

Méthode de Monte Carlo

1 Principe	2
1.1 Calcul numérique d'une espérance	2
1.2 Calcul numérique d'une probabilité	3
2 Applications	3
2.1 Calcul d'intégrales	3
2.2 Calcul d'aires	5
2.3 Calcul de la fonction de répartition d'une variable aléatoire	8
2.4 Comparaison de différents estimateurs ponctuels	10
2.5 Calcul du niveau de confiance réel d'un inter- valle de confiance	12

Compétences attendues.

- ✓ Mettre en oeuvre la méthode de Monte-Carlo pour estimer une probabilité, la valeur d'une intégrale, ...

Introduction

On appelle communément *méthode de Monte-Carlo* toute méthode visant à estimer une quantité numérique, difficile à calculer explicitement, à l'aide de procédés probabilistes qui eux sont plus faciles à mettre en oeuvre (mais avec un risque, bien que faible, que la valeur retournée ne soit pas bonne). Les applications sont variées, comme par exemple :

- le calcul d'aires ou de volumes ;
- l'estimation du risque d'une décision financière ;
- en physique nucléaire...

Nous présentons ici le principe de cette méthode, et des exemples d'applications.

Le saviez vous ?

Le problème du jeu de Go était jusque très récemment considéré comme l'un des grands défis de l'intelligence artificielle. Il fut résolu par le développeur Google DeepMind et son programme informatique AlphaGo. Ses premières versions utilisaient justement la méthode de Monte-Carlo afin de déterminer la stratégie qui a la probabilité la plus grande de gagner la partie. En octobre 2015, il devient le premier programme à battre un joueur professionnel (le français Fan Hui) sur un goban de taille normale (19×19) sans handicap. Le 27 mai 2017, AlphaGo bat le champion du monde Ke Jie. Pour en savoir plus sur le sujet, vous pouvez consulter cette [vidéo](#).

1 Principe

1.1 Calcul numérique d'une espérance

On dispose d'une quantité C difficile à calculer directement, mais que l'on sait écrire comme l'espérance d'une variable aléatoire X . La méthode de Monte-Carlo consiste à simuler un échantillon (X_1, \dots, X_n) i.i.d. de la loi de X . Les deux résultats suivants permettent alors d'obtenir une estimation de $E(X) = C$ avec une garantie d'approximation.

- La *loi faible des grands nombres* assure que $\bar{X}_n = \frac{1}{n} \sum_{k=1}^n X_k$ est un estimateur sans biais et convergent de C .
- Le *théorème limite central* assure que $[\bar{X}_n - t_\alpha \frac{\sigma}{\sqrt{n}}, \bar{X}_n + t_\alpha \frac{\sigma}{\sqrt{n}}]$ est un intervalle de confiance asymptotique de C au niveau de confiance $1 - \alpha$, en notant t_α le réel vérifiant $\Phi(t_\alpha) = 1 - \frac{\alpha}{2}$ et σ l'écart-type de X .

Si on ne connaît pas σ , on le remplacera par un estimateur convergent de σ .

Méthode.

Pour obtenir une estimation numérique de C , on procèdera ainsi :

- on exprime C comme l'espérance d'une variable aléatoire X : $C = E(X)$;
- on réalise un échantillon i.i.d. de la loi de X de taille n avec n grand ($n = 1000$ par exemple) ;
- on renvoie la moyenne de l'échantillon obtenu (à l'aide de la commande `mean`).

1.2 Calcul numérique d'une probabilité

Une probabilité $p = P(A)$ peut être considérée comme l'espérance d'une variable de Bernoulli $X \leftrightarrow \mathcal{B}(p)$, avec $X = \begin{cases} 1 & \text{si } A \text{ est réalisée} \\ 0 & \text{sinon.} \end{cases}$. La méthode de Monte-Carlo s'applique donc également dans ce cas pour obtenir une estimation de la probabilité d'un l'évènement.



Méthode.

Pour obtenir une estimation numérique de p , on procèdera ainsi :

- on identifie l'épreuve de Bernoulli associée à la probabilité $p = P(A)$;
- on réalise n fois l'épreuve de Bernoulli de manière indépendante avec n grand ;
- on compte le nombre c de succès dans cette succession d'épreuves de Bernoulli indépendantes ;
- on renvoie la fréquence de réussite $\frac{c}{n}$ de l'épreuve de Bernoulli.

