ECOLE NATIONALE D'INGÉNIEURS DE TUNIS

DÉPARTEMENT : GÉNIE INDUSTRIEL

Filière : Modélisation pour l'Industrie et le Service

Projet Probabilités :

Ruine du joueur

Elaboré par :

Ben Brahim Sami

Encadré par :

M.Makhlouf Azmi

Table des matières

Remerciement

Je tiens à remercier M. Azmi Makhlouf qui a contribué au succès de mon mini-projet et qui m'a aidé lors de la rédaction de ce rapport. Son écoute et ses conseils m'ont permis de mieux comprendre la marche aléatoire et de surmonter les difficultés affrontées.

Introduction

La "ruine du joueur" est un problème qui dérive des marches aléatoires. Il est un exemple d'une chaîne de Markov et d'une martingale. Donc il constitue un modèle mathématique d'un système possédant une dynamique discrète composée d'une succession de pas aléatoires. Ces pas aléatoires sont indépendants les uns aux autres, c'est-à-dire à chaque instant, le futur du système dépend de son état présent, mais pas de son passé, même le plus proche. Autrement dit, le système « perd la mémoire » à mesure qu'il évolue dans le temps. C'est pourquoi la marche aléatoire est appelée aussi « marche de l'ivrogne ». Les domaines d'applications de ce processus sont nombreux. On le trouve souvent présent en mathématiques, finance(évolution du prix d'un actif), biologie et écologie(trajectoires de bactéries ou d'animaux, évolution de la taille d'une population),... D'apparence simple, son étude théorique se révèle pourtant assez compliquée. C'est pourquoi au cours de l'étude théorique, on a admis certaines formules. Dans ce mini-projet, on s'intéresse plus à l'étude numérique. On va étudier la probabilité de ruine d'un joueur de casino et le temps moyen du jeu pour un groupe de joueurs. Ainsi, on va découvrir le caractère des jeux de casinos. Est-ce qu'il s'agit d'un jeu équitable ou bien d'un jeu truqué ?

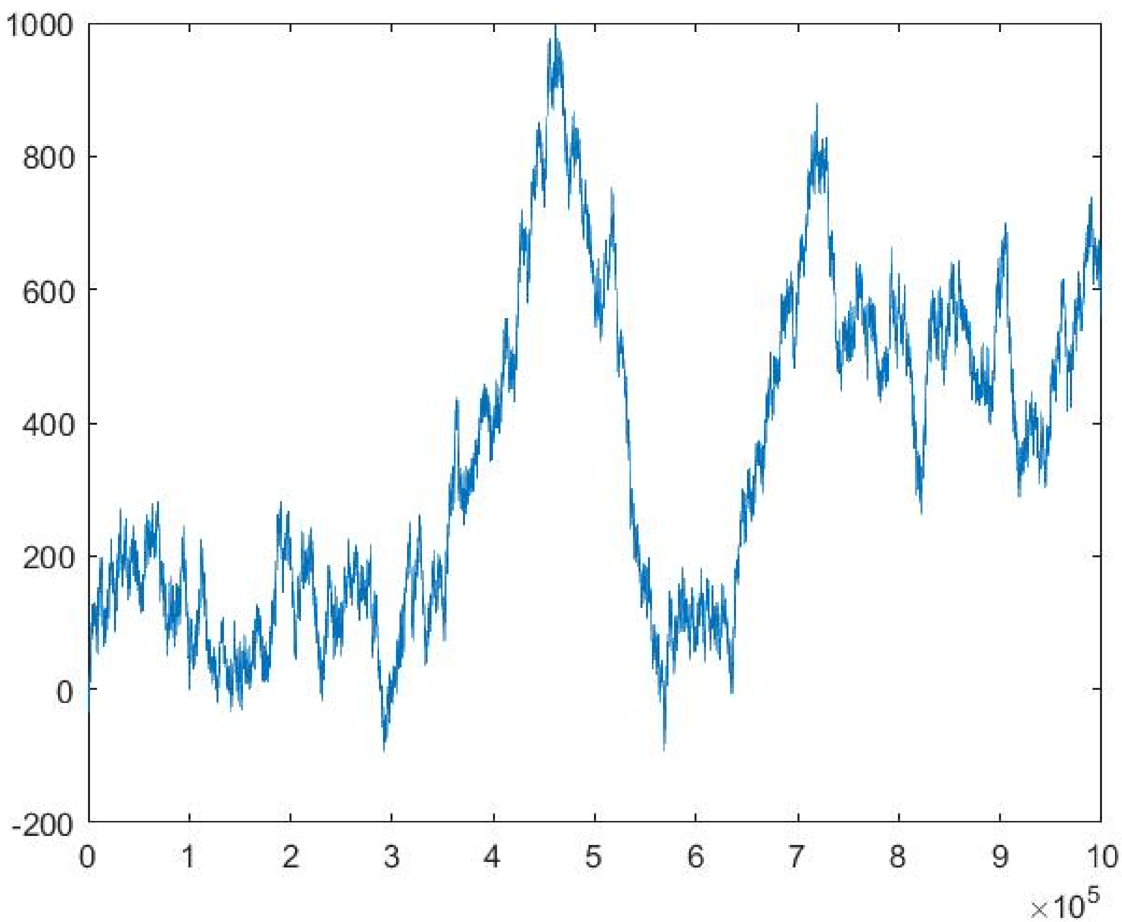

FIGURE 1: Marche aléatoire avec un pas=+1/-1

Chapitre 1

Etude théorique

Enoncé du problème :

On considère une somme fixée de S dinars ($S \in \mathbb{N}^*$), qui sera à partager entre deux joueurs A et B (e.g A est vous comme joueur, et B est le casino). La fortune initiale de A vaut $X_0 \in \{0, 1, ..., S\}$ et celle de B vaut S - X_0. Le jeu se fait en des tours successifs aux instants $n \in \mathbb{N}$. A chaque tour, A peut gagner 1 dinar de plus avec probabilité $p \in]0,1[$, et dans ce cas B perd 1 dinar en moins. Inversement, A peut perdre 1 dinar avec probabilité q=1-p, et dans ce cas B gagne 1 dinar.
Au bout du n-ième tour, la fortune de A vaut X_n et celle de B vaut S-X_n, toutes deux suivant une marche aléatoire. Le jeu s'arrête dès que l'un des deux joueurs est ruiné,c-à-d. dès que $X_n \in \{0, S\}$. Ainsi, pour tout $n \in \mathbb{N}$,$X_n \in \{0, 1, ..., S\}$.
La problématique qu'on se pose ici concerne :

- la probabilité que A(ou B) se ruine ?
- la durée moyenne du jeu ?

Probabilités de ruine :

On s'intéresse à l'évènement :
R_A = " A se ruine à un certain moment " $= \bigcup(X_n = 0)$, et aux probabilités de ruine :
$f_S(k) := \mathbb{P}(R_A | X_0 = k), 0 \leq k \leq S$.
On a évidemment $f_S(0) = 1$ et $f_S(S) = 0$.

Le cas $S = 2$:
Si k = 1 alors la seule possibilité de ruine pour A est de perdre 1 dinar au premier tirage. Donc $f_2(1) = q$.

Le cas S = 3 :

Si k = 1 alors le déroulement du jeu sera ainsi :
$1 \xrightarrow{p} 2 \xrightarrow{q} 1 \xrightarrow{p} 2 \xrightarrow{q} 1 ... 1 \xrightarrow{p} 2 \xrightarrow{q} 1 \xrightarrow{q} 0.$

Donc on remarque que cet évènement est une suite de transitions $1 \xrightarrow{p} 2 \xrightarrow{q} 1$ dont la probabilité de chacune est pq. Cette suite se termine par la transition $1 \xrightarrow{q} 0$ de probabilité q. On sait bien que les tirages sont 2 à 2 indépendants. Alors :

$f_3(1) = q\sum(pq)^n = \frac{q}{1-pq}$

Si k = 2 alors le déroulement du jeu sera ainsi :
$2 \xrightarrow{q} 1 \xrightarrow{p} 2 \xrightarrow{q} 1 \xrightarrow{p} 2 ... 2 \xrightarrow{q} 1 \xrightarrow{p} 2 \xrightarrow{q} 1 \xrightarrow{q} 0.$

Donc on remarque que cet évènement est une suite de transitions $2 \xrightarrow{q} 1$ q dont la probabilité de chacune est pq. Cette suite se termine par une double transition finale $2 \xrightarrow{q} 1 \xrightarrow{q} 0$ de probabilité q^2.

Enfin $f_3(2) = q^2\sum(pq)^n = \frac{q^2}{1-pq}$

Cas général

Clairement, les raisonnements précédents se compliquent pour $S \geq 4$.

