\sum_{j=0}^n p_j \leq y < \sum_{j=0}^{n+1} p_j$

$x_i = x_n$;

Fin Pour ;

Fin ;

ii) Cas continue :

Génération d'une loi uniforme sur $[a, b]$:

La fonction de densité d'une variable aléatoire suivant une loi uniforme sur $[a, b]$ est donnée par :

$$f(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b; \\ 0, & \text{ailleurs.} \end{cases} \quad (1.4)$$

La fonction de répartition est donnée par :

$$\mathbb{F}(x) = \begin{cases} 0, & x < a; \\ \frac{x-a}{b-a}, & a \leq x \leq b; \\ 1, & x > b. \end{cases} \quad (1.5)$$

Calcul d'inverse : On a,

$$\mathbb{F}(x) = u, \text{ avec } u \in [0, 1] \Rightarrow \frac{x-a}{b-a} = u \Rightarrow x = u(b-a) + a.$$

Donc, pour simuler une variable aléatoire uniforme sur $[a, b]$, il suffit de générer des nombres aléatoires u_i de variable aléatoire uniforme sur l'intervalle $[0, 1]$ et déduire les réalisations x_i , tel que : $x_i = a + (b - a)u_i$.

2. **Méthode de rejet** : Le but est de simuler une variable aléatoire X de densité $f(x)$ et de fonction de répartition $\mathbb{F}(x)$. La méthode du rejet évoque deux hypothèses sur la fonction $f(x)$:

- f est définie sur un support compact, c'est-à-dire qu'elle prend la valeur nulle en dehors d'un certain intervalle fermé $[a, b]$. Cette méthode se résume comme suit :
 - Normaliser la fonction $f(x)$ de telle façon que : $f(x) = c \times g(x)$, où c est une constante et $g(x) \leq 1$.
 - Générer une variable aléatoire U_1 qui suit une loi uniforme sur $[0, 1]$.
 - Définir x comme fonction linéaire de u_1 , $x = a + (b - a) \times u_1$.
 - Générer une autre variable aléatoire U_2 qui suit une loi uniforme sur $[0, 1]$. Si $U_2 \leq g(x)$, alors on accepte x , comme une réalisation de X . Sinon, on la rejette et on revient à l'étape b.
- Supposons qu'on dispose d'une fonction $g(x)$ majorant de $f(x)$, C'est-à-dire : $f(x) < g(x)$, $\forall x$. Si $g(x)$ n'est pas une fonction de densité (c'est le cas en général), alors on prend :

$$c = \int_0^{\infty} g(t)dt, \quad \int_0^{\infty} f(t)dt = 1 \quad \text{et} \quad h(x) = \frac{g(x)}{c}.$$

Dans ce cas, les étapes précédentes b), c) et d) deviennent sous la forme suivante :

- Générer y selon h .
- Générer $U \sim [0, 1]$.
- Si $U \leq \frac{f(y)}{g(y)}$, alors on prend $x = y$;
sinon, on rejette et on revient à l'étape a).

3. **Méthode de convolution** : La distribution de la somme de plusieurs variables aléatoires indépendantes est appelée : convolution des distributions initiales. Cette méthode consiste à sommer deux variables (ou plus) pour obtenir une variable aléatoire distribuée selon la loi de probabilité souhaitée (voir l'exemple ??). Si la variable aléatoire qu'on désire générer peut se représenter comme somme d'autres variables que l'on peut générer aisément :

$X = Y_1 + Y_2 + Y_3 + \dots + Y_n$, alors on génère les Y_i et on obtient X en les sommant.

Exemple 1

- Une loi Gamma est une somme des lois Exponentielles de même paramètre.
- Une loi Khi-Deux (χ_n^2) à n degré de liberté est une somme de n lois Normales $[0, 1]$ indépendantes au carré.
- Une loi Binomiale(n, p) est une somme de n lois Bernoulli(p).

Génération d'une variable qui suit une loi normale On dit que X suit une loi normale $N(m, \sigma)$, lorsque la densité de sa loi est donnée par :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-m}{\sigma})^2}.$$

La fonction de répartition de la loi Normale ne peut pas être explicitée, donc la méthode d'inversion ne s'applique pas. On la génère de la manière suivante :

$$X = \sigma Y + m,$$

où Y est une variable aléatoire Normale centrée réduite, calculée à partir d'une suite de n variables aléatoires $U_1, U_2, U_3, \dots, U_n$ uniformément distribuées entre $[0, 1]$.

$$Y = \frac{\sum_{i=1}^n U_i - \frac{n}{2}}{\sqrt{\frac{n}{12}}}. \quad (1.6)$$

Remarque 1.3 Cette écriture est justifiée par le théorème des grands nombres : $\sum_1^n X_i \rightsquigarrow N(n\bar{X}, n\sigma_X)$, \forall la loi de X .