2 Applications

2.1 Calcul d'intégrales

Exercice 1 (★★)

Soit (U_n) une suite de variables aléatoires indépendantes suivant toutes la loi uniforme sur $[0, 1]$, et soit $g : [0, 1] \rightarrow \mathbb{R}$ une fonction continue. Pour tout $n \geq 1$, on pose $S_n = \frac{1}{n} \sum_{i=1}^n g(U_i)$.

1. En utilisant la loi faible des grands nombres, montrer que :

$$S_n \xrightarrow{P} \int_0^1 g(x) dx.$$

À quelle propriété de cours ce résultat vous fait-il penser ?

2. On souhaite déterminer une valeur approchée de l'intégrale $I = \int_0^1 \cos(x^2) dx$.

Compléter le programme suivant afin qu'il retourne une valeur approchée de I .

```

1 | function y = g(x)
2 |     y = ...
3 | endfunction
4 | n = 1000 ; S = 0 ;
5 | for i=1:n
6 |     S = S + ...
7 | end
8 | disp(...)
```

Essayer ce programme avec différentes valeurs de n , en faisant plusieurs essais pour chaque valeur de n . Comparer les valeurs obtenues avec celle donnée par `integrate('g(x)', 'x', 0, 1)`.

1. Vérifions les hypothèses de la LFGN. Les U_i étant indépendantes, les variables $g(U_i)$ le sont aussi par lemme de coalition. De plus si U suit une loi $\mathcal{U}([0, 1])$, et en notant $f_U : x \mapsto \begin{cases} 1 & \text{si } 0 \leq x \leq 1 \\ 0 & \text{sinon.} \end{cases}$, on a :

- $g(U)$ admet une espérance si et seulement si $\int_{-\infty}^{+\infty} g(t) f_U(t) dt = \int_0^1 g(t) dt$ converge

absolument par le théorème de transfert. Or c'est l'intégrale d'une fonction continue sur un segment, donc c'est bien le cas, et on a :

$$E(g(U)) = \int_0^1 g(t)dt.$$

- $g(U)$ admet une variance si et seulement si $\int_{-\infty}^{+\infty} g(t)^2 f_U(t)dt = \int_0^1 g(t)^2 dt$ converge absolument toujours par le théorème de transfert. Ce qui est bien le cas car là aussi, il s'agit d'une intégrale d'une fonction continue sur un segment.

On peut donc conclure par la loi faible des grands nombres que :

$$S_n \xrightarrow{P} \int_0^1 g(x)dx.$$

Ce résultat est comparable au théorème des sommes de Riemann : si $g : [0, 1] \rightarrow \mathbb{R}$ est continue, alors on a (méthode des rectangles à gauche) :

$$\frac{1}{n} \sum_{k=0}^{n-1} g(k/n) \xrightarrow{n \rightarrow +\infty} \int_0^1 g(x)dx.$$

Ici les points de $[0, 1]$ pour constituer la somme sont pris de façon équirépartie sur $[0, 1]$. Ils le sont de façon aléatoire dans cet exercice.

2. On cherche une fonction g continue sur $[0, 1]$ telle que :

$$\int_0^1 g(x)dx = \int_0^1 \cos(x^2)dx.$$

Prenons donc $g : x \mapsto \cos(x^2)$. On peut alors compléter le programme proposé :

```

1  function y = g(x)
2      y = cos(x^2)
3  endfunction
4
5  n = 1000 ; S = 0 ;
6  for i=1:n
7      S = S + g(rand())
8  end
9  disp(S/n)

```

On obtient par exemple pour $n = 1000$ en exécutant deux fois le programme 0.9118163 et 0.8988760 respectivement. La valeur obtenue par la commande `integrate('g(x)', 'x', 0, 1)` donne 0.9045242. On obtient donc une valeur approchée de cette intégrale. On pourrait augmenter n pour préciser notre estimation de l'intégrale.

Remarque. Dans l'exemple précédent, l'intérêt d'une méthode probabiliste n'est pas flagrant, puisque nous aurions également pu utiliser la méthode de rectangles (c'est-à-dire les sommes de Riemann) pour obtenir une valeur approchée de l'intégrale, sans risque d'erreur (il n'y a pas de hasard dans la méthode des rectangles). En revanche, cette méthode peut aisément être adaptée pour des intégrales impropres, pour lesquelles la méthode des rectangles marche mal.