On admet la relation de récurrence suivante, basée sur une *analyse à un pas* (qui exprime un conditionnement par rapport à la première transition de X_0 vers X_1) :

$f_S(k) = pf_S(k+1) + qf_S(k-1), 1 \leq k \leq S-1.$

On peut vérifier que la solution de cette équation (avec les conditions aux bords en $k = 0$ et $k = S$) est, pour tous $0 \leq k \leq S$,

$$f_S(k) = P(R_A|X_0 = k) = \frac{(q/p)^k - (q/p)^S}{1-(q/p)^S}, si\ p \neq q \text{(jeu non équilibré)};$$

$$f_S(k) = \mathbb{P}(R_A|X_0 = k) = 1 - \frac{k}{S}, si\ p = q = 0.5 \text{(jeu équilibré)}$$

Si $p = q = 0.5$: $f_\infty(k) = 1$

Si $p < q$: $f_\infty(k) = 1$

Si $p > q$: $f_S(k) = \frac{(q/p)^k - (q/p)^S}{1-(q/p)^S} \xrightarrow{S \to +\infty} (q/p)^k = f_\infty(k)$

Donc :

Si $q \geq p$: $f_\infty(k) = 1 \longrightarrow$ Le joueur A est ruiné.

Si $p > q$: $f_\infty(k) = (q/p)^k < 1 \longrightarrow$ Plus la probabilité du joueur A de gagner 1 dinar augmente plus sa probabilité de ruine diminue.

En supposant que le jeu est fini, c-à-d nécessairement l'un des deux joueurs doit se ruiner, alors on peut dire que l'évènement " le joueur B se ruine " est le complémentaire de l'évènement " Le joueur A se ruine ". D'où : $g_S(k) = 1 - f_S(k)$.

Durée moyenne du jeu

Soient $T_{0,S}$ la durée totale du jeu :

$T_{0,S} := \inf\{n \in \mathbb{N} t.q. X_n = 0 ou X_n = S\}$, et $h_S(k)$ la durée moyenne du jeu lorsque A part d'une fortune initiale X_0 =k:

$h_S(k) := \mathbb{E}[T_{0,S}|X_0 = k]$.

Avec la même approche que pour les probabilités de ruine, on peut montrer (admis) que

$h_S(k) = 1 + p h_S(k-1) + q h_S(k+1)$,

et que, pour tous $0 \leq k \leq S$,

$h_S(k) = \mathbb{E}[T_{0,S}|X_0 = k] = \frac{1}{q-p}(k - S\frac{1-(q/p)^k}{1-(q/p)^S})$, $si\, p \neq q$ (jeu non équilibré) ;

$h_S(k) = \mathbb{E}[T_{0,S}|X_0 = k] = k(S-k)$, $si\,$p=q= 0.5 (jeu équilibré).

Si $p = q = 0.5$: $h_\infty(k) = \infty$: Quand les deux joueurs ont la même chance de gagner 1 dinar, le temps du jeu tend vers l'infini.

Si $p > q$: $h_S(k) = \frac{1}{q-p}(k - S\frac{1-(q/p)^k}{1-(q/p)^S}) \sim \frac{S}{p-q} \to +\infty$

$\longrightarrow$ $Si\, p \geq q : h_\infty(k) = +\infty$: Même si le joueur a plus de probabilité de gagner, le temps moyen du jeu tend vers l'infini.

Si $q > p$: $h_S(k) \sim \frac{(p/q)^S}{S}\frac{(q/p)^k}{p-q} \to 0$

$\longrightarrow$ Donc même si le casino a un léger avantage, le temps moyen du jeu tend vers 0. Ceci montre que le jeu suit une stratégie qui assure l'avantage du casino par rapport au joueur. C'est ce qui explique la richesse des casinos.

Chapitre 2

Etude numérique

2.1 APPROXIMER LA PROBABILITÉ DE RUINE

L'idée est la suivante. Pour approcher la probabilité de ruine f_S(k) = $\mathbb{P}$(R$_A$|X$_0$=k), on simule un grand nombre M de scénarii possibles de notre jeu (e.g. M = 1000 ou 10 000), indépendants, et tous à la même fortune S et partant du même $X_0 = k$. On réalise les scénarii sur le même intervalle de temps [0, N] assez grand (e.g. N = 100 ou 200; on ne peut pas attendre un temps infini). On incrémente in compteur de ruines de +1 chaque fois qu'on trouve que le joueur A_m du m-ième scénario (m ∈ {1...M}) s'est ruiné sur l'intervalle de temps [0, N].

On approche alors la probabilité de ruine par

$f_S(k) = \mathbb{P}(\mathrm{R}_A|\mathrm{X}_0 = \mathrm{k}) = \mathrm{E}(\mathbb{1}_{R_A}|\mathrm{X}_0 = k)$

$f_S^{approx}(\mathrm{k}) := \frac{1}{M}\sum_{m=1}^{M}(1_{R_{A_m}}|\mathrm{X}_{m,0} = k)$.

Toute approximation de ce type c-à-d. toute approximation d'une espérance par la Loi des Grands Nombres, est appelée *approximation Monte-Carlo.*

On fixe $S = 20$, $k = 10$ et on étudie les 2 cas suivants :

- Jeu équilibré : $p = q = 0.5$
- $p = 0.45$, $q = 0.55$ (léger avantage pour le casino, ce qui est autorisé par la loi de la majorité des pays).

N.B :

- Les notes mises entre parenthèses sont des commentaires explicatifs qui ne font pas partie des instructions du code matlab.
- Le code de la probabilité de ruine et celui du temps moyen de jeu se basent sur le même principe. Donc, on les a mis ensemble pour éviter la répétition.

On implémente le code suivant :

```
*******
S=20;k=10;
fS=0;hS=0;
p=0.45;q=1-p; (p est la probabilité de gagner 1 dinar. A choisir par l'utilisateur)
N=200;M=400; (N est le temps, M est le nombre des joueurs : à choisir par l'utilisateur)

J=1 :M; (vecteur joueurs)
temps=0 :N-1; (vecteur temps)
R=0; (nombre de ruines initialisé à 0)
T=zeros(1,M); (vecteur temps de jeu)
Tm=0; (temps moyen de jeu initialisé à 0)

for j=1 :M (boucle sur les joueurs)
u=rand(1,N);
i=2;
x=zeros(1,N); (somme initiale) x(1)=k;

while (i<=N)
if u(i)<p
x(i)=x(i-1)+1; (le joueur gagne 1 dinar)

else
x(i)=x(i-1)-1; ( sinon il perd 1 dinar)

end

T(j)=T(j)+1; (temps de jeu du joueur j est incrémenté)

if (x(i)==0)
R=R+1;
for l=i+1 :N
x(l)=0;
end
break;
end

if (x(i)==S)
```

for l=i+1 :N
x(l)=S ;
end
break ;
end
i=i+1 ;

end

plot(temps,x) ;title('simulation des joueurs') ;
xlabel('temps') ;ylabel('somme du joueur ') ;
hold on
Tm=Tm+T(j)/M ; (Temps moyen de jeu en cours d'implémentation)
end
R=R/M ; (=$f_S^{approx}(k)$)

(Tm= $h_S^{approx}(k)$)

figure(2) ;plot(J,T) ;title('simulation du temps de jeu des joueurs') ;
xlabel('joueurs') ;ylabel('temps de jeu') ;

(calcul théorique de $f_S(k) et \mathrm{h}_S(k)$)
Pièce équilibrée :)
if (p==0.5)
fS=1-k/S ;
hS=k*(S-k) ;
end

(pièce non équilibrée :)
if (p>0.5)||(p<0.5)
fS=[(q/p)$^k - (q/p)^S]/[1-(q/p)^S]$;
$hS = 1/(q-p)*(k-S*(1-(q/p)^k)/(1-(q/p)^S))$;
end
$*********$

<u>*Jeu équilibré*</u> : $p = q$ =0.5
Pour M=200 et N=1000, on obtient après compilation : R = 0.449.

$f_{20}(10)$ = 1−k/S= $1-10/20 = 0.5$. Donc f_{20}^{approx} ≈$f_{20}(10)$.
$f_{20}(10) = 0.5$, donc dans le cas d'équiprobabilité, le joueur a la même chance de gagner que de perdre.

Jeu truqué : p = 0.45 et q = 0.55

M=200 et N=1000 :
Si on fait la simulation de la variation de la somme restante de M joueurs au cours du jeu, on obtient la figure ci-dessous :

