

Méthodes probabilistes pour la simulation

par Rémi PEYRE

printemps 2012



Ce polycopié est mis à disposition selon les termes de la licence publique Créative Commons avec paternité et partage à l'identique (CC BY-SA), version 3.0 transposée en droit français : voir le texte de la licence sur <http://creativecommons.org/licenses/by-sa/3.0/fr/legalcode>.

En résumé, vous êtes libres de reproduire, distribuer et communiquer ce polycopié, ainsi que de le modifier, transformer ou adapter, y compris à des fins commerciales, selon les conditions suivantes :

- Vous devez mentionner l'auteur du présent document (mais pas d'une manière qui suggérerait qu'il vous soutienne ou approuve votre utilisation de celui-ci) ;*
- Si vous modifiez, transformez ou adaptez ce document, vous n'avez le droit de distribuer votre création que sous une licence identique ou similaire à celle-ci.*

Table des matières

La méthode de Monte-Carlo	5
I Principes	6
I.1 Introduction	6
I.2 Base de la méthode	7
I.2.a Exemple introductif : le problème du yahtzee	7
I.2.b Aspects théoriques	7
I.2.c Méthode générale de calcul d'une espérance	8
I.2.d Illustration : la course à pied	8
I.3 Intervalles de confiance	8
I.3.a Problématique	8
I.3.b Intervalles de confiance gaussiens	9
I.3.c Exemple d'application	11
I.3.d Annexe : table des intervalles de confiance gaussiens	12
I.4 Méthode de Monte-Carlo pour le calcul d'une intégrale	13
I.4.a Fonctions à support borné	13
I.4.b Cas général	14
I.4.c Comparaison avec les méthodes d'intégration déterministes	16
I.5 Choix de l'échantillonnage	18
I.5.a Notion d'efficacité	18
I.5.b Échantillonnage idéal	19
I.5.c Principe d'échantillonnage préférentiel	20
I.5.d Exemples	21
I.5.e Manipulations préliminaires à l'échantillonnage préférentiel	25
I.5.f Échantillonnage interstrates	26
I.5.g Échantillonnage idéal généralisé	29
II Techniques de réduction de la variance	30
II.1 Généralités	30
II.2 Conditionnement	30
II.3 Variables antithétiques	32
II.3.a Exemple : dilemme rugbystique	32
II.3.b Approche générale	34
II.4 Variables de contrôle	35
II.4.a Variable de contrôle unique	35
II.4.b Variables de contrôle multiples	36
II.4.c Exemple	37

II.5	Stratification	38
II.5.a	Approche théorique	38
II.5.b	La stratification en pratique	40
II.6	Optimisation du code	41
II.7	Synthèse	42
III	Annexe : Codes <i>MATLAB</i>	43
III.1	Les codes pour le problème du yahtzee	43
III.1.a	Algorithme « naïf »	43
III.1.b	Algorithme avec conditionnement	44
III.1.c	Algorithme avec échantillonnage préférentiel	46
III.2	Le code pour la course à pied	48
III.3	Codes pour les calculs de volume de boule	48
III.3.a	Méthode de Monte-Carlo sur le cube	48
III.3.b	Méthode de Monte-Carlo avec échantillonnage préférentiel	49
III.4	Les codes pour le calcul de l'intégrale cubique	50
III.4.a	Échantillonnage selon la loi <i>Cauchy</i> (1)	50
III.4.b	Échantillonnage double-exponentiel	51
III.4.c	Échantillonnage selon <i>Cauchy</i> (0,65)	51
III.4.d	Échantillonnage de type Pareto	52
III.5	Grandes déviations de 100 lancers d'une pièce	53
III.5.a	Algorithme « naïf »	53
III.5.b	Avec échantillonnage préférentiel	53
III.6	Code pour le dilemme rugbystique	54
	Application aux processus aléatoires	56
IV	Le mouvement brownien	57
IV.1	Approche intuitive du mouvement brownien	57
IV.2	Propriétés du mouvement brownien	59
IV.3	Simulation du mouvement brownien	64

Première partie

La méthode de Monte-Carlo

Chapitre I

Principes

I.1 Introduction

En mathématiques, on appelle *méthodes de Monte-Carlo* les techniques permettant d'évaluer une quantité *déterministe* à l'aide de l'utilisation de tirages *aléatoires*. C'est de cette idée de recours au hasard que vient la dénomination « Monte-Carlo », par allusion au célèbre quartier de Monaco réputé pour son casino...