4. Méthode de décomposition

Il s'agit de décomposer la fonction de répartition \mathbb{F} de la variable aléatoire X comme suit :

$$\mathbb{F}(x) = \sum_{i \geq 1} p_i \mathbb{F}_i(x). \quad (1.7)$$

La méthode se résume ainsi :

Écrire $\mathbb{F}(x) = \sum_{i \geq 1} p_i \mathbb{F}_i(x)$ où : $\sum_{i \geq 1} p_i = 1$.

- Générer $U \rightsquigarrow U[0, 1]$.
- Trouver i tel que : $0 + p_1 + p_2 + p_3 + \dots + p_{i-1} < U \leq p_1 + p_2 + p_3 + \dots + p_i$.
- Générer X selon $\mathbb{F}_i(x)$.

Pour s'assurer que les suites des nombres générées possèdent un comportement aléatoire selon la loi désirée, il faut tester ce comportement selon deux critères : l'Uniformité et l'Indépendance, tout en utilisant les tests d'Ajustement.

1.2.2 Quelques tests statistiques

Les tests d'ajustement ont pour but de vérifier si un échantillon provient ou pas d'une variable aléatoire de fonction de distribution connue $\mathbb{F}_0(x)$. Soit $\mathbb{F}(x)$, la fonction de répartition de la variable échantillonnée. Il s'agit de tester

H_0 " $\mathbb{F}(x) = \mathbb{F}_0(x)$ " contre H_1 " $\mathbb{F}(x) \neq \mathbb{F}_0(x)$ " [1].

Les tests classiques les plus usuels sont :

1. Test de Khi-Deux

Soit X_1, X_2, \dots, X_n un n-échantillon issu d'une variable aléatoire X . On partage le domaine D de la variable X , partie de l'ensemble des réels R , en r classes c_1, c_2, \dots, c_r . Généralement, on prend $r \simeq \sqrt{n}$.

Soit :

- n_i : l'effectif de la classe c_i .
- p_i : la probabilité de se trouver dans la classe c_i . Elle est déduite à partir de la loi de probabilité à tester.
- $n_i p_i$: effectif théorique de la classe c_i .

Pearson a démontré que la variable aléatoire

$$K_n^2 = \sum_{i=1}^r \frac{(n_i - np_i)^2}{np_i}, \quad (1.8)$$

suit asymptotiquement une loi de Khi-Deux à $(r - 1)$ degré de liberté. n_i étant une variable aléatoire représentant l'effectif de la classe c_i et dont la réalisation est n_i . Soit k_n^2 la réalisation de la variable aléatoire K_n^2 .

La règle de décision est alors sous la forme :

- Si $k_n^2 < \chi_{(r-1,\alpha)}^2$, on accepte l'ajustement de la variable aléatoire X par la loi choisie.
- Si $k_n^2 \geq \chi_{(r-1,\alpha)}^2$, on rejette l'ajustement de la variable aléatoire X par la loi choisie.

Lorsque les paramètres de la loi à valider sont estimés à partir de l'échantillon, le degré de liberté du Khi-Deux est alors égal à $(r - q - 1)$, q étant le nombre de paramètres estimés. L'application du test Khi-Deux doit satisfaire les conditions suivantes :

- (a) Le nombre de classes doit être supérieur ou égal à 7.
- (b) L'effectif théorique de chaque classe doit être supérieur ou égal à 8.
- (c) Les effectifs théoriques des k classes doivent être sensiblement égaux.

2. Test de Kolmogorov-Smirnov

Ce test est plus puissant que le précédent car, c'est celui dans lequel le risque d'accepter H_0 à tort est plus faible. La procédure à suivre est la suivante :

- On tire un échantillon de n observations à l'aide du générateur ;
- On classe les observations selon un ordre croissant ;
- On compare la fonction de répartition empirique $\mathbb{F}_n(x)$ calculée à partir de ces n nombres pseudo-aléatoires avec la fonction de répartition théorique $\mathbb{F}(x)$ (loi uniforme sur $[0, 1]$) et on calcul :

$$D_n = \max | \mathbb{F}_n(x) - \mathbb{F}(x) | = \max D(x_u), \quad (1.9)$$

où

$$\mathbb{F}_n(x) = \frac{\{\text{nombre d'observations} \leq x\}}{\text{la taille de l'échantillon}}$$

et

$$\mathbb{F}(x) = x, \quad \text{si } x \in [0, 1].$$

On fixe un seuil de signification α , et soit $d(\alpha)$ la valeur tabulée (obtenue à partir de la table de Kolmogorov-Smirnov), tel que $P(D_n > d(\alpha)) = \alpha$. La règle de décision est alors de la forme :

- On rejette H_0 Si $D_n > d(\alpha)$;
- On accepte H_0 Sinon.