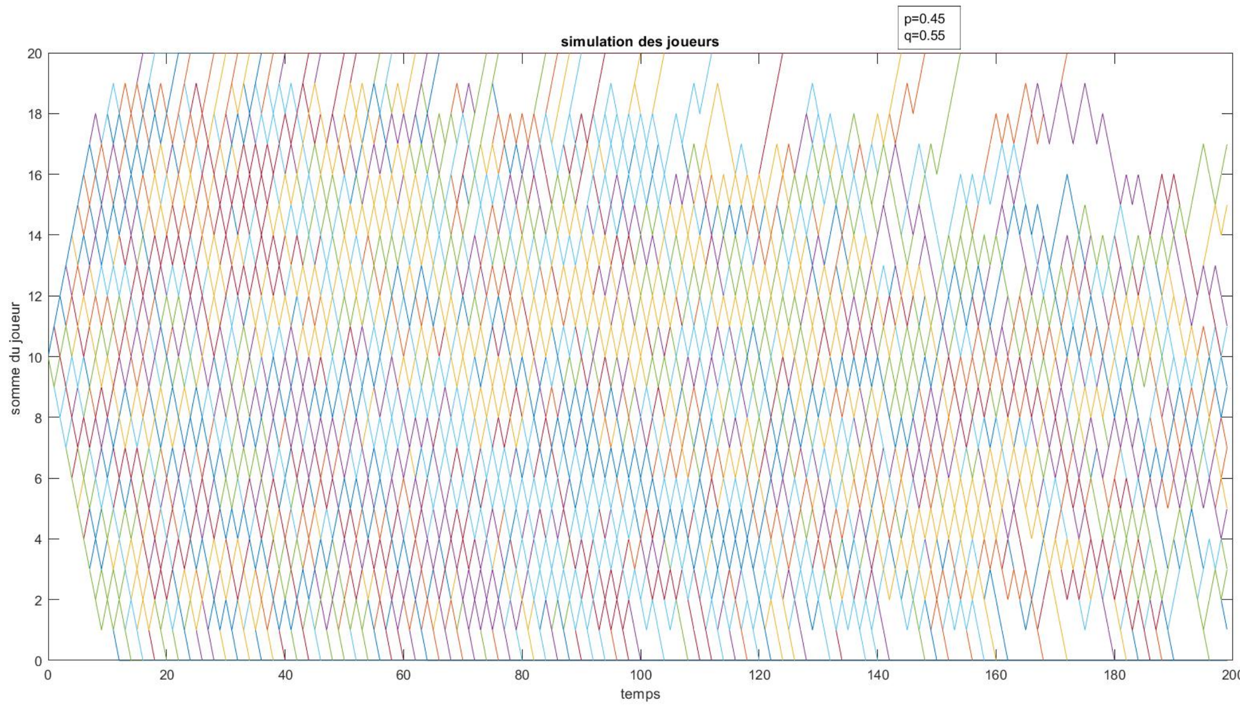

FIGURE 2.1: Simulation du jeu

Après implémentation du code, on obtient : R = 0.835.
$f_{20}(10) = \frac{(q/p)^k-(q/p)^S}{1-(q/p)^S} = 1.016$. Les deux valeurs sont proches. En plus, on remarque que $f_{20}(10)$ est presque égale à 1 malgré que le casino avait un léger avantage de gagner. C'est ce qui explique la richesse des casinos.

On dispose toujours d'une somme $S = 20$ dinars. On fait varier $X_0 = k$ de 0 à 20 dinars, et on trace simultanément $f_{20}^{approx}(k)$ et $f_{20}(k)$ dans chacun des deux cas précédents.
Voici le code tapé :
****

r=zeros(1,21) ; (vecteur proba de ruines initialisé)

t=zeros(1,21) ; (vecteur temps moyen de jeu initialisé)

K=0 :20 ; (vecteur des k)

```
for k=0 :20
temps=0 :N-1 ; (vecteur temps)
R=0 ; (nombre de ruines initialisé)
Tm=0 ;
for j=1 :M
u=rand(1,N) ;
i=2 ;
x=zeros(1,N) ; ( somme initiale)
x(1)=k ;
while (i<=N)
if u(i)<p
x(i)=x(i-1)+1 ;

else
x(i)=x(i-1)-1 ;

end
t(k+1)=t(k+1)+1 ;

if (x(i)==0)
R=R+1 ;
for l=i+1 :N
x(l)=0 ;
end
break ;
end
if (x(i)==S)
for l=i+1 :N
x(l)=S ;
end
break ;
end
i=i+1 ;

end

end
t(k+1)=t(k+1)/M ; (temps de jeu moyen pour la valeur k)
```

R=R/M; (proba de ruine pour k)
r(k+1)=R;

end

(les vecteurs h20approx(k) et f20approx(k) sont construits)
(construisons les vecteurs h20(k) et f20(k))
rt=zeros(1,21); (vecteur f20(k) initialisé)
tt=zeros(1,21); (vecteur h20(k) initialisé)
if (p==0.5)
for k=0 :20
rt(k+1)=1-k/20;
tt(k+1)=k*(20-k);
end
end

if (p>0.5) || (p<0.5)
for k=0 :20
rt(k+1)=((q/p)$^k - (q/p)^S)/(1-(q/p)^S)$;
$tt(k+1) = 1/(q-p) * (k-20*(1-(q/p)^k)/(1-(q/p)^S))$;
end
end

(ainsi les vecteurs h20(k),h20approx(20),f20(k) et f20(k)approx sont construits)

(traçage de f20(k) et f20approx(k))
figure(3) ;plot(K,r,'r') ;title('simulation de f20(k) et f20approx(k)') ;hold on
xlabel('k') ;ylabel('proba de ruine');

plot(K,rt,'b') ;legend('f20approx(k)','f20(k)') ;

(traçage de h20(k) et h20approx(k))
figure(4) ;plot(K,t,'r') ;title('simulation de h20(k) et h20approx(k)') ;hold on
xlabel('k') ;ylabel('temps de jeu moyen') ;

plot(K,tt,'b') ;legend('h20approx(k)','h20(k)') ;

Jeu équilibré : $p = q = 0.5$
D'après les deux courbes, les valeurs de $f_{20}^{approx}(k)$ et $f_{20}(k)$ sont proches $\forall k \in \{0, \ldots, 20\}$.