Historiquement, c'est en 1949 que le physicien gréco-américain Nicholas METROPOLIS et le mathématicien américain d'origine polonaise Stanisław ULAM publient l'article fondateur de cette méthode de calcul et lui donnent son nom. Pour être plus exact, l'idée de procéder à des tirages aléatoires pour évaluer des intégrales compliquées était dans l'air du temps parmi la communauté des physiciens, mais l'apport majeur de Metropolis & Ulam fut de proposer la technique d'échantillonnage préférentiel (cf. § I.5.c), qui améliore largement l'efficacité de la méthode. Pour l'anecdote, c'est dans le cadre des recherches du projet *Manhattan* sur le développement de la bombe atomique que ces chercheurs (avec quelques autres dont notamment John VON NEUMANN) avaient commencé à développer leurs idées.

Des développements importants des méthodes de Monte-Carlo furent l'algorithme de Metropolis–Hastings pour la simulation de certaines variables aléatoires en physique statistique (travaux dus notamment à Marshall ROSENBLUTH en 1953 et à Keith HASTINGS en 1970), algorithme qui à son tour fut la base de la méthode du recuit simulé (1983) pour trouver des extrema globaux de fonctions définies sur des espaces de grande dimension. Plus récemment (2008), on a aussi parlé des méthodes de Monte-Carlo à l'occasion de leur utilisation dans des logiciels joueurs de go (très grossièrement, l'idée est que l'ordinateur évalue la qualité d'une position en imaginant que les joueurs terminent leur partie en jouant au hasard), où ces méthodes ont permis des progrès spectaculaires.^[*]

Dans ce cours, nous ne nous intéresserons qu'à la méthode de Monte-Carlo pour l'évaluation d'une intégrale (ou d'une espérance).

[*]. Avec notamment le logiciel *MoGo*, développé au sein de laboratoires de recherche français !

I.2 Base de la méthode

I.2.a Exemple introductif : le problème du yahtzee

Pour bien comprendre la méthode de Monte-Carlo, commençons par étudier un problème lié au jeu du *yahtzee*. Le *yahtzee* se joue avec cinq dés qu'on peut lancer trois fois : au premier lancer le joueur jette les cinq dés, puis aux deux lancers suivants il choisit librement quels dés il veut relancer.^[†] Le but du jeu est, à l'issue des trois lancers, d'obtenir la meilleure combinaison possible.

La combinaison de plus haute valeur, également appelée *yahtzee*, consiste à ce que tous les dés affichent le même chiffre. On se pose la question suivante : quelle est la probabilité qu'un joueur se fixant pour unique objectif d'obtenir un *yahtzee* atteigne son but à l'issue de ses trois lancers ?

Il est clair que la meilleure stratégie à suivre pour les deuxième et troisième lancers, dans ce cadre, consiste simplement à regarder quel est le chiffre qui apparaît le plus souvent sur la table à l'issue du lancer précédent,^[‡] puis à choisir de relancer tous les dés n'affichant pas ce chiffre. En revanche, il n'est pas évident d'en déduire un calcul de probabilité pour la réussite finale, car cette réussite peut passer par un grand nombre de scénarios différents — en tout cas, les calculs sont particulièrement compliqués. Du coup, on se propose de *simuler* cette probabilité : on joue un très grand nombre de fois, et on s'attend à ce que la proportion de réussite tende asymptotiquement vers sa probabilité — puisque c'est là la définition « physique » de ce qu'est une probabilité ! Évidemment, il sera beaucoup plus rapide d'utiliser un ordinateur que de véritables dés ; c'est donc ainsi que nous allons procéder.

L'algorithme (informel) utilisé est donc le suivant, pour un nombre d'expériences N au choix de l'expérimentateur :

1. *Simuler un joueur effectuant ses trois lancers, avec la règle qu'il ne garde que les dés affichant le chiffre qui apparaît le plus souvent sur la table ; le faire N fois en tout ;*
2. *À chaque essai, regarder si le joueur a réussi à obtenir un yahtzee ; compter le nombre total de ses succès, qu'on appellera **succès** ;*
3. *Retourner la valeur numérique de **succès** / N .*

Le code correspondant est donné dans l'annexe III.1.a. Un essai avec $N = 1\,000\,000$ m'a donné 46 161 succès, soit une proportion empirique d'environ 4,616 %, ce qui est effectivement proche de la valeur théorique $100\,194\,336/6^{12} \simeq 4,602$ %.

I.2.b Aspects théoriques

L'argument théorique qui permet d'assurer le bon fonctionnement de l'algorithme est la *loi des grands nombres*. D'un point de vue abstrait, notre algorithme revient en effet à tirer N variables aléatoires i.i.d. X_1, \dots, X_N , telles que X_i vaut 1 si le joueur a réussi et 0 sinon, et à renvoyer $(X_1 +$

[†]. Il peut éventuellement choisir de n'en relancer aucun, ou au contraire de les relancer tous.