Exercice 2 (★)

Estimer la valeur des intégrales suivantes à l'aide de la méthode de Monte-Carlo.

$$\bullet I = \int_0^{+\infty} \frac{e^{-x}}{1+x^4} dx \qquad \bullet J = \int_{-2}^4 e^{-x^2} dx.$$

Essayons d'adapter le raisonnement de l'exercice précédent. Pour la première intégrale, on cherche U une variable dont une densité est f_U et g une fonction continue sur un intervalle à déterminer tels que :

$$\int_{-\infty}^{+\infty} g(x)f_U(x)dx = \int_0^{+\infty} \frac{e^{-x}}{1+x^4} dx.$$

Cela suggère de prendre $f_U : x \in \mathbb{R} \mapsto \begin{cases} e^{-x} & \text{si } x \geq 0 \\ 0 & \text{sinon.} \end{cases}$, et donc $U \hookrightarrow \mathcal{E}(1)$, et $g : x \in \mathbb{R}_+ \mapsto$

$\frac{1}{1+x^4}$. On obtient donc le code suivant :

```

1  function y = g(x)
2      y = 1/(1+x^4)
3  endfunction
4
5  n = 1000 ; S = 0 ; lbd = 1 ;
6  for i=1:n
7      S = S + g(grand(1,1,'exp',1/lbd))
8  end
9  disp(S/n)

```

On obtient 0.6129161 comme estimation ponctuelle de la valeur de I .

Pour J en procédant toujours par analogie avec ce qui a été fait avant, on est amené à prendre $f_U : x \in \mathbb{R} \mapsto \begin{cases} \frac{1}{6} & \text{si } -2 \leq x \leq 4 \\ 0 & \text{sinon.} \end{cases}$, et donc $U \hookrightarrow \mathcal{U}([-2,4])$, et $g : x \in \mathbb{R}_+ \mapsto 6e^{-x^2}$. On

obtient donc le code suivant :

```

1  function y = g(x)
2      y = 6*exp(-x^2)
3  endfunction
4
5  n = 1000 ; S = 0 ;
6  for i=1:n
7      S = S + g(grand(1,1,'unf',-2,4))
8  end
9  disp(S/n)

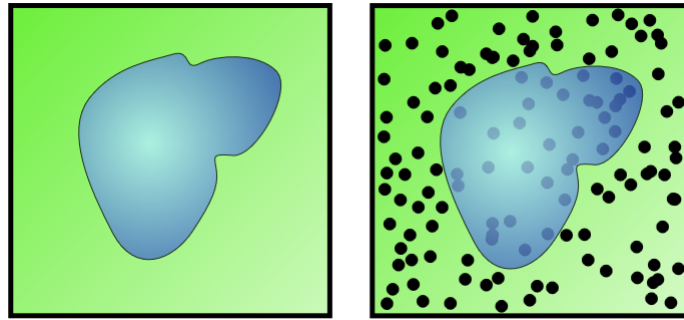
```

On obtient 1.7047088 comme estimation ponctuelle de la valeur de J .

2.2 Calcul d'aires

Il est également possible d'utiliser des raisonnements probabilistes pour approcher des aires.

Considérons par exemple un terrain carré dont la longueur des côtés, et donc l'aire $\mathcal{A}_{\text{terrain}}$, sont connues. Au sein de cette zone se trouve un lac dont la superficie est inconnue. Pour trouver l'aire du lac \mathcal{A}_{lac} , on demande à une armée de tirer N coups de canon de manière aléatoire sur cette zone.



La probabilité p qu'un boulet soit tombé dans le lac est égale à :

$$p = \frac{\mathcal{A}_{\text{lac}}}{\mathcal{A}_{\text{terrain}}}.$$

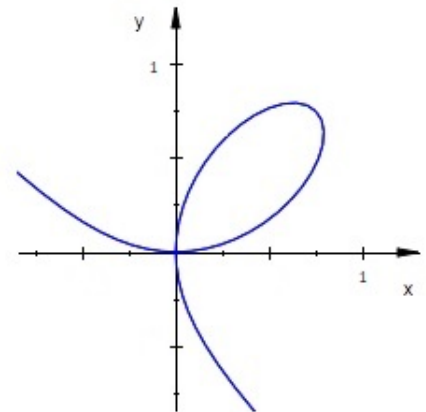
Pour estimer la superficie du lac, il suffit de compter le nombre n de boulets qui sont restés sur le terrain. En effet pour N grand, on a par le théorème d'or de Bernoulli :

$$p \approx \frac{N - n}{N} \quad \text{et donc} \quad \mathcal{A}_{\text{lac}} \approx \frac{N - n}{N} \times \mathcal{A}_{\text{terrain}}.$$