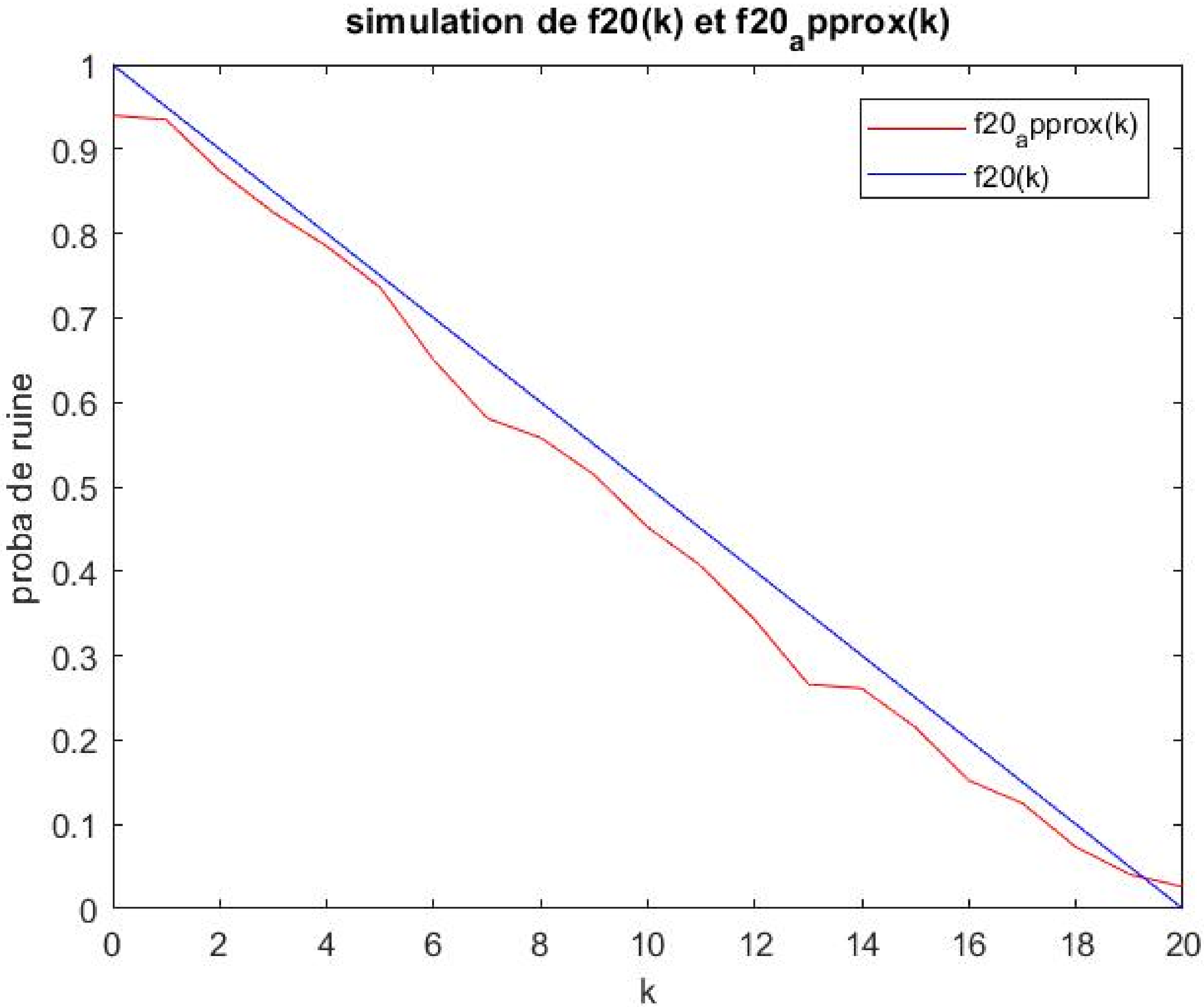

FIGURE 2.2: $f_{20}(k)$ et $f_{20}^{approx}(k)$

Jeu truqué : p = 0.45 et q = 0.55

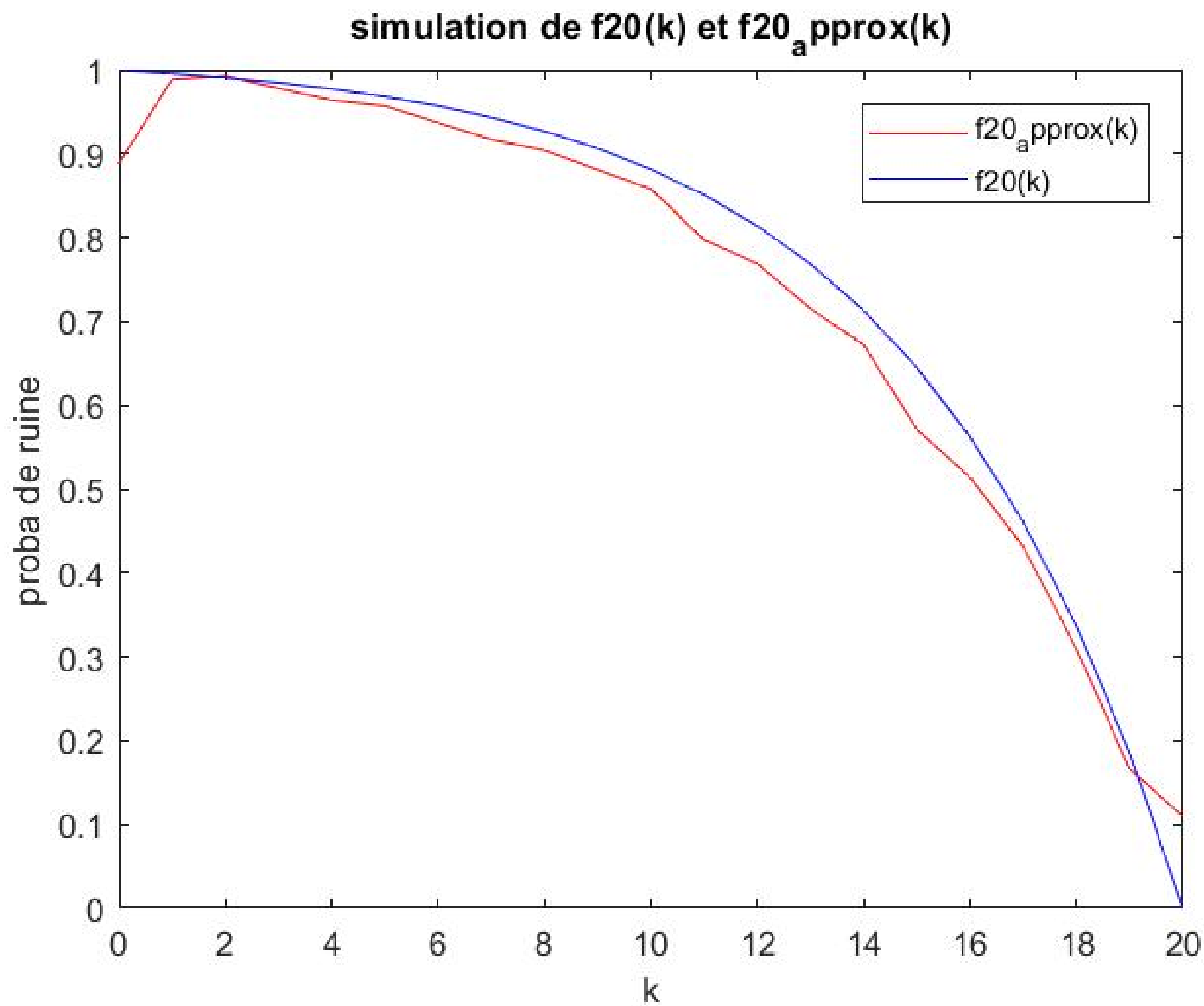

FIGURE 2.3: $f_{20}(k)$ et $f_{20}^{approx}(k)$

D'après les deux courbes :

— les valeurs de $f_{20}^{approx}(k)$ et $f_{20}(k)$ sont proches $\forall k \in \{0, \dots, 20\}$.

— La courbe $f_{20}^{approx}(k)$ croit quand k passe de 0 à 1, puis à partir de $k = 2$, les deux courbes décroissent jusqu'à 0 (pour $f_{20}(k)$) et 0.09 (pour $f_{20}^{approx}(k)$).

— Quand la somme initiale k augmente, le joueur a plus de chance de rester en jeu.

Maintenant, pour $S = 20$ et $k = 10$ fixés, on fait varier $p \in]0,1[$ et on trace les courbes de $f_{20}(10)$ et de $f_{20}^{approx}(10)$ correspondantes. Voici le code :

****

r=[] ; (vecteur f20(10))

rt=[] ; (vecteur f20approx(10))

t=[] ; (vecteur h20(10))

tt=[] ; (vecteur h20approx(10))

P=0 :0.1 :1 ;

p=0 ;q=1-p ;

c=1 ;

while (p<=1)

if (p<0.5 || p>0.5)

(calcul théorique de f20(10) et h20(10))

rt(c)=((q/p)$^{10} - (q/p)^{20})/(1-(q/p)^{20})$;

$tt(c) = 1/(q-p) * (10 - 20 * (1-(q/p)^{10})/(1-(q/p)^{20}))$;

(calcul numerique de f20approx(10) et h20approx(10))

R=0 ; (nombre de ruines initialisé)

tm=0 ; (temps moyen total initialisé)

for j=1 :M (boucle sur les joueurs)

u=rand(1,N) ;

i=2 ;

x=zeros(1,N) ; (somme initiale)

x(1)=10 ;

tmj=0 ; (temps moyen du joueur j)

while (i<=N)

if u(i)<p

x(i)=x(i-1)+1 ; (le joueur gagne 1 dinar)

else

x(i)=x(i-1)-1 ; (sinon il perd 1 dinar)

```
end
tmj=tmj+1;

if (x(i)==0)
R=R+1;

break;
end

if (x(i)==20)

break;
end
i=i+1;

end

tm=tm+tmj/M; (tm=h20approx(10) pour la proba p)
end

R=R/M; ( =f20approx(10) pour la proba p)
r(c)=R;
t(c)=tm;
c=c+1;
p=p+0.1;
end

if p==0.5
( calcul theorique de f20(10) et h20(10) )
rt(c)=1-10/20;
tt(c)=10*(20-10);

(calcul de f20approx(10) et h20approx(10) )
R=0; (nombre de ruines initialisé )

tm=0; (temps moyen total )
```

```
for j=1 :M (boucle sur les joueurs )
u=rand(1,N) ;
i=2 ;
x=zeros(1,N) ; (somme initiale)
x(1)=10 ;
tmj=0 ; (temps moyen du joueur j )
while (i<=N)
if u(i)<p
x(i)=x(i-1)+1 ; ( le joueur gagne 1 dinar )

else
x(i)=x(i-1)-1 ; ( sinon il perd 1 dinar)

end

tmj=tmj+1 ;

if (x(i)==0)
R=R+1 ;

break;
end

if (x(i)==S)

break;
end
i=i+1 ;

end
tm=tm+tmj/M; ( =h20approx(10) pour la proba p )

end

R=R/M; ( =f20approx(10) pour la proba p )
r(c)=R;
t(c)=tm;
```

c=c+1 ;

p=p+0.1 ;

end

end

figure(5) ;plot(P,rt,'b') ;hold on
plot(P,r,'r') ;title('simulation de f20(10) et f20approx(10)') ;
legend('f20(10)','f20approx(10)') ;
xlabel('proba p') ;ylabel('proba de ruine') ;

figure(6) ;plot(P,tt,'b') ;hold on
plot(P,t,'r') ;title('simulation de h20(10) et h20approx(10)') ;
legend('h20(10)','h20approx(10)') ;
xlabel('proba p') ;ylabel('temps moyen du jeu') ;

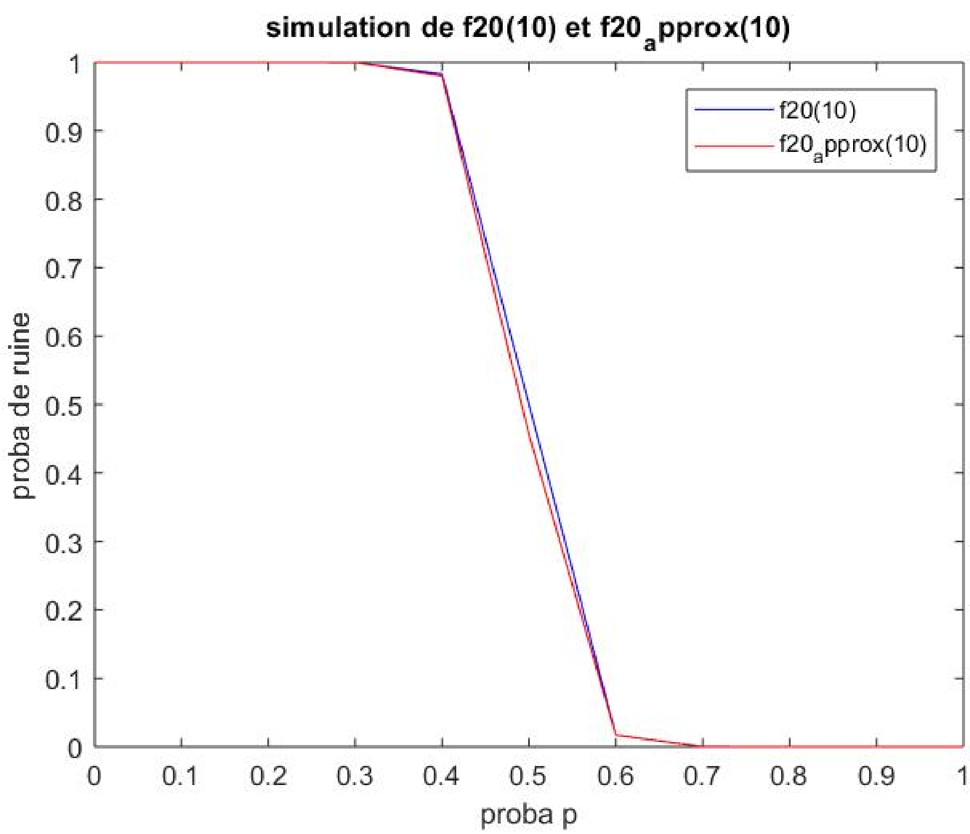

FIGURE 2.4: $f_{20}(10)$ et $f_{20}^{approx}(10)$ en fonction de p

— Pour $p \in]0, 0.4[$, le joueur est ruiné : évènement presque sûr.