[‡]. Il se peut que plusieurs chiffres détiennent ce record *ex-æquo* ; dans ce cas, on désigne arbitrairement un de ces chiffres comme étant « le » chiffre qui apparaît le plus souvent.

$\dots + X_N) / N =: \hat{p}_N$ comme estimation de $\mathbb{P}(X = 1) =: p$. Or la loi des grands nombres nous dit précisément que quand N tend vers l'infini, la moyenne empirique que renvoie l'algorithme tend en probabilité (et même presque-sûrement) vers $\mathbb{E}(X) = p$. En d'autres termes, \hat{p}_N est un estimateur (fortement) convergent de p .

I.2.c Méthode générale de calcul d'une espérance

Dans la section précédente, nous avons utilisé une méthode de Monte-Carlo pour évaluer une probabilité. Mais en fait, tout ce que nous avons dit s'adapte immédiatement à l'évaluation d'une espérance ! Présentons cela sous la forme d'un théorème de synthèse :

8.a Théorème. *Soit X une variable aléatoire réelle L^1 et soient X_1, X_2, \dots des v.a. i.i.d. de même loi que X . Alors*

$$\hat{m}_N := N^{-1} \sum_{i=1}^N X_i \quad (1)$$

est un estimateur de $\mathbb{E}(X) =: m$ qui converge (fortement) vers sa cible.

I.2.d Illustration : la course à pied

Voyons maintenant un nouvel exemple concret. On imagine une course entre 8 athlètes dont les chronos sont modélisés par autant de variables aléatoires X_1, \dots, X_8 i.i.d. de loi $\ln\mathcal{N}(4; 0,03)$ (i.e. $\ln X$ suit la loi $\mathcal{N}(4; 0,03)$). Pour organiser efficacement la course, on a besoin de savoir quel temps, en moyenne, met le plus lent de coureurs, c.à.d. qu'on s'intéresse à $m := \mathbb{E}(\max_{1 \leq i \leq 8} X_i)$. Comme il n'y a pas d'expression analytique pour cette quantité, on recourt à la méthode de Monte-Carlo.

L'algorithme utilisé est alors le suivant (voir le code en annexe III.2) :

1. *Simuler les chronos des 8 coureurs et déterminer leur maximum ; le faire N fois en tout ;*
2. *Faire la somme des N chronos maximums obtenus, qu'on appellera *somme* ;*
3. *Retourner la valeur numérique de *somme* / N .*

Une simulation pour $N = 10^4$ m'a donné pour m l'estimateur 70,38, à comparer à la valeur exacte $\simeq 70,26$.

I.3 Intervalles de confiance

I.3.a Problématique

Grâce aux sections précédentes, nous disposons donc d'un estimateur convergent pour le calcul d'une espérance. Tout cela est bel et bien, mais pour que cette méthode soit utilisable en pratique il faudrait aussi pouvoir dire quelle est la *qualité* de cette évaluation empirique. En d'autres termes, on souhaite répondre aux deux questions suivantes :

1. Quel est l'ordre de grandeur de la différence entre la moyenne empirique et la véritable espérance ? Autrement dit, à quelle précision notre estimation est-elle pertinente ?
2. Quel est la probabilité que la différence entre moyenne empirique et véritable espérance soit plus grande que son ordre de grandeur typique ; autrement dit, quel risque courons-nous que notre évaluation soit nettement fautive ? Car après tout il y a toujours une probabilité non nulle que la moyenne empirique soit significativement différente de l'espérance ; par exemple, dans le cas du yahtzee, il se peut exceptionnellement que les N essais donnent autant de succès...

I.3.b Intervalles de confiance gaussiens

Nous allons répondre à ces questions dans le cadre du théorème 8.a, i.e. quand nous cherchons à déterminer l'espérance m d'une variable aléatoire X dont on sait simuler la loi facilement. Ici, nous supposons en outre que la v.a. X est L^2 .

Nous sommes alors en mesure d'appliquer le *théorème-limite central* (TLC). Notant $\sigma^2 := \text{Var}(X)$, nous savons en effet que $(\hat{m}_N - m)/N^{-1/2}$ converge en loi vers $\mathcal{N}(\sigma^2)$; [§] Par conséquent, quand N tend vers l'infini, la probabilité que $|\hat{m}_N - m|$ soit plus grand que $2\sigma/N^{1/2}$ (par exemple) tend vers $\mathbb{P}(\mathcal{N}(1) > 2) \leq 5\%$. Autrement dit, il y a une probabilité asymptotique de plus de 95 % pour que m soit dans l'intervalle $[\hat{m}_N - 2\sigma/N^{1/2}, \hat{m}_N + 2\sigma/N^{1/2}]$.