Exercice 3 (★)

Considérons la courbe ci-contre (appelée folium de Descartes), dont on cherche à calculer une valeur approchée de l'aire formée par la boucle. On admet que l'intérieur de cette boucle est l'ensemble :

$$\mathcal{B} = \left\{ (x, y) \in \mathbb{R}^2 : x \geq 0, y \geq 0, x^3 + y^3 \leq \frac{3xy}{2} \right\}.$$



Folium de Descartes.

Compléter le programme suivant afin qu'il retourne une valeur approchée de l'aire de \mathcal{B} .

```

1 | n = 10000
2 | X = grand(1,n, 'unf', 0, 1)
3 | Y = .....
4 | S = 0 ;
5 | for i = 1:n
6 |     if ..... then
7 |         S = S + 1 ;
8 |     end
9 | end
10 | disp(.....)

```

En suivant le procédé expliqué au début de cette section, on prend des points au hasard dans $[0, 1] \times [0, 1]$, et on teste si ces points sont à l'intérieur de la boucle ou non. On compte alors le nombre de points satisfaisant cette condition, et on renvoie leur fréquence. Ce qui donne :

```

1  n = 10000
2  X = grand(1,n,'unf',0,1)
3  Y = rand(1,n)
4  S = 0 ;
5
6  for i = 1:n
7      if X(i)^3+Y(i)^3 <= 3*X(i)*Y(i)/2
8          S = S + 1 ;
9      end
10 end
11 disp(S/n)

```

On obtient 0.382 comme estimation ponctuelle de l'aire de cette boucle.

Exercice 4 (★★ - Calcul d'une valeur approchée de π)

Soit $\mathcal{D} = \{(x, y) \in \mathbb{R}^2, x, y \geq 0, x^2 + y^2 \leq 1\}$

- Représenter \mathcal{D} . Quelle est son aire ?
- Soient $(X_i)_{1 \leq i \leq n}$ et $(Y_i)_{1 \leq i \leq n}$ des variables i.i.d. suivant la loi $\mathcal{U}([0, 1])$. On note T_i la variable qui vaut 1 si le point (X_i, Y_i) appartient à \mathcal{D} , et 0 sinon.
 - Déterminer la loi de T_i .
 - On pose $\overline{T}_n = \frac{1}{n} \sum_{i=1}^n T_i$. Montrer que $4\overline{T}_n$ converge en probabilité vers π .
- Écrire un programme qui calcule une valeur approchée de π .
- Déterminer un intervalle de confiance asymptotique de π , au niveau de risque $\alpha = 0.05$, dont les bornes dépendent de \overline{T}_n .
 - Quelle valeur de n faut-il choisir pour obtenir une approximation de π à $\varepsilon = 0.01$ près, au niveau de risque $\alpha = 0.05$?

- \mathcal{D} correspond à un quart du disque de centre $(0, 0)$ et de rayon 1. Il est d'aire $\frac{\pi 1^2}{4} = \frac{\pi}{4}$.
- T_i suit une loi de Bernoulli de paramètre p où p représente la probabilité qu'un point pris au hasard dans $C = [0, 1] \times [0, 1]$ appartienne à \mathcal{D} , soit :

$$p = \frac{\mathcal{A}_{\mathcal{D}}}{\mathcal{A}_C} = \frac{\pi}{4}.$$

Ainsi T_i suit la loi $\mathcal{B}(\pi/4)$.

- Les T_i étant indépendants, car les X_i et Y_i le sont, et admettant une variance, on a par la loi faible des grands nombres que \overline{T}_n converge en probabilité vers $E(T_1) = \frac{\pi}{4}$. En composant par $f : x \in \mathbb{R} \mapsto 4x$ continue, on en déduit que $4\overline{T}_n$ converge en probabilité vers $E(4T_1) = \pi$.
- On adapte le programme de l'exercice précédent :

```

1  n = 10000
2  X = rand(1,n)
3  Y = rand(1,n)
4  T = 0 ;
5
6  for i = 1:n
7      if X(i)^2+Y(i)^2 <= 1 then
8          T = T + 1 ;
9      end
10 end
11 disp(4*T/n)

```

On obtient 3.1216 comme valeur approchée de π .