— Pour $p \in [0.4, 0.6[$, la restriction de chaque courbe sur cet intervalle estt une droite décroissante qui décroit rapidement. C'est logique car plus p augmente plus la probabilité de ruine du joueur diminue.

— Pour $p \in [0.6, 1[$, les deux courbes sont presque constantes à 0. Ce qui parait logique car plus la probabilité p augmente plus le joueur a de chance de rester en jeu.

2.2 APPROXIMER LA DURÉE MOYENNE DU JEU

La durée moyenne du jeu lorsque A part d'une fortune initiale $X_0 = k$ est définie par :
$\mathrm{h}_S(k) := \mathbb{E}[T_{0,S}|X_0 = \mathrm{k}]$ avec $\mathrm{T}_{0,S} := inf\{\mathrm{n} \in \mathbb{N} t.q \mathrm{X}_n = 0 ou X_n = \mathrm{S}\}$. Donc l'idée, comme indiqué dans le code précédent de la partie 3.1, est d'incrémenter un compteur de temps +1 à chaque fois que le joueur joue et on arrête le compteur quand le joueur est ruiné. Et on somme les temps du jeu de tous les joueurs, puis on divise la somme par M pour obtenir enfin le temps moyen du jeu.
Pour N = 200 et M = 1000, on implémente le code qui donne la valeur de $h_{20}^{approx}(10)$.

<u>*Jeu équilibré*</u> : $p = q = 0.5$
Ainsi, on obtient : $h_{20}^{approx}(10) = 93.668$ une valeur proche de la valeur théorique donnée par la formule suivante : $\mathrm{h}_{20}(10) := \mathbb{E}[T_{0,S}|X_0 = \mathrm{k}] = \mathrm{k}(\mathrm{S}-\mathrm{k}) = 10.(20-10) = 100$.

<u>*Jeu truqué*</u> :$\mathrm{p}= 0.45 et \mathrm{q}= 0.55$
On obtient $h_{20}^{approx}(10) = 74.742$ une valeur proche de la valeur théorique donnée par la formule suivante : $\mathrm{h}_{20}(10) := \mathbb{E}[T_{0,S}|X_0 = \mathrm{k}] = \frac{1}{q-p}(k - S\frac{1-(q/p)^k}{1-(q/p)^S}) = 76.299$
Plus la probabilité de gagner 1 dinar diminue plus la probabilité de ruine augmente, donc plus le temps moyen du jeu diminue.

Maintenant, pour $S = 20$ dinars, on fait varier $\mathrm{X}_0 = k$ de 0 à 20 dinars et on trace $\mathrm{h}_{20}^{approx}(\mathrm{k}) et h_{20}(k)$ en fonction de k dans chacun des deux cas précédents.

Jeu équilibré : $p = q = 0.5$

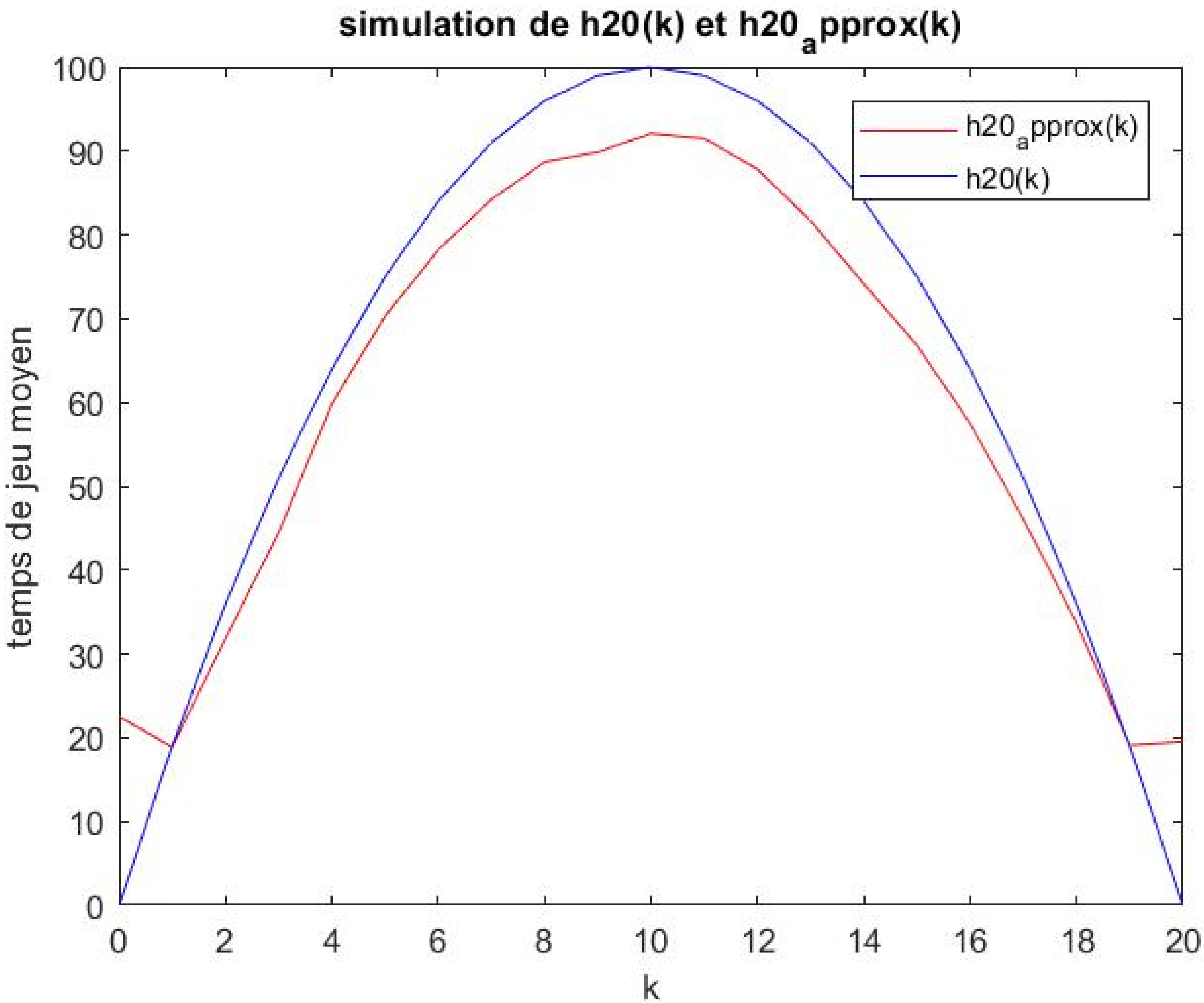

FIGURE 2.5: $h_{20}(k)$ et $h_{20}^{approx}(k)$

On remarque bien que :

— Les deux courbes ont preque la même allure.

— Pour k $\in \{0, \ldots, 10\}$ le temps moyen du jeu augmente vu que la somme initiale augmente.

— Pour k $\in \{11, \ldots, 20\}$ le temps moyen du jeu diminue car plus la somme initiale k du joueur s'approche de la somme S totale plus le joueur a de chance de rester dans le jeu.