À ce stade, il y a deux objections importantes :

- Notre intervalle de confiance est défini en fonction de σ , qu'*a priori* nous ne connaissons pas !
- Notre probabilité d'erreur n'est valable qu'*asymptotiquement*, alors que nous voudrions savoir ce qu'on peut dire pour une valeur *fixée* de N ...

La première objection est la plus importante. Puisque nous ne connaissons pas σ *a priori*, la solution va être d'estimer la variance, estimation que nous faisons... par la méthode de Monte-Carlo ! On utilise pour cela le théorème suivant :

g.a Théorème. Soit X une variable aléatoire réelle L^2 ; alors pour X_1, X_2, \dots des v.a. i.i.d. de même loi que X , la variance empirique

$$\hat{\sigma}_N^2 := N^{-1} \sum_{i=1}^N X_i^2 - \left(N^{-1} \sum_{i=1}^N X_i \right)^2 \quad (2)$$

est un estimateur (fortement) convergent de $\sigma^2 := \text{Var}(X)$, et donc l'écart-type empirique $\hat{\sigma}_N$ est un estimateur (fortement) convergent de σ .

Démonstration. D'après la loi des grands nombres, $N^{-1} \sum_{i=1}^N X_i^2$ est un estimateur convergent de $\mathbb{E}(X^2)$ et $N^{-1} \sum_{i=1}^N X_i$ est un estimateur convergent de $\mathbb{E}(X)$, donc $\hat{\sigma}_N^2$ est un estimateur convergent de $\mathbb{E}(X^2) - \mathbb{E}(X)^2 = \sigma^2$. \square

g.b Remarque. On préconise parfois de choisir pour évaluer σ^2 l'estimateur $N/(N-1) \times \hat{\sigma}_N^2$, qui a l'avantage d'être sans biais, mais cela ne change pas grand-chose en pratique.

[§]. Convention : Dans ce cours, $\mathcal{N}(\sigma^2)$ notera la loi normale *centrée* de variance σ^2 .

Passons maintenant à la seconde objection : nous allons pas tout-à-fait répondre à celle-ci, mais simplement faire observer qu'elle n'est pas « trop » importante. En effet, même si notre approximation de la probabilité d'erreur est assez mauvaise (par exemple, si on s'attendait à une probabilité d'erreur de 5 % et qu'on a en fait une probabilité d'erreur de 6 %), l'essentiel est que *l'ordre de grandeur* de cette probabilité reste le même : tout ce qui nous intéresse, c'est de dire qu'il est très peu probable que la vraie valeur de p soit hors de notre intervalle de confiance, et de ce point de vue cela ne change pas grand-chose de savoir si notre estimation est fiable à 95 % ou à 94 %.

10.a Remarque. Attention toutefois : ce que nous venons de dire devient douteux quand on cherche des intervalles correspondant à des risques d'erreur très petits (disons, inférieurs à 0,1 %), car à un tel degré de précision la convergence vers la loi normale dans le théorème-limite central devient trop lente. En effet, pour être sûr que la probabilité d'erreur réelle soit raisonnablement proche d'une probabilité d'erreur estimée ε , il faut grossièrement réaliser au moins ε^{-2} expériences, [¶] soit au moins 1 000 000 expériences pour s'assurer qu'un intervalle de confiance à 99,9 % donné par l'approximation normale est fiable — alors que 500 expériences suffisent pour un intervalle de confiance à 95 %.

De toutes façons, comme l'estimateur de σ est convergent, l'intervalle de confiance obtenu aura *asymptotiquement* le niveau souhaité. [||] Nous pouvons résumer tout cela par le théorème suivant :

10.b Théorème. Soit X une variable aléatoire réelle L^2 et soient X_1, X_2, \dots des v.a. i.i.d. de même loi que X . Notons

$$\hat{m}_N := N^{-1} \sum_{i=1}^N X_i, \quad (3)$$

resp.

$$\hat{\sigma}_N^2 := N^{-1} \sum_{i=1}^N X_i^2 - \left(N^{-1} \sum_{i=1}^N X_i \right)^2. \quad (4)$$

Alors, pour $c_1, c_2 \geq 0$ (éventuellement infinis), $\mathbb{E}(X) := m$ appartient à l'intervalle $[\hat{m}_N - c_1 \hat{\sigma}_N / N^{1/2}, \hat{m}_N + c_2 \hat{\sigma}_N / N^{1/2}]$ avec un niveau de confiance asymptotiquement égal à $\mathbb{P}(\mathcal{N}(1) \in [-c_1, c_2]) = \int_{-c_1}^{c_2} e^{-x^2/2} dx / \sqrt{2\pi}$.