4. (a) Les calculs ont déjà été effectués en cours : on a obtenu qu'un intervalle de confiance de $\frac{\pi}{4}$ au niveau de risque $\alpha = 0.05$ est donné par $\left[\bar{T}_n - \frac{t_\alpha}{2\sqrt{n}}; \bar{T}_n + \frac{t_\alpha}{2\sqrt{n}} \right]$. Ainsi un intervalle de confiance de π au niveau de risque $\alpha = 0.05$ est donné par $\left[4\bar{T}_n - \frac{2t_\alpha}{\sqrt{n}}; 4\bar{T}_n + \frac{2t_\alpha}{\sqrt{n}} \right]$. À l'aide de l'estimation observée précédente, on obtient l'un intervalle de confiance observé [3.0824, 3.1608]. On notera qu'il contient bien π , on avait théoriquement une probabilité de 0.95 que ça soit effectivement le cas.
- (b) Il faut prendre n de sorte que :

$$\frac{2t_\alpha}{\sqrt{n}} \leq 0.01 \quad \Leftrightarrow \quad n \geq 153664.$$

On doit prendre $n = 153664$ pour obtenir un intervalle de confiance de π au niveau de confiance de 0.95. On notera que cette valeur est très élevée, et que notre méthode ne permet pas de converger très rapidement vers le nombre π .

2.3 Calcul de la fonction de répartition d'une variable aléatoire

Exercice 5 (★★ - Fonction de répartition empirique de la loi $\mathcal{N}(0,1)$)

- Écrire une fonction `phi_empirique` qui, à un réel x entré par l'utilisateur, renvoie une estimation de $\Phi(x)$ par la méthode de Monte-Carlo.
- On rappelle que `[P,Q]=cdfnor("PQ",x,m,s)` renvoie $P = P(X \leq x)$ et $Q = 1 - P(X \leq x)$ où $X \hookrightarrow \mathcal{N}(m, s)$. Comparer les résultats obtenus à l'aide de la fonction `phi_empirique` avec ceux renvoyés par la commande `cdfnor`.

- On va créer un échantillon \mathbf{E} de taille N grand de la loi normale $\mathcal{N}(0,1)$. On cherchera ensuite parmi cet échantillon la proportion des modalités $\mathbf{E}(\mathbf{k})$ plus petites qu'un réel donné x , en comptant le nombre \mathbf{C} de telles modalités. On renverra alors cette proportion \mathbf{p} qui est une estimation de $P(X \leq x) = \Phi(x)$ où $X \hookrightarrow \mathcal{N}(0,1)$. Ce qui donne la fonction suivante :


```

1 function p = phi_empirique(x)
2     N = 10000
3     E = grand(1,N,'nor',0,1)
4     C = 0
5     for k = 1:N
6         if E(k) <=x then
7             C = C+1
8         end
9     end
10    p = C/N
11 endfunction

```

2. En ajoutant par exemple les commandes :

```

1 [P,Q] = cdfnor('PQ',1,0,1)
2 disp(phi(1),P)

```

on obtient 0.8413447 avec la fonction `cdfnor` et 0.847 avec `phi_empirique`.

Remarque. La méthode de Monte-Carlo avait déjà été utilisée au TP5 (sans la nommer ainsi) pour tracer la fonction de répartition empirique G_X d'une variable aléatoire X . Rappelons-en le principe :

- on génère un échantillon observé E distribué suivant la loi X ;
- on trie cet échantillon par modalités croissantes à l'aide de la commande `y = tabul(E,'i')` ;
- on trace la fonction en escalier G_X d'abscisse les modalités et d'ordonnée la fréquence cumulée croissante de ces modalités à l'aide de la commande `plot2d2(y(:,1),cumsum(y(:,2))/length(E))`.

Pour tout $x \in \mathbb{R}$, $G_X(x)$ correspond à la fréquence des modalités inférieures ou égales à x dans l'échantillon E . Ce qui donne bien à une estimation de $F_X(x) = P(X \leq x)$.