Jeu truqué : p = 0.45 et q =0.55

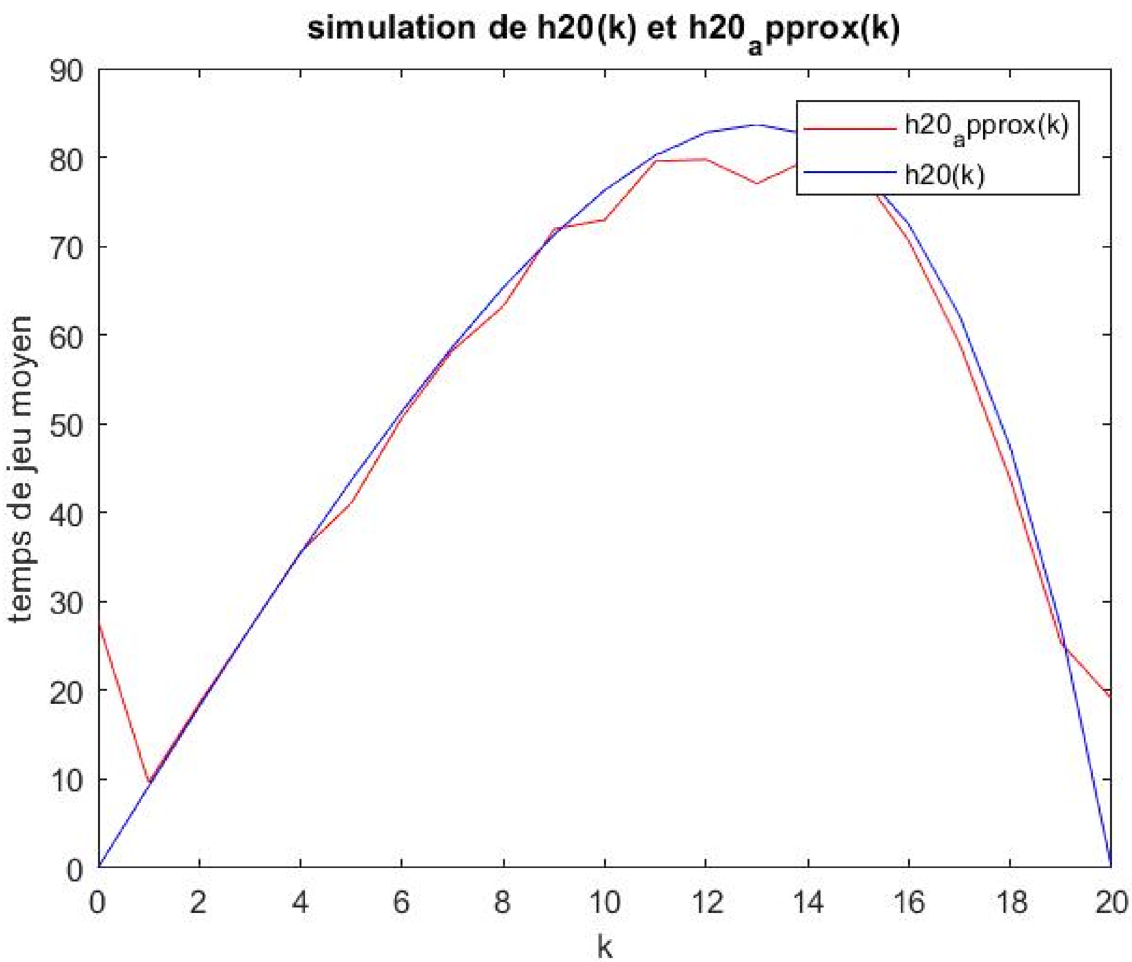

FIGURE 2.6: $h_{20}(k)$ et $h_{20}^{approx}(k)$

On remarque bien que :

- Les deux courbes ont presque la même allure.
- La vitesse d'augmentation du temps moyen du jeu en fonction de k est inférieure à celle observée pour $p = q = 0.5$ vu que le casino avait du départ un avantage. Donc il faut une somme initiale k supérieure afin que le joueur puisse augmenter sa chance de rester dans le jeu.

Maintenant, on fait varier $p \in]0,1[$ pour $S = 20$ et $k = 10$ fixés et on trace les courbes de $\mathrm{h}_{20}(10) et h_{20}^{approx}(10)$ correspondantes.

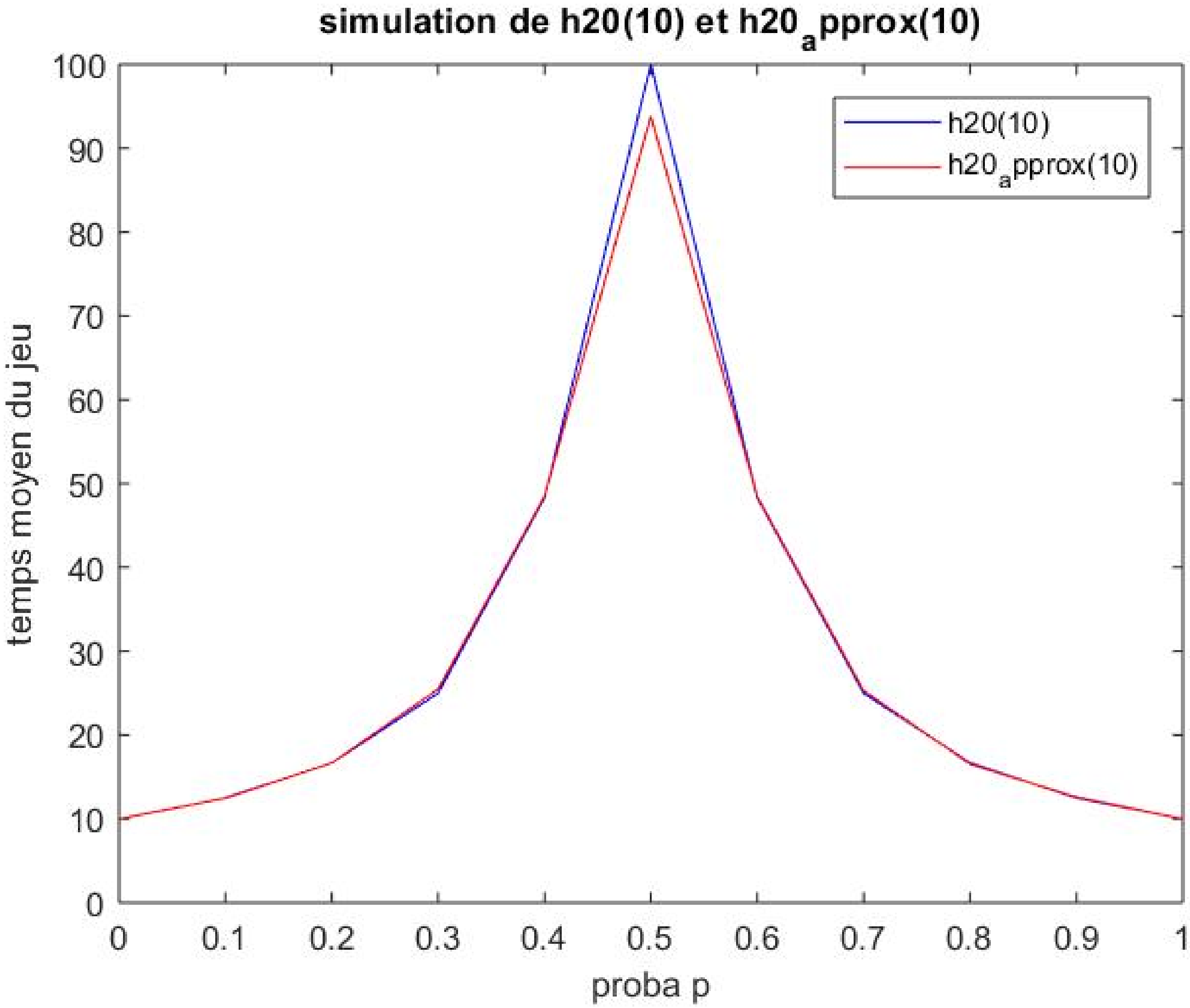

FIGURE 2.7: $h_{20}(10)$ et $h_{20}^{approx}(10)$ en fonction de p

Les deux courbes ont la même allure.

- Pour $p \in [0, 0.5]$: le temps moyen du jeu augmente, ce qui est logique car plus p s'approche de 0.5 plus la partie devient "serrée".
- Pour $p \in [0.5, 1]$: le temps moyen du jeu diminue tant que p s'approche de 1. Ce qui est logique car plus s'approche de 1 plus la chance du joueur de gagner la partie augmente, donc le temps moyen du jeu diminue.

Chapitre 3

Partie personnelle

Dans cette partie, on a choisi d'étudier un cas particulier de marches aléatoires qui est le *mouvement brownien*. Ce mouvement a été remarqué pour la première fois par le botaniste Robert Brown lors de son observation du mouvement des particules des grains de pollen. Donc, tout d'abord on va définir ce mouvement et explorer le modèle élaboré par Albert Einstein. Puis, on va citer des exemples et découvrir ses applications.