3. Test de Student Un des plus importants usages de la simulation est la comparaison des performances de deux ou plusieurs systèmes. Pour réaliser cette comparaison à partir d'échantillons (données statistiques) obtenus de la simulation des systèmes observés, nous devons recourir aux méthodes statistiques. Dans ce qui suit, nous présenterons l'une de ces méthodes à savoir le test de Student.

Etant donné deux échantillons de taille n_1, n_2 , peut-on admettre qu'ils ont été prélevés dans une même population relativement à la variable étudiée.

Mathématiquement, le problème se formalise de la manière suivante :

On observe sur le premier échantillon (les réalisations d'une variable aléatoire X_1), de fonction de répartition $F_1(x)$ et sur le deuxième échantillon (les réalisations d'une variable aléatoire X_2), de fonction de répartition $F_2(x)$; on veut tester :

$$H_0 : F_1(x) = F_2(x) \quad \text{contre} \quad H_1 : F_1(x) \neq F_2(x).$$

Dans la pratique, on se contentera de vérifier l'égalité des espérances, donc on a à tester :

$$H_0 : " \bar{X} = m_{th} " \quad \text{contre} \quad H_1 : " \bar{X} \neq m_{th} ",$$

où

\bar{X} : est la moyenne de l'échantillon / m_{th} : est la moyenne théorique.

Ce test est basé sur la statistique suivante :

$$T_{n-1} = \frac{(\bar{X} - m_{th})\sqrt{n-1}}{S}, \quad (1.10)$$

avec :

$$S^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2.$$

H_0 est acceptée si :

$$T_{n-1} < t_{(n-1, \frac{\alpha}{2})}, \quad (1.11)$$

où $t_{(n-1, \frac{\alpha}{2})}$ est la quantité de Student au seuil α .

Conclusion

La simulation à événements discret est un outil puissant et universel. Les gains à tirer d'une expérience de simulation sont variés : description (validation d'une architecture), explication (observation, expérimentation sur une maquette), et prédition (mesure de performances, ou prévision de comportement). Son champ d'application recouvre les problèmes d'analyse, de conception, d'optimisation.

La simulation nous permet d'étudier les conditions d'opération extrême du système et d'évaluer les conséquences sans mettre en danger le système ou son environnement. Permet de concevoir des systèmes très complexes, elle devient le moyen le plus sûr pour la compréhension d'un système quand son étude s'avère difficile. C'est un instrument idéal de "décideur", aussi bien que de technicien.

Malgré tout ce qu'elle présente comme avantages la simulation est un outil coûteux en termes de temps de calcul, consommateur de ressources informatiques, et ne fournit que des estimations de ce que l'on cherche.

Chapitre 2

Simulation de lois usuelles et la fonction *RANDOM* sous Matlab

2.1 Simulation de lois usuelles

Dans cette section nous allons présenter quelques informations sur les loi utiliser dans le chapitre trois ainsi que les algorithme nous permetont de générer un échantillon de taille n à partir de ces lois.

2.1.1 Loi Exponentielle

La densité de probabilité d'une loi exponentielle de paramètre $\lambda > 0$ est donnée par :

$$f(x) = \lambda e^{-\lambda x}, \quad \forall x \geq 0,$$

sa fonction de répartition est définie par :

$$F(x) = 1 - e^{-\lambda x}, \quad \forall x \geq 0.$$

Pour cette distribution on a :

$$\mathbb{E}(X) = \frac{1}{\lambda} \quad \text{et} \quad \sigma^2(X) = \frac{1}{\lambda^2}.$$

Propriété "sans mémoire" : Une propriété importante de la loi exponentielle est son absence de mémoire c'est-à-dire,

$$P(X < t + t_0 | X > t_0) = P(X < t), \forall t_0, t > 0.$$

Pour générer une observation d'une loi exponnetielle de paramètre λ on exploite l'algorithme suivant :

Algorithme :

Début

Générer une *v.a* U uniforme sur $[0,1]$;

$X \leftarrow -\frac{1}{\lambda} \ln(U)$;

Retourner X ;

Fin ;