Exercice 6 (★)

Tracer la fonction de répartition empirique de $Z = XY$ où $X \leftrightarrow \mathcal{P}(2)$ et $Y \leftrightarrow \mathcal{E}(1)$ sont indépendantes.

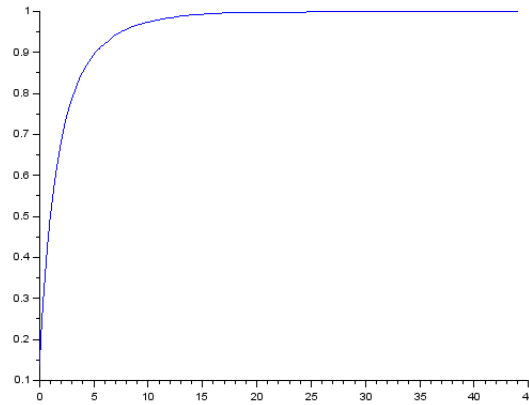
En utilisant les commandes rappelées plus haut, on obtient :

```

1 N = 10000 //taille de l'echantillon
2 X = grand(1,N,'poi',2)
3 Y = grand(1,N,'exp',1/1)
4 Z = X.*Y //echantillon de la loi de Z
5 y = tabul(Z,'i')
6 plot2d2(y(:,1),cumsum(y(:,2))/length(Z))

```

qui génère le graphe suivant :



2.4 Comparaison de différents estimateurs ponctuels

Exercice 7 (★★)

Soit (X_1, \dots, X_n) un échantillon de loi mère $\mathcal{B}(p)$. On note $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$ la moyenne empirique.

Nous disposons des deux estimateurs convergents de l'écart-type $\sqrt{p(1-p)}$:

$$\bullet S_n = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (X_k - \bar{X}_n)^2} ; \quad \bullet T_n = \sqrt{\bar{X}_n(1 - \bar{X}_n)}.$$

On souhaite comparer ces deux estimateurs. Prenons pour commencer $p = \frac{1}{2}$ et $n = 1000$.

1. On considère les instructions suivantes :

```

1 | p = 0.5 ; n = 1000 ; theta = sqrt(p*(1-p))
2 | E = grand(n,1000,'bin',1,1/2) ;
3 | S = stdev(E,'r') ;
4 | bS = mean(S)-theta ;
5 | rS = mean((S-theta).^2) ;

```

Que contiennent les variables S, bS et rS ?

- Estimer de même le biais et le risque quadratique de T_n comme estimateur de l'écart-type.
- Recommencer avec d'autres valeurs de p et de n . Quel estimateur privilégieriez vous ?
- On souhaite enfin comparer ces deux estimateurs à l'aide de leur histogramme des fréquences. Écrire pour cela un programme qui :

- simule $m = 1000$ n -échantillons de la loi $\mathcal{B}(p)$;
- calcule, pour chaque échantillon observé, l'estimation correspondante obtenue à l'aide de chacun des estimateurs S_n et T_n ;
- trace l'historgramme des 10000 estimations obtenues avec chacun des estimateurs (on pourra utiliser les commandes `subplot(1,2,1)` et `subplot(1,2,2)` pour tracer les histogrammes sur la même fenêtre graphique l'un à côté de l'autre). On prendra `c = 0:0.01:1` pour vecteur définissant les classes.

Comparer les deux histogrammes. Recommencer pour d'autres valeurs de p . Quel semble être le meilleur estimateur ?

1. La variable `E` contient une matrice de taille $1000 \times n$ de réalisations de la loi $\mathcal{B}(1/2)$. La commande `stdev(E, 'r')` calcule l'écart-type de chacune des colonnes de `E` ('r' pour row, indiquant qu'on fait l'écart-type suivant les lignes de la matrice). `S` contient donc n réalisations de S_n . `mean(S)` contient la moyenne de ces réalisations, ce qui correspond à une estimation de $E(S_n)$. Ainsi la variable `bS` contient une estimation du biais $b_\theta(S_n)$. De même, `rS` contient une estimation de $E((S_n - p)^2)$, c'est-à-dire du risque quadratique $r_\theta(S_n)$.

En exécutant ce code, on obtient `bS = -0.0000045` et `rS = 0.0000001`.