3.1 MODÈLE PHYSIQUE D'EINSTEIN

Le mouvement brownien est une description mathématique du mouvement d'une "grosse" particule entourée de "petites" particules qui s'agitent au cours du temps. En plus, la "grosse" particule n'est soumise qu'à des chocs avec les "petites" particules immergées dans le fluide environnant.
Exemple : Prenons un mélange de lait et de l'eau et on en observe un échantillon par un microscope. L'idée est d'avoir le schéma ci-dessous :

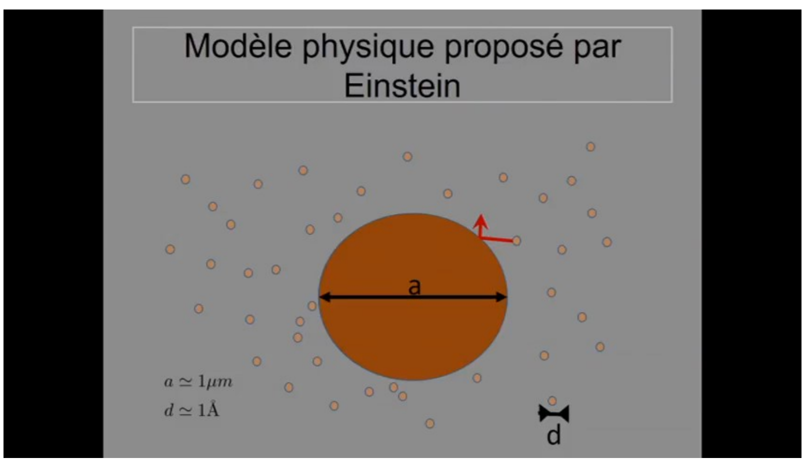

FIGURE 3.1: Observation d'un échantillon

L'agitation thermique des petites particules qui cognent la grosse particule engendre un mouvement aléatoire de celle-ci. Le fait de ne pas savoir où la grosse particule va se déplacer complique un peu les choses. C'est pourquoi on a eu recours à étudier les propriétés statistiques du mouvement brownien. D'ailleurs, en observant le mouvement de la grosse particule, on remarque bien qu'elle reste en moyenne à la même place. Mais quand même elle explore une certaine zone. Et on s'intéresse en fait à la taille de cette zone à propos de laquelle Einstein a énoncé une formule importante.

Si on onsidère le mouvement de la grosse particule entre deux instants t et s (t>s), on va trouver que ce mouvement suit une loi gaussienne de moyenne 0 et de variance (t-s). A ce propos, Einstein a énoncé la formule suivante :

$\sum^2(t) =< x(t)^2 >= \frac{RT}{n}\frac{1}{3\pi a v}t$

Alors, on voit bien que cette variance croit avec le temps et qu'elle dépend d'un certain nombre de coefficients :

— R : constante des gaz parfaits ≈8.3 J.K/mol
— T : la température en Kelvin
— a : rayon de la grosse particule qui peut être déterminé par un microscope.
— v : viscosité du fluide

Et on remarque aussi le N. C'est le célèvre nombre d'Avogadro qui est, par définition, le nombre d'atomes de carbone contenus dans 12g de carbone12. C'est environ le nombre de nucléons contenus dans 1g de nucléons.

Cette formule alors nous permet de peser un nucléon. C'est impressionnant car c'est à grâce à la variance du mouvement brownien(quantité mésoscopique) qu'on a pu avoir une information sur la masse d'un nucléon qui est une particule vraiment microscopique et qu'on ne peut pas observer avec un microscope.

3.2 APPLICATIONS ET EXEMPLES

Le mouvement brownien reste encore une thématique de recherche active vu ses diverses applications :

— Propriétés mathématiques : propriétés d'enroulement, aire couverte, statistiques extrêmes...
— Applications physiques : mouvement des molécules d'un gaz, colloîdes,...
— Application aux sciences sociales : théorie des jeux,...

Maintenant, on va citer quelques exemples pour mieux expliquer le mouvement brownien. On commence par un exemple simple.

Exemple 1 : Marches aléatoires

Il s'agit d'un mouvement brownien unidimentionnel à 1000 pas. On simule une particule qui part de la position 0. En fait, le mouvement brownien représente un cas particulier de marches aléatoires (pour p

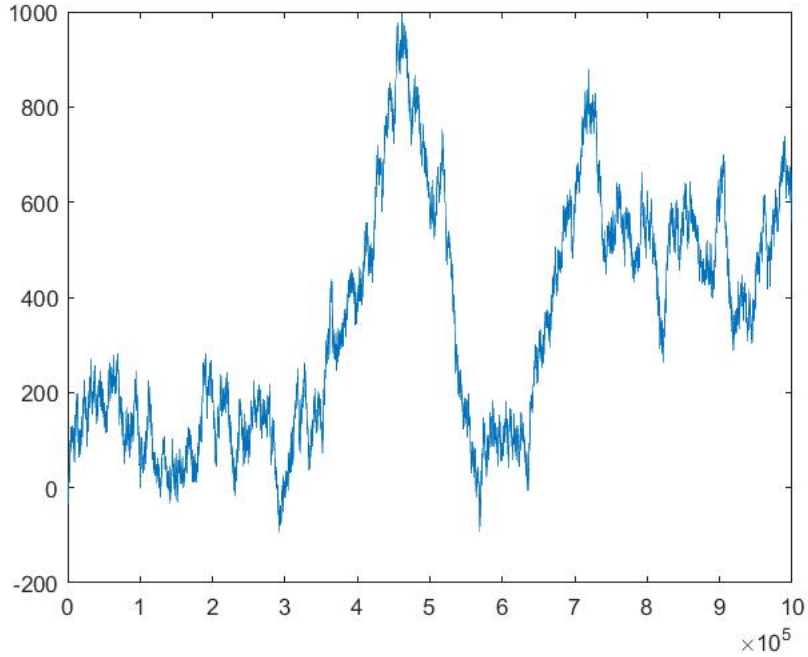

FIGURE 3.2: Mouvement brownien 1-D à 1000 pas

$= q = 1/2$) car on ne sait pas à un instant donné où la particule va se déplacer : *Isotropie spatiale.* Tous les déplacements sont décorrélés, c-à-d l'état du mouvement à chaque instant est indépendant de son passé même le passé le plus proche.

Exemple 2 : Mouvement bidimentionnel

Il s'agit d'un mouvement brownien 2D à 3000 pas. Chaque pas est gaussien en ordonnéee et en abscisse.

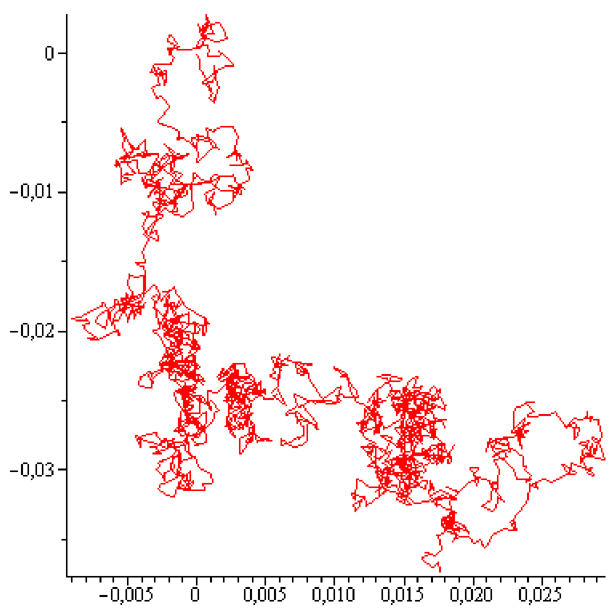

FIGURE 3.3: Mouvement brownien 1-D à 1000 pas

Maitenant, on va examiner un mouvement brownien 3D.

Exemple3 : Mouvement 3D

Il s'agit d'implémenter un code matlab qui permet de :

- Décrire la trajectoire d'une particule.
- Calculer la distance séparant le point de départ et le point d'arrivée.
- Calculer le temps mis par la particule pour passer pour la première fois à l'intérieur d'une sphère fixée au centre du repère.

Dans la première partie, on va simuler un mouvement brownien à 1000 pas et calculer ensuite la distance séparant les deux points d'arrivée et de départ.

Voici le code implémenté :

****

(pour dessiner une sphère de centre (0 0 0) et de rayon r)

r=2 ;
x y z

= sphere ;

surf(r*x,r*y,r*z,'facecolor','m') ; (la sphère est mauve)

(tracer le chemin d'une particule)

N=1000 ; (le temps)

(pour le mouvement brownien, la différence x(i+1)-x(i) suit la loi normale de même pour y et z)

x=cumsum(randn(1,N)) ;

y=cumsum(randn(1,N)) ;

z=cumsum(randn(1,N)) ;

(la boucle suivante sert à connaitre l'instant de la première atteinte de la sphère par la particule)

Tm=0 ;

for i=1 :N

if x(i)$^2 + y(i)^2 + z(i)^2 <= r^2$

$Tm = i;$

break

end

end

```
(instructions pour l’affichage)
view(3)
set(gca,’GridLineStyle’,’–’)
hold on;
plot3(x,y,z,’LineWidth’,1.5)
axis([min(x) max(x) min(y) max(y) min(z) max(z)]);
xlabel(’x’);
ylabel(’y’);
zlabel(’z’);

(position initiale)
r0=[x(1) y(1) z(1)];
( position finale)
rf=[x(N) y(N) z(N)];

(on distingue les points de départ et d’arrivée par la couleur)
plot3(r0(1),r0(2),r0(3),’or’,’MarkerFaceColor’,’r’); hold on ( rouge )
plot3(rf(1),rf(2),rf(3),’ok’,’MarkerFaceColor’,’k’); hold on ( noir )

grid on;

( Distance entre le point de départ et le point de fin de la trajectoire )
d=sqrt(sum((rf-r0).^2));

( titre qui affiche la distance calculée précédemment )
Information=strcat(’Un mouvement brownien 3D, d=’,num2str(d),’ units’);
title(Information ,’FontWeight’,’bold’);
view(120,60);
**********
```