2.1.2 Loi d'Erlang

Une variable aléatoire X suit une loi de Erlang d'ordre k ($k = 1, 2, \dots$) de paramètre λ , si elle est la somme de k variables aléatoires indépendantes X_1, X_2, \dots, X_k et identiquement distribuées suivant une loi exponentielle de paramètre λ . La densité de probabilité d'une loi Erlang d'ordre k est donnée par :

$$f(t) = \lambda \frac{(\lambda t)^{(k-1)}}{(k-1)!} e^{-\lambda t}, t \geq 0$$

La fonction de répartition est ainsi donnée comme suit :

$$\mathbb{F}(t) = 1 - \int_t^{\infty} f(x) dx$$

Les paramètres λ et k sont appelés respectivement paramètre d'échelle et de forme. Pour cette distribution, on a :

$$\mathbb{E}(X) = \frac{k}{\lambda} \quad \text{et} \quad \sigma^2(X) = \frac{k}{\lambda^2}.$$

La loi d'Erlang est un cas particulier de la loi de $Beta(a, b)$ lorsque $a \in \mathbb{N}$.

Pour générer une variable aléatoire selon une loi d'Erlang d'ordre k de paramètre λ , on peut utiliser les algorithmes de génération des nombres aléatoires d'une loi Gamma. Mais sa définition nous permet de construire un simple algorithme de simulation étant donnée une somme k de variables aléatoires d'une loi exponentielle de paramètre λ .

L'algorithme

Début

Générer k v.a uniforme sur $[0, 1]$ (U_1, U_2, \dots, U_n) ;

$$X \leftarrow \sum_{i=1}^n \left(-\frac{1}{\lambda} \right) \log(U_i) = -\frac{1}{\lambda} \log\left(\prod_{i=1}^n U_i\right);$$

Retourner X ;

fin ;

2.1.3 Loi de Weibull

On dit que la variable aléatoire X suit la loi de Weibull, de paramètres (α, β, ν) , et on note $X \rightsquigarrow wbl(\alpha, \beta, \nu)$, si sa densité s'écrit sous la forme :

$$f(X) = \begin{cases} \frac{\beta}{\alpha} \left(\frac{x-\nu}{\alpha} \right)^{\beta-1} e^{-\left(\frac{x-\nu}{\alpha} \right)^\beta} & \text{si } x \geq \nu \\ 0 & \text{si } x < \nu \end{cases}. \quad (2.1)$$

où : $\alpha > 0$ est un paramètre d'échelle, $\beta > 0$ est un paramètre de forme et $\nu \in \mathbb{R}$ est un paramètre de position. Son espérance et sa variance sont respectivement :

$$E(X) = \alpha \Gamma\left(1 + \frac{1}{\beta}\right) + \nu \quad \text{et} \quad Var(X) = \alpha^2 \left[\Gamma\left(1 + \frac{2}{\beta}\right) - \Gamma^2\left(1 + \frac{1}{\beta}\right) \right].$$

avec $\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t}$, $x \in \mathbb{R}$ (la fonction gamma).

2.2 La fonction *RANDOM* sous Matlab

Tous les logiciels de calcul sont munis de fonctions prédefinies pour la génération de nombre pseudo-aléatoires dont la syntaxe et les lois qui peuvent être simulées diffèrent d'un logiciel à un autre. Sous Matlab on trouve qu'il existe plusieurs fonctions de génération de nombres aléatoires, entre autres on cite : *rand*, *randn* et *random*. La fonction *rand* renvoie des nombres à virgule flottante entre 0 et 1 tiré d'une distribution uniforme, la fonction *randn* est spécifique pour la génération des nombres aléatoires tirés d'une distribution normale centrée et réduite, tandis que la fonction *random* est plus générale où elle peut être utilisée pour générer des variables aléatoires issue d'une loi usuelle fixée préalablement par l'utilisateur. Pour cette dernière, l'utilisateur peut utiliser la commande :

random(nom,A) qui renvoie un nombre aléatoire de la famille de distribution à un paramètre spécifiée par "nom" et le paramètre de la distribution *A*. **Exemple** : *random('exp',2)* rend une observation d'une variable aléatoire de loi exponentielle de paramètre 2.

random(nom,A,B) qui renvoie un nombre aléatoire de la famille de distribution ayant deux paramètres, spécifiée par "nom" et les paramètres de la distribution *A* et *B*. **Exemple** : *random('norm',20,2)* rend une observation d'une variable aléatoire de loi normal de moyenne 20 et d'écart-type 2.

random(nom,A,B,C) qui renvoie un nombre aléatoire de la famille de distribution à trois paramètres spécifiée par nom et les paramètres de cette distribution *A*, *B* et *C*. **Exemple** : *random('Weibull',2,3,4)* rend une observation d'une variable aléatoire de loi de Weibull de paramètre d'échelle 2, de paramètre de forme 3 et de paramètre de position 4.

random(...,n₁,n₂,...,n_p) ou random(...,[n₁,n₂,...,n_p]) renvoie une matrice de dimension $n_1 \times n_2 \times \dots \times n_p$ de nombres aléatoires.