10.c Remarque. Dans le cas où on dispose d'un contrôle *a priori* sur la loi de X (par exemple si X est bornée, comme c'était le cas pour le yahtzee où on avait une indicatrice), il est aussi possible d'établir des intervalles de confiance *non asymptotiques* (dont l'expression serait plus compliquée). Toutefois nous ne nous intéresserons pas à cette question dans ce cours.

10.d Remarque. Attention, tout ce que nous venons de dire n'est valable que quand X est L^2 !! (alors que la convergence de l'estimateur de Monte-Carlo, elle, marche dès que X est L^1). Il faudra donc bien veiller à rester dans ce cadre pour appliquer le théorème 10.b ci-dessus. En pratique, pour vérifier

[¶]. Cela provient du *théorème de Berry-Essen*.

[||]. Cela provient du théorème de Slutsky, qui sera rappelé en cours.

l'hypothèse " $X \in L^2$ ", on pourra souvent majorer grossièrement $|X|$ (voir la § I.3.c pour un exemple). Dans les cas plus compliqués, on pourra éventuellement faire de cette hypothèse un « acte de foi », mais alors il faudra surveiller attentivement les indices qui alerteraient que celui-ci est erroné (en particulier, si l'estimateur $\hat{\sigma}_N^2$ semble ne pas converger), et dans tous les cas rester très prudent sur l'interprétation des résultats. À noter que si $X \notin L^2$ mais qu'on sait que $X \in L^p$ pour un $p > 1$, il existe des théorèmes analogues au théorème 10.b pour obtenir des intervalles de confiance asymptotiques, avec une vitesse de convergence en $N^{-1+1/p}$ (mais ce n'est plus la loi gaussienne qui intervient alors). Toutefois nous ne nous intéresserons pas à cette question dans ce cours.

I.3.c Exemple d'application

Voyons maintenant comment ce que nous venons de dire s'applique dans un cas concret. Nous reprenons ici l'exemple de la course à pied du § I.2.d, et nous cherchons à déterminer un intervalle de confiance à 80 % (ce qui, d'après la table de la § I.3.d *infra*, correspondra à plus ou moins 1,29 fois l'écart-type autour de la valeur empirique) de largeur 10^{-1} pour m , en essayant de faire le moins possible de calculs.

Commençons par vérifier que la variable aléatoire $\max_{1 \leq i \leq 8} X_i := X_*$ dont nous cherchons à déterminer l'espérance est bien L^2 . Comme tous les X_i sont positifs, on a $X_* \leq \sum_{i=1}^8 X_i$, d'où $\mathbb{E}[X_*^2] = \sum_{i,j=1}^8 \mathbb{E}[X_i X_j] \leq 64 \mathbb{E}[X^2]$ (par l'inégalité de Cauchy-Schwarz), où $\mathbb{E}_{\ln \mathcal{N}(4;0,03)}[X^2] = \mathbb{E}_{\mathcal{N}(4;0,03)}[(e^X)^2] = \mathbb{E}_{\mathcal{N}(1)}[e^{2 \times (4+0,03^{1/2} X)}] = e^8 \mathbb{E}_{\mathcal{N}(1)}[e^{2 \times 0,03^{1/2} X}]$ est fini vu que la loi normale a tous ses moments exponentiels finis.

Avant de passer à l'implémentation, faisons deux remarques sur l'évaluation pratique de σ :

- 11.a *Astuce.* (i). Pour évaluer σ , on peut se servir de la même simulation qui sert à évaluer m ;
- (ii). On n'a jamais besoin de connaître σ avec une précision trop grande ; par conséquent on peut s'en faire une première idée en lançant quelques centaines de simulations.

Notre algorithme suit les lignes suivantes (voir le code en annexe III.2) :

1. Choisir un nombre de simulations N ;
2. Simuler les chronos des 8 coureurs et déterminer leur maximum ; le faire N fois en tout ;
3. Faire la somme des N chronos maximums obtenus, qu'on appellera *somme*, ainsi que la somme des carrés de ces chronos maximum, qu'on appellera *sommecarrés* ;
4. Calculer $\hat{\sigma} := (\text{sommecarrés}/N - (\text{somme}/N)^2)^{1/2}$, et afficher cette valeur pour information ;
5. Calculer $\hat{m} := \text{somme}/N$ et retourner les valeurs numériques des bornes de l'intervalle $[\hat{m} - 1,29\hat{\sigma}/N^{1/2}; \hat{m} + 1,29\hat{\sigma}/N^{1/2}]$.