Le résultat de la simulation est représenté dans la figure suivante :

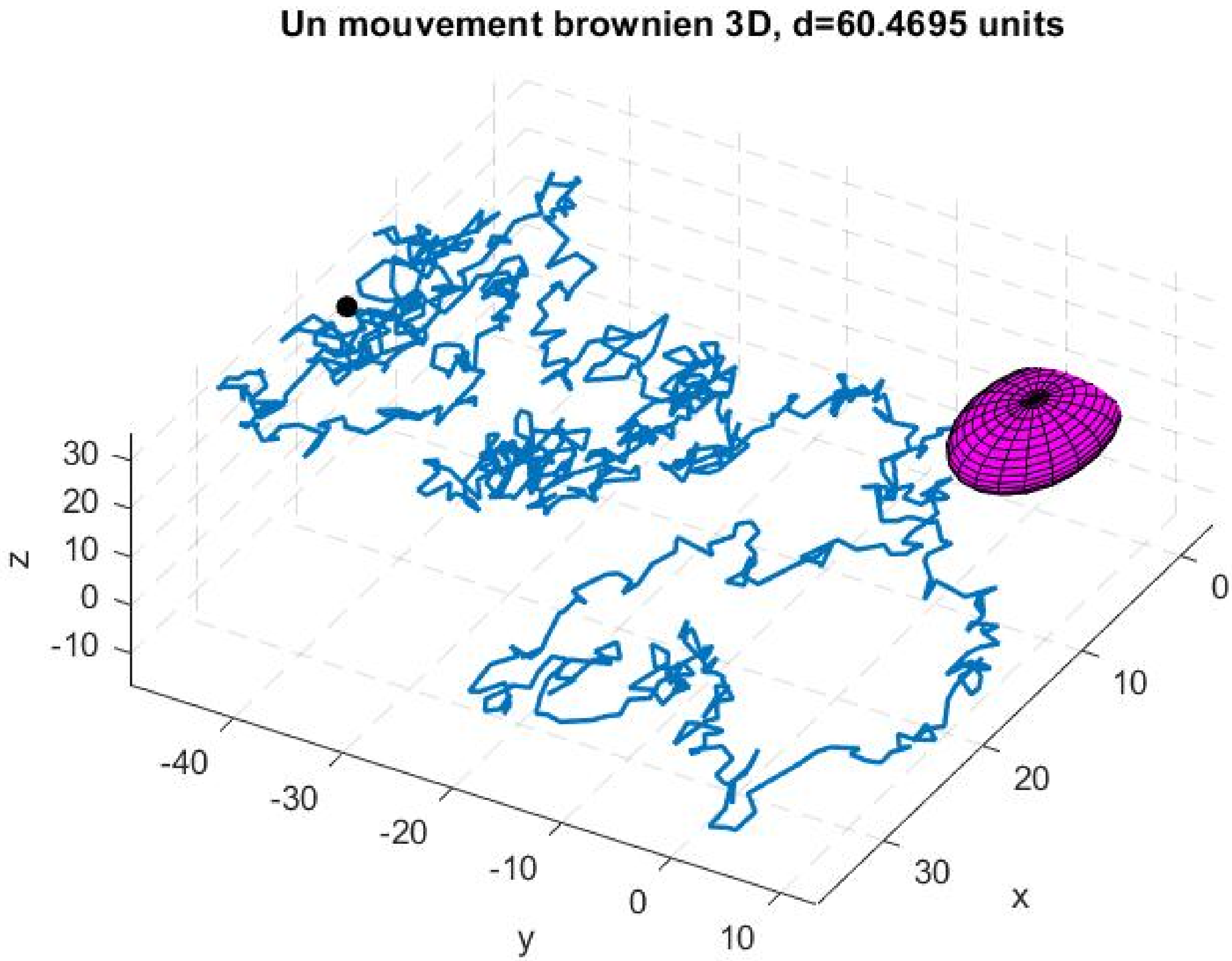

FIGURE 3.4: Mouvement brownien 3-D à 1000 pas

Pour cet exemple 3-D, la distance séparant les deux points de départ et d'arrivée est d = 60.4695 unités

Dans la deuxième partie, on va déterminer le temps moyen d'atteinte d'une sphère par un échantillon de particules et pour différentes valeurs du rayon de la sphère et tracer le résultat de l'implémentation. Voici le code :

****

P=100 ; (le nombre de particules)

T=0 ; (temps moyen initialisé)

r=0.1 ; (rayon initialisé)

c=1 ; (indice)

TM=[] ; (vecteur temps moyen)

while (r<=1)

for n=1 :P

(construction de la sphère)

```
[x y z]=sphere ;
surf(r*x,r*y,r*z) ;

N=1000 ; ( temps de la trajectoire )

x=cumsum(randn(1,N)) ;
y=cumsum(randn(1,N)) ;
z=cumsum(randn(1,N)) ;

for i=1 :N
```
if x(i)$^2 + y(i)^2 + z(i)^2 <= r^2$

$T = T + i/P$;

break

end

end

```
end
TM(c)=T ;
r=r+0.1 ;
c=c+1 ;
end
VR=0.1 :0.1 :1 ;figure(2) ;plot(VR,TM) ;
```

****

Ainsi, on obtient le tracé suivant :

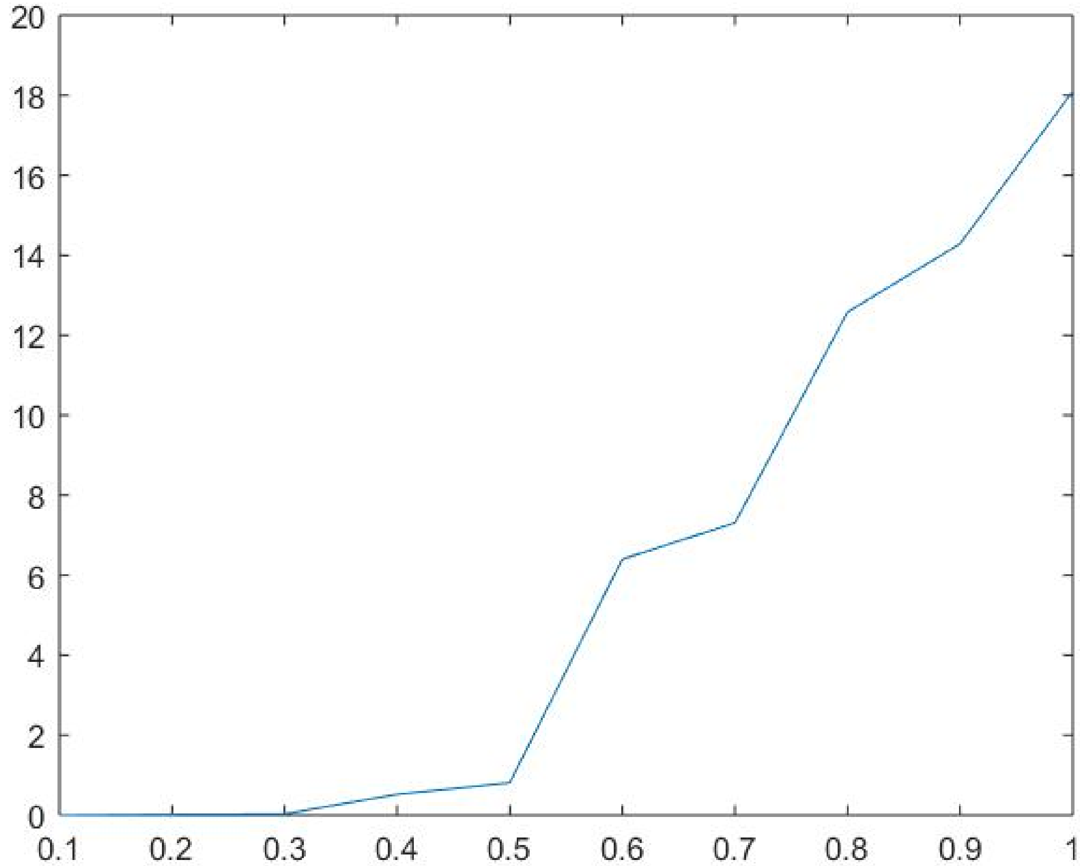

FIGURE 3.5: Mouvement brownien 3-D à 1000 pas

On remarque que plus le rayon de la sphère augmente plus le temps moyen pour atteindre la sphère(être à la surface ou à l'intérieur de la sphère) augmente aussi.

Conclusion :

"La ruine du joueur" est un problème de marches aléatoires. On s'est focalisé sur son étude numérique à partir de laquelle on a simulé la probabilité de ruine d'un joueur de casino qu'on a comparée ensuite avec la probabilté théorique. On s'est mis d'accord que les deux formules donnent des résultats à peu près similaires. De même, on a déterminé une formule d'approximation du temps moyen du jeu qu'on a comparée avec la formule théorique. Et on a trouvé le même résultat. Enfin, on peut juger le jeu du casino comme un jeu "mathématiquement truqué" car c'est le casino qui a le plus souvent l'avantage. Ceci explique évidemment la richesse des casinos.
Personnellement, ce mini-projet m'a profondément aidé à comprendre mieux les marches aléatoires et leur domaine d'application et à me familiariser avec Matlab.