Exemples :

1. *random('exp',2,5,3)* : rend une matrice de dimension 5×3 d'observations issues d'une v.a de loi exponentielle de paramètre 2.
2. *random('exp',2,5)* : rend une matrice de dimension 5×5 d'observations issues d'une v.a de loi exponentielle de paramètre 2.
3. *random('exp',2,5,1)* : rend un vecteur colonne de dimension 5 d'observations issues d'une v.a de loi exponentielle de paramètre 2.

Pour plus de détails sur la fonction prédefinie *RANDOM* et les noms et les détails des différentes lois qui peuvent être simulées sous Matlab, le lecteur peut se référer au **HELP** du logiciel Matlab.

Chapitre 3

Applications : Modélisation et simulation

Introduction

L'objectif du présent passage est, à travers des exemples de différents champs, d'expliquer à l'étudiant les différentes étapes à suivre, de la modélisation à l'interprétation des résultats, pour résoudre un problème réel ou théorique à l'aide de la technique de simulation.

3.1 Méthode de rejet pour le calcul d'intégrales

Implémenter sous Matlab des fonctions qui nous permettent de :

1. calculer $I = \int_a^b e^{-x^2} dx$ avec $a < b$.
2. calculer $I = \int_a^b \ln(\sin(x) + \cos(x))dx$ avec $0 \leq a < b \leq \frac{\pi}{2}$.
3. estimer la valeur de π .

3.2 Fiabilité d'un système électrique

Soit un système électrique bien déterminée, composée de trois types de composants. Supposons que nous nous intéressons à l'analyse statistique de la durée de vie du système.

Pour cela, soit X la variable aléatoire représentant la durée de vie du système global et X_1, X_2 et X_3 des variables aléatoires représentant respectivement la durée de vie de chacun des composants. Les informations dont dispose sur les trois composants sont comme suit :

	composant 1	composant 2	composant 3
Nombre de composants (Nbr)	2	5	3
Loi de X_i	$N(\mu = 10, \sigma^2 = 3^2)$	$Exp(\lambda = 15)$	$wbl(\alpha = 10, \beta = 3)$

Question :

1. Donner la forme générale de la densité de X (sans développement) qu'on note f .
2. Implémenter sous MATLAB une fonction qui nous permette de générer un échantillon de taille n dont les paramètres d'entrer sont : $Nbr, \mu, \sigma, \lambda, \alpha, \beta$ et la taille de l'échantillon n .
3. Générer un échantillon de taille n (utiliser différentes taille n) et visualiser graphiquement l'estimateur à noyau de la densité f (utiliser la fonction **ksdensity**).

4. En exploitant le programme de simulation implémenter, donner une estimation ponctuelle et par intervalle de confiance de la durée moyenne de vie du stsrème et sa variance.
5. Vérifier la normalité asymptotique de l'estimateur de la moyenne de X .

3.3 Processus Markovien : Marche aléatoire

Soit une particule qui se déplace aléatoirement sur une ligne d'une distance d'un centimètre (en avant +1 ou en arrière -1) chaque 2 secondes.

1. Donner un modèle mathématique (processus aléatoire) décrivant le phénomène ainsi que les probabilités de transition d'un état à un autre.
2. Donner un algorithme qui nous permet de simuler ce phénomène.
3. Si la particule est initialement au point x_0 et le déplacement (+1 ou -1) se fait d'une manière équiprobable, alors :
 - (a) Donner une estimation de la probabilité que la particule soit à l'état i , après 1 minute (et 10 minutes) de mouvement pour différentes point initiales $x_0 \in \{0, 1, 2, 3, -1, -2, -3\}$.
 - (b) Donner une estimation de la durée moyenne du premier retour de la particule à l'origine $x_0 \in \{0, 1, 2, 3, -1, -2, -3\}$.
 - (c) la position initiale de la particule a-t-elle une influence sur les probabilités de position de la particule et de la durée moyenne de la première visite de l'état initial ?
4. Revoir la question 3 dans le cas :
 - (a) $P(\text{déplacer de } +1)=0.6$ et $P(\text{déplacer de } -1)=0.4$
 - (b) $P(\text{déplacer de } +1)=P(\text{déplacer de } -1)=0.4$, et $P(\text{déplacer de } 0)=0.2$.