2. On adapte ce qui a été fait plus haut :

```

1 | p = 0.5 ; n = 1000 ; theta = sqrt(p*(1-p))
2 | E = grand(n,1000,'bin',1,1/2) ;
3 | M = mean(E, 'r') ;
4 | T = sqrt(M.*(1-M))
5 | bT = mean(T)-theta ;
6 | rT = mean((T-theta).^2) ;

```

On obtient ici `bT = - 0.0002296` et `rT = 0.0000002`.

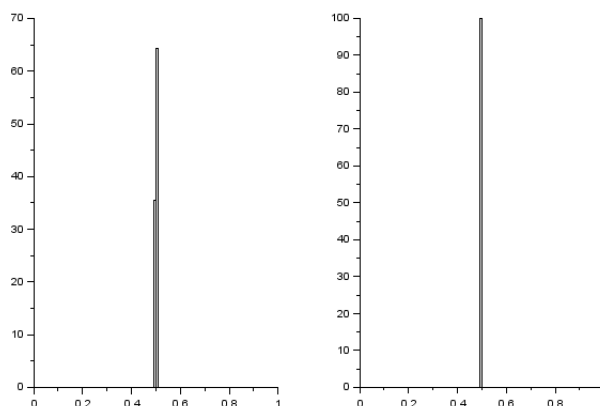
3. En exécutant plusieurs fois ce code pour différentes valeurs de p , on obtient que $r_\theta(T_n)$ semble plus petit que $r_\theta(S_n)$, sauf pour $p = 0.5$. Difficile de trancher donc entre ces deux estimateurs.
4. On peut utiliser le programme suivant :

```

1 | p = 0.5, n = 1000, m = 1000
2 | E = grand(n,m,'bin',1,p)
3 | S = stdev(E, 'r')
4 | M = mean(E, 'r') ;
5 | T = sqrt(M.*(1-M))
6 | c = 0:0.01:1
7 | subplot(1,2,1)
8 | histplot(c,S)
9 | subplot(1,2,2)
10 | histplot(c,T)

```

On obtient les histogramme suivants :



On peut recommencer avec différentes valeurs de p . Mais difficile de trancher sur la base des histogrammes qui sont très similaires.

2.5 Calcul du niveau de confiance réel d'un intervalle de confiance

Afin de déterminer des intervalles de confiance, nous aurons souvent besoin d'inverser la fonction Φ , notamment pour déterminer le réel t_α tel que $\Phi(t_\alpha) = 1 - \frac{\alpha}{2}$, soit encore $t_\alpha = \Phi^{-1}(1 - \frac{\alpha}{2})$. Nous utiliserons à cet effet la commande `cdfnor` de la manière suivante : pour déterminer l'unique réel x tel qu'une variable aléatoire $X \hookrightarrow \mathcal{N}(\mu, \sigma)$ vérifie $P(X \leq x) = p$, on saisit :

```
1 | x = cdfnor('X',mu,sigma,p,1-p)
```

Par exemple, si l'on souhaite déterminer le nombre x tel que $\Phi(x) = 0.75$, on entre `x = cdfnor('X',0,1,0.75,0.25)`.

Exercice 8 (★ - Des valeurs classiques)

Avec les notations habituelles, déterminer les valeurs de t_α lorsque $\alpha = 0.05$ ou $\alpha = 0.01$.

On utilise le code suivant :

```
1 | alpha = 0.05
2 | p = 1-alpha/2
3 | x = cdfnor('X',0,1,p,1-p)
4 | disp(x)
```

qui renvoie la valeur 1.959964. Pour $\alpha = 0.01$, on obtient la valeur 2.5758293.

Exercice 9 (★★ - Intervalle de confiance de l'espérance d'une loi normale)

Nous prouverons en TD que si X_1, \dots, X_n sont i.i.d. suivant la loi normale $\mathcal{N}(\theta, 1)$, alors $\left[\bar{X}_n - \frac{t_\alpha}{\sqrt{n}}, \bar{X}_n + \frac{t_\alpha}{\sqrt{n}} \right]$ est un intervalle de confiance de θ au niveau de confiance $1 - \alpha$.

1. Étendue de l'intervalle de confiance.

- Écrire une fonction `y=etendue(n,alpha)` qui prend en paramètres un entier n et un réel $\alpha \in]0, 1[$, et renvoie l'étendue de l'intervalle de confiance de θ correspondant.
- Essayer ce programme pour $\alpha = 0.05$ en faisant varier n . Puis à n fixé, étudier l'étendue de l'intervalle pour $\alpha = 0.05$, $\alpha = 0.01$, $\alpha = 0.001$ et $\alpha = 0.0001$. Que constate-t-on ?