Nous commençons par lancer 500 simulations pour évaluer la valeur de σ : nous trouvons alors, dans mon cas, $\hat{\sigma} \simeq 8,31$ Maintenant, combien dois-je

faire de simulations pour la largeur de mon intervalle de confiance à 80 % soit inférieure à 10^{-1} ? La largeur de cet intervalle étant $2 \times 1,29\sigma \times N^{-1/2}$, il faut avoir $N \geq (2,58\sigma/0,1)^2$, qui vaut environ 46 000 en remplaçant σ par le $\hat{\sigma}$ que nous venons de trouver, valeur que nous arrondissons à 50 000 pour avoir un peu de marge.

La simulation avec $N = 50\,000$ donne alors, dans mon cas, un intervalle de confiance égal à $[70,233; 70,321]$ (qui est bien de largeur plus petite que 10^{-1}), ce qui permet de conclure que m commence vraisemblablement par 70,2... ou 70,3... — et c'est effectivement le cas.

Au passage, quelle était la pertinence de notre évaluation de σ sur 500 simulations ? Notre simulation principale donne finalement une valeur de $\hat{\sigma}$ égale à 7,64 : ainsi, bien que notre estimation initiale fût assez grossière, elle était effectivement suffisamment proche de la valeur véritable pour être utilisable dans le calcul du nombre de simulations nécessaires à l'obtention d'un intervalle de confiance de taille imposée.

12.a Astuce. Nous aurions pu garder en mémoire les résultats de notre première série de simulations [j'entends par là les valeurs de **somme** et **sommecarrés**] pour faire comme s'il s'était agi des 500 premières simulations de notre seconde série, de sorte que nous n'aurions plus eu que 49 500 simulations à effectuer. En l'occurrence, l'économie apportée aurait été minime ; toutefois cette idée peut s'avérer très utile pour pousser plus loin un calcul de Monte-Carlo déjà coûteux. Ainsi, si nous voulions passer de 50 000 à 70 000 simulations (par exemple, parce que l'intervalle de confiance obtenu serait accidentellement plus grand que 10^{-1}), nous pourrions utiliser les résultats des 50 000 premières simulations afin de ne plus avoir que 20 000 simulations à faire au lieu d'en recommencer 70 000 ! Il est donc recommandé, de manière générale, de toujours garder en mémoire les valeurs de **somme** et **sommecarrés** données par une série de simulations.

I.3.d Annexe : table des intervalles de confiance gaussiens

Il peut être utile de savoir convertir un niveau de confiance en nombre d'écarts-types, autrement dit de savoir quelles valeurs de c_1 et c_2 suffisent pour obtenir une valeur fixée de $\mathbb{P}(\mathcal{N}(1) \in [-c_1, c_2])$. À cette fin, vous trouverez ci-dessous une petite table dans le cas (le plus courant) où on prend $c_1 = c_2$: notez que dans ce cas, $\mathbb{P}(|\mathcal{N}(1)| \leq c)$ s'exprime en termes de la *fonction d'erreur* définie par :

$$\text{erf}(x) := \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy = \mathbb{P}(|\mathcal{N}(1)| \leq \sqrt{2}x). \quad (5)$$

c	0	0,68	1	1,29	1,65	1,96	2
$\mathbb{P}(\mathcal{N}(1) \leq c)$	0	0,50	0,68	0,80	0,90	0,95	0,954

c	2,33	2,58	2,81	3	3,10	3,30	∞
$\mathbb{P}(\mathcal{N}(1) \leq c)$	0,98	0,99	0,995	0,997 3	0,998	0,999	1

TABLE 1 – Niveau de confiance en fonction du nombre d'écarts-types. Tous les arrondis sont faits de façon à sous-évaluer le niveau de confiance.

I.4 Méthode de Monte-Carlo pour le calcul d'une intégrale

I.4.a Fonctions à support borné

Vous avez sans doute appris, lors de votre cours de probabilités, qu'une espérance pouvait formellement se définir comme une intégrale. Or la méthode de Monte-Carlo nous permet d'évaluer facilement une espérance... Du coup, nous avons l'idée de renverser le paradigme : *en écrivant une intégrale comme une espérance*, nous allons pouvoir utiliser la méthode de Monte-Carlo pour calculer une intégrale !