3.4 Modèle de ruine

Considérons une compagnie d'assurance dont le fonctionnement de sa liquidité peut être décrit par le modèle de risque suivant :

$$X(t) = u + ct - \sum_{i=1}^{N(t)} Z_i, \quad t \geq 0, \tag{3.1}$$

où : Le processus de comptage $\{N(t), t \geq 0\}$ est le processus du nombre de réclamations des sinistrés et Z_i est le montant du $i^{\text{ème}}$ sinistré avant la date t par conséquence la variable $Z(t) = \sum_{i=1}^{N(t)} Z_i$ représente le montant cumulé des réclamations à l'instant t . La constante u représente la réserve initiale. La prime est proportionnelle au temps (ct) où $c > 0$ est le taux de prime constant.

Supposons que le modèle (??) est construit selon les hypothèses suivantes :

- Le processus de comptage $\{N(t), t \geq 0\}$ est un processus de Poisson d'intensité λ .
- Les montants des réclamations est une suite de variables aléatoires indépendantes et identiquement distribuées selon une loi de Weibull de paramètres α et β .
- La réserve initiale $u = 1000$ unités monétaires et la prime $c = 20$ unités monétaires.
- Afin d'éviter une ruine certaine, nous supposons que le chargement de sécurité relative $\theta = \frac{c-\lambda}{\lambda m} > 0$ avec m le montant moyen des réclamations donné par : $m = \alpha \Gamma(1 + \beta^{-1})$ (moyenne de la loi de Weibull).

Travail demandé : A l'aide de la simulation, donner une estimation de la probabilité de ruine de cette compagnie à l'horizon $T \in \{100, 500, 1000, 5000\}$ toute en utilisant

1. Approche activité.
2. Approche événement.
3. Comparer vos résultats.

3.5 Corrigé des exemples

3.5.1 Exemples d'intégration

Pour répondre à de tels exemples (calcul d'intégrales ou de surfaces) il est clair nous faisons recours à la méthode de rejet-acceptation. Mais pour implémenter cette technique nous devons d'abord :

1. Vérifier si la fonction f change de signe sur l'intervalle d'intégration $[a, b]$
2. Déterminer le max et le min de f sur l'intervalle $[a, b]$.

Pour les deux premiers exemples une présentation graphique des deux fonctions considérées dans l'intégration sont présenté dans la figure ??.

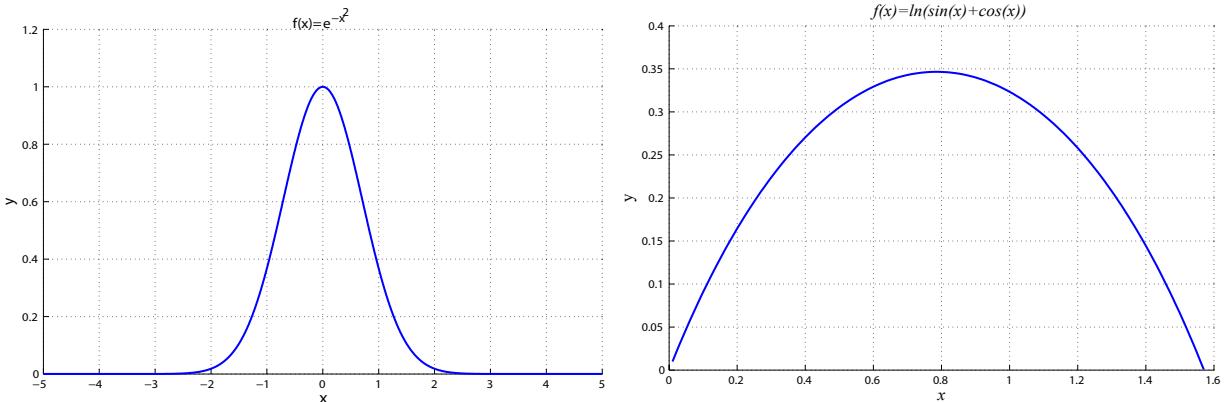


FIGURE 3.1: Allure graphique des fonctions à intégrées.

D'après la figure ??, on constate que :

Cas $f(x) = e^{-x^2} dx$

1. $\forall x \in I = [a, b] \subseteq \mathbb{R}, f(x) \geq 0$.
2. $\forall x \in I = [a, b] \subseteq \mathbb{R}, f(x) \leq 1$.