2. Niveau de confiance réel.

Prenons dans la suite $\alpha = 0.05$ et $n = 1000$. On souhaite déterminer le niveau de confiance réel de l'intervalle de confiance par la méthode de Monte-Carlo.

(a) Qu'effectue le code suivant :

```

1 | theta = grand(1,1,'exp',1) ; n = 1000 ; m = 1000 ; t=1.96 ;
2 | E = grand(m,n,'nor',theta,1)
3 | Xbar = mean(E,"r")
4 | c = 0 ;
5 | for k = 1:m
6 |     if abs(Xbar(k)-theta)<t/sqrt(n) then
7 |         c = c+1 ;
8 |     end
9 | end
10 | disp(theta,"theta :") ;
11 | disp(100*c/m,"Proportion d IdC contenant theta :")

```

(b) Exécuter plusieurs fois ce code. Est-ce conforme à ce que vous vous attendiez ?

1. (a) La fonction suivante convient :

```

1 | function y = etendue(n,alpha)
2 |     p = 1-alpha/2
3 |     t = cdfnor('X',0,1,p,1-p)
4 |     y = 2*t/sqrt(n)
5 | endfunction

```

(b) On constate qu'à n fixé, l'étendue augmente si α diminue. Et à α fixé, l'étendue diminue si n augmente.

2. On choisit le paramètre `theta` au hasard à l'aide d'une loi exponentielle $\mathcal{E}(1)$. On crée ensuite $m = 1000$ échantillons de taille $n = 1000$ de la loi $\mathcal{N}(\theta, 1)$. On calcule alors 1000 réalisations de \overline{X}_n à l'aide de la commande `Xbar = mean(E,'r')`. La boucle `for` permet alors de compter sur les 10000 intervalles de confiance observés, le nombre d'intervalles contenant effectivement `theta`. On stocke ce nombre dans la variable `c`, et on renvoie le pourcentage d'intervalles de confiances qui contiennent `theta`. On obtient ainsi une estimation du niveau de confiance réelle de l'intervalle.

3. On obtient en exécutant plusieurs fois le code :

θ réel	Proportion d'IdC contenant θ
0.4844947	94.6
0.9987862	93.7
0.7165302	94.4
2.7665696	95.6

On obtient un niveau de confiance réel proche de 95%, conforme à ce qu'on attendait.

 **Le saviez vous ?**

Les méthodes de Monte-Carlo ont été développées par les physiciens J. Von Neumann et S. Ulam du Projet Manhattan lors de l'élaboration en secret les premières bombes atomiques au laboratoire national de Los Alamos (Nouveau-Mexique) au début des années 1940. Ils disposaient alors du tout premier ordinateur électronique, l'ENIAC, qui pesait 30 tonnes, calculait à peu près 10 millions de fois moins vite que le dernier Iphone et était très souvent en panne.

Il s'agissait évidemment d'un énorme progrès autorisant des calculs qu'il n'aurait pas été possible de faire à la main, mais les capacités de calcul étaient tout de même limitées.

Les utilisateurs de l'ENIAC développèrent alors les méthodes expliquées précédemment pour faire des calculs approchés avec le moins d'opérations possible afin de minimiser le temps d'utilisation de leur ordinateur. L'avantage des méthodes expliquées précédemment est qu'on sait à l'avance combien d'opérations seront nécessaires, puisque nous fixons nous même la valeur de n . La contrepartie est qu'on obtient alors des valeurs qui sont peut-être inexactes... Puisque le projet était secret, il lui fallait un nom de code, et le nom de Monte-Carlo fut choisi en référence aux jeux de hasard des casinos monégasques.

Les algorithmes probabilistes sont généralement regroupés en trois catégories :

- les algorithmes de Monte-Carlo, qui retournent des valeurs « probablement correctes » en un temps déterminé à l'avance ;
- les algorithmes de Las Vegas qui donnent des valeurs exactes en temps probablement court (mais qui peut éventuellement être très long) ;
- les algorithmes d'Atlantic City qui donnent une réponse probablement correcte en un temps probablement court (mais qui peuvent éventuellement donner une « mauvaise » réponse avec un temps de calcul long).