Détaillons un premier exemple. Supposons qu'on vous demande d'évaluer le volume de la boule unité de dimension 6 et que vous ne connaissiez pas la formule pour celle-ci. Notant B notre boule, ce qu'on demande est en fait de calculer

$$\text{Vol} := \int_B dx = \int_{\mathbb{R}^4} \mathbf{1}_{|x| \leq 1} dx. \quad (6)$$

Comment interpréter cette intégrale comme une espérance ? Commençons par remarquer que B est incluse dans l'hypercube centré en 0 de demi-arête 1, que nous noterons Q , de sorte que

$$\text{Vol} = \int_Q \mathbf{1}_{|x| \leq 1} dx. \quad (7)$$

Notons maintenant \mathbb{P}_Q la probabilité uniforme sur Q , à savoir $d\mathbb{P}_Q(x) = dx/64$,^[*] de sorte que

$$\text{Vol}(B) = 64 \int_Q \mathbf{1}_{|x| \leq 1} d\mathbb{P}_Q(x) = 64 \mathbb{E}_Q[\mathbf{1}_{|X| \leq 1}]. \quad (8)$$

Or, \mathbb{P}_Q est très facile à simuler, puisque la loi d'un point (x_1, \dots, x_6) sous \mathbb{P}_Q revient à tirer indépendamment les x_i selon la loi uniforme sur $[-1, 1]$! On en déduit donc l'algorithme de Monte-Carlo suivant pour calculer $\text{Vol}(B)$:

1. Pour N grand, simuler N variables aléatoires i.i.d. uniformes sur Q (chacune étant simulée par 6 v.a. i.i.d. uniformes sur $[-1, 1]$) ;
2. Pour chaque $x \in Q$ simulé, calculer $|x|$ pour savoir si $x \in B$;
3. En déduire un intervalle de confiance pour $\mathbb{E}(\mathbf{1}_{X \in B})$ par la méthode du théorème 10.b ;
4. Multiplier cet intervalle par 64 pour en déduire un intervalle de confiance (de même niveau) pour $\text{Vol}(B)$.

Ici par exemple, en choisissant $N = 1\,000\,000$ et en cherchant un intervalle de confiance à 95 % (soit des fluctuations de plus ou moins 1,96 fois l'écart-type), je trouve un intervalle de confiance égal à $[5,140, 5,209]$, qui contient effectivement la bonne réponse $\pi^3/6 \simeq 5,168$

**

Nous pouvons immédiatement généraliser cette méthode sous la forme du théorème suivant :

[*]. Le nombre 64 correspond au volume de Q , qui est simplement le produit de ses six dimensions, soit 2^6 .

13.a Théorème. Soit $f: \mathbb{R}^d \rightarrow \mathbb{R}$ une fonction L^1 telle que f est nulle en-dehors d'un pavé $Q := [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_d, b_d]$. Alors, si \mathbb{P}_Q note la loi d'une v.a. uniforme sur Q , on a

$$\int_{\mathbb{R}^d} f(x) dx = \prod_{i=1}^d (b_i - a_i) \times \mathbb{E}_Q[f(X)], \quad (9)$$

ce qui permet de calculer $\int f$ par la méthode de Monte-Carlo du théorème 8.a. En outre, si f est L^2 , alors la v.a. $f(X)$ est L^2 sous \mathbb{P}_Q , ce qui permet de calculer des intervalles de confiance pour f d'après la méthode du théorème 10.b.

Par ailleurs, pour simuler une variable aléatoire X de loi \mathbb{P}_Q sur \mathbb{R}^d , il suffit de tirer des v.a. U_1, \dots, U_d uniformes sur $[0, 1]$ et de poser $X = (a_1 + [b_1 - a_1]U_1, \dots, a_d + [b_d - a_d]U_d)$.

I.4.b Cas général

Imaginons maintenant qu'on cherche à évaluer l'intégrale

$$\int_{-\infty}^{+\infty} \frac{1}{1 + |x|^3} dx \quad (10)$$

par la méthode de Monte-Carlo. [†] On ne peut clairement pas appliquer le théorème 13.a, puisque la fonction à intégrer n'est pas à support borné... Certes. Toutefois, rien dans la méthode de Monte-Carlo n'exige que la loi de probabilité utilisée pour l'échantillonnage (que nous appellerons dans la suite de ce cours *loi d'échantillonnage*) soit une loi de type uniforme! Voyons ainsi comment nous pouvons procéder dans le cas de l'intégrale (10). Soit \mathbb{P}_1 la loi de Cauchy standard, de densité

$$d\mathbb{P}_1(x) = \frac{dx}{\pi(1 + x^2)}. \quad [\ddagger] \quad (11)$$

On peut alors réécrire

$$\int_{-\infty}^{+\infty} \frac{1}{1 + |x|^3} dx = \int \frac{\pi(1 + x^2)}{1 + |x|^3} d\mathbb{P}_1(x) = \mathbb{E}_1 \left[\frac{\pi(1 + x^2)}{1 + |x|^3} \right], \quad (12)$$

qui est bien l'espérance d'une fonction L^2 . Au final, on peut donc évaluer (10) par l'algorithme suivant (voir le code en III.4.a) :