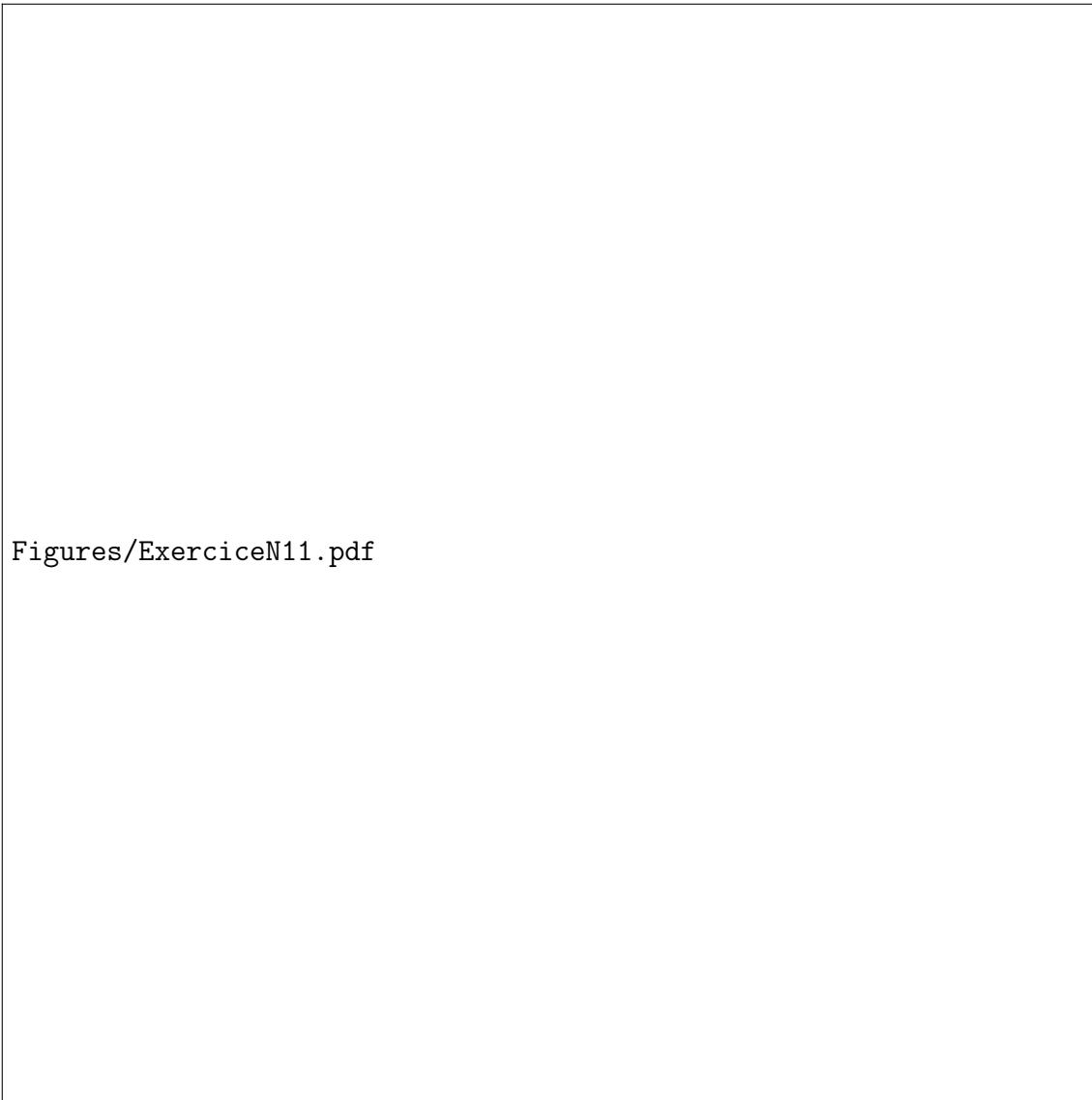
Par conséquence, la surface (l'intégrale) recherché peut être délimitée par un rectangle dont les sommets sont : $(a, 0)$, $(a, 1)$, $(b, 0)$ et $(b, 1)$ dont la surface $S = (b - a)$. Alors, pour générer des point aléatoire uniformément distribuées sur ce dernier rectangle, il suffit des générer leurs abscisses selon une loi uniforme entre a et b et leurs ordonnée selon une loi uniforme entre 0 et 1.

Cas $f(x) = \ln(\sin(x) + \cos(x)) dx$

1. $\forall x \in I = [a, b] \subseteq \mathbb{R}, f(x) \geq 0$.
2. $\forall x \in I = [a, b] \subseteq \mathbb{R}, f(x) \leq 0.35$.

Par conséquence, la surface (l'intégrale) recherché peut être délimitée par un rectangle dont les sommets sont : $(a, 0)$, $(a, 0.35)$, $(b, 0)$ et $(b, 0.35)$ dont la surface $S = (b - a) * 0.35$. Alors, pour générer des point aléatoire uniformément distribuées sur ce dernier rectangle, il suffit de générer leurs abscisses selon une loi uniforme entre a et b et leurs ordonnée selon une loi uniforme entre 0 et 0.35.

Ainsi, en tenant compte des différents aspects cités ci-haut les programmes Matlab qui nous permettra d'estimer la valeur des intégrales en question toute en utilisant l'approche Monté Carlo seront comme suit (voir figure ??) :



Figures/ExerciceN11.pdf

FIGURE 3.2: Coude source Matlab pour l'estimation des deux exemples d'intégration.

Pour l'estimation de π via la simulation, on sait d'une part que la surface s d'un disque de délimiter par un cercle de centre $(0,0)$ et de rayon $R=1$ dont l'équation est $x^2 + y^2 < 1$ est donnée par $s = \pi R^2 = \pi$. D'autre part, le disque en question est situé à l'intérieur d'un carré de sommets $(-1,0)$, $(-1,1)$, $(1,0)$ et $(1,1)$ dont la surface $S=4$.

Ainsi, si on généré n point uniformément sur le carré décrit ci-haut alors l'équation suivante aura lieu :

$$\frac{s}{S} = \frac{Nbr}{n}, \quad \text{avec } Nbr \text{ est le nombre de points se trouvant sur le disque.}$$

Par conséquence,

$$\frac{s}{S} = \frac{Nbr}{n} \Rightarrow s = S * \frac{Nbr}{n} \Rightarrow \pi = 4 * \frac{Nbr}{n}.$$

On se base sur ces données, on peut proposer le programme suivant pour estimer la valeur de π .

```
function [lhat]=Exemple1(a,b,n,mc)
S=(b-a)*1; % Surface du rectangle
for j=1:mc
    Nbr=0;
    for i=1:n
        x=random('unif',a,b);
        y=random('unif',0,1);
        if (y<exp(-x^2))
            Nbr=Nbr+1;
        end
    end
    l(j)=(Nbr/n)*S;
end
lhat=mean(l);
end
```

```
function [lhat]=Exemple2(a,b,n,mc)
S=(b-a)*0.35; % Surface du rectangle
for j=1:mc
    Nbr=0;
    for i=1:n
        x=random('unif',a,b);
        y=random('unif',0,0.35);
        if (y<(log(sin(x)+cos(x))))
            Nbr=Nbr+1;
        end
    end
    l(j)=(Nbr/n)*S;
end
lhat=mean(l);
end
```

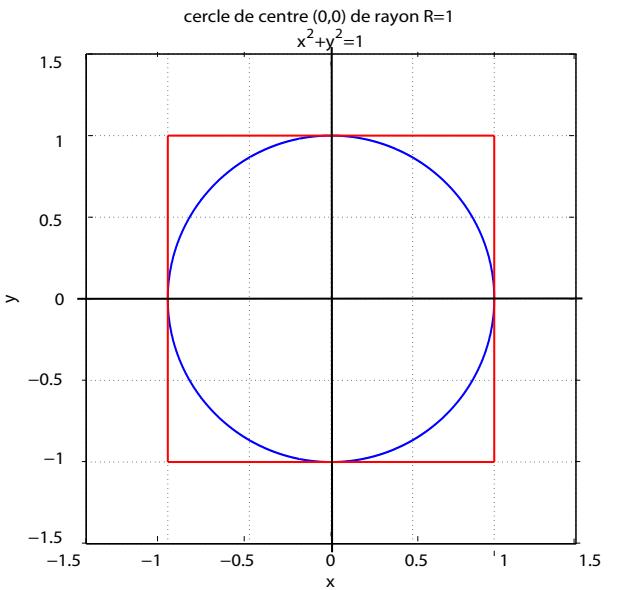


FIGURE 3.3: Allure graphique des fonctions à intégrées.

Bibliographie

- [1] Babes, M. (1995) Statistiques, Files d'attente et Simulation.
- [2] Banks, J. (1998) Handbook of simulation : Principles, methodology, advances, applications and practice. Wiley.
- [3] Erard, P.J. et Déguénon, P. (1996) Simulation par événements discrets. Presses Polytechniques et Universitaires Romande.
- [4] Perros, H. (2003) Computer Simulation Techniques : The Definitive Introduction. Computer Science Department NC State University Raleigh, NC.
- [5] Sheldon, M.R. (2001) Simulation, Academic Press.
- [6] Sinclair, J.B. (2004) Simulation of computer systems and computer networks : A Process-Oriented Approach.
- [7] Toutain, L. (1991) Un simulateur pour systèmes répartis et temps-réel. PhD thesis, Université du Havre.
- [8] Werra, D.D. Liebling, T.M. et Heche, J.F. (2003) Recherche Opérationnelle pour ingénieurs : Tome 2. Presses Polytechniques et Universitaires Romandes.