1. Simuler N variables indépendantes de loi de Cauchy standard ;
2. Pour chacune de ces variables, évaluer $\pi(1 + x^2) / (1 + |x|^3)$;
3. Prendre la moyenne des valeurs obtenues pour obtenir un estimateur de l'intégrale (on peut également calculer un intervalle de confiance par la méthode du théorème 10.b).

[†]. Dans ce cas précis, c'est à vrai dire une idée plutôt saugrenue, car les méthodes d'intégration déterministes seraient plus efficaces — cf. § I.4.c. On peut même calculer une expression analytique exacte si on se souvient de son cours de fonctions holomorphes...

[‡]. Rappelons que cette loi se simule facilement : c'est la tangente d'un angle uniforme sur $(-\pi/2, +\pi/2)$.

Ainsi fait avec $N = 60\,000$, j'obtiens un intervalle de confiance à 95 % égal à $[2,406; 2,425]$. La vraie valeur est $4\sqrt{3}\pi/9 \simeq 2,418$.

✱

Bien sûr, on aurait pu choisir une autre loi d'échantillonnage que la loi de Cauchy standard. Par exemple, si on avait pris la loi double-exponentielle de paramètre 1, dont la densité est

$$d\mathbb{P}_2(x) = \frac{dx}{2e^{|x|}}, \quad (13)$$

on aurait pu écrire de même

$$\int_{-\infty}^{+\infty} \frac{1}{1+|x|^3} dx = \mathbb{E}_2 \left[\frac{2e^{|x|}}{1+|x|^3} \right]. \quad (14)$$

Si nous regardons le résultat obtenu pour $N = 60\,000$ avec cette loi d'échantillonnage [voir l'annexe III.4.b pour le code], je trouve pour ma part un intervalle de confiance égal à $[2,407; 2,422]$, ce qui est *a priori* de qualité comparable à la méthode précédente.

Cependant, regardons de plus près nos deux lois d'échantillonnages en poussant les calculs jusqu'à $N = 1\,000\,000$. L'intervalle de confiance obtenu pour la densité m_1 est alors $[2,415\,2; 2,419\,8]$, tandis que celui pour la densité m_2 est $[2,409\,4; 2,423\,3]$: cette fois-ci, la densité m_2 apparaît comme bien moins bonne que sa concurrente ! Pourquoi ? Eh bien, l'échantillonnage double-exponentiel présente en fait un gros défaut : la variable aléatoire qu'il conduit à intégrer n'est pas L^2 ! En effet,

$$\mathbb{E}_2 \left[\left(\frac{2e^{|x|}}{1+|x|^3} \right)^2 \right] = \int_{\mathbb{R}} \left(\frac{2e^{|x|}}{1+|x|^3} \right)^2 \frac{dx}{2e^{|x|}} = \int_{\mathbb{R}} \frac{2e^{|x|}}{(1+|x|^3)^2} dx = \infty. \quad (15)$$

Cette seconde méthode a donc une convergence asymptotique plus lente que $O(N^{-1/2})$, et par-dessus le marché, on ne pouvait pas légitimement lui appliquer le théorème 10.b sur les intervalles de confiance gaussiens ! *Les intervalles de confiance que nous avons écrits pour la loi d'échantillonnage m_2 n'avaient donc pas de sens* ; seuls les estimateurs de (10) (càd. les centres des « intervalles de confiance » en question) en avaient un. [§]

✱

De manière générale, on retiendra bien l'idée suivante :

15.a Mémoire.

- ☛ Il n'y a pas une façon de calculer une intégrale par la méthode de Monte-Carlo, mais une infinité, en fonction du choix de la loi d'échantillonnage.
- ☛ Le choix de la loi d'échantillonnage est pratiquement libre. Il doit toutefois vérifier les contraintes suivantes :
 - ☛ La loi d'échantillonnage doit avoir une densité non nulle en presque-tous les points où la fonction à intégrer est non nulle ;

[§]. Vous voyez donc sur cet exemple qu'il n'est pas toujours évident de détecter expérimentalement que la condition L^2 n'est pas vérifiée. On pourrait en fait montrer qu'ici, les choses se passent « numériquement comme dans le cas L^2 » tant que $N \lesssim 200\,000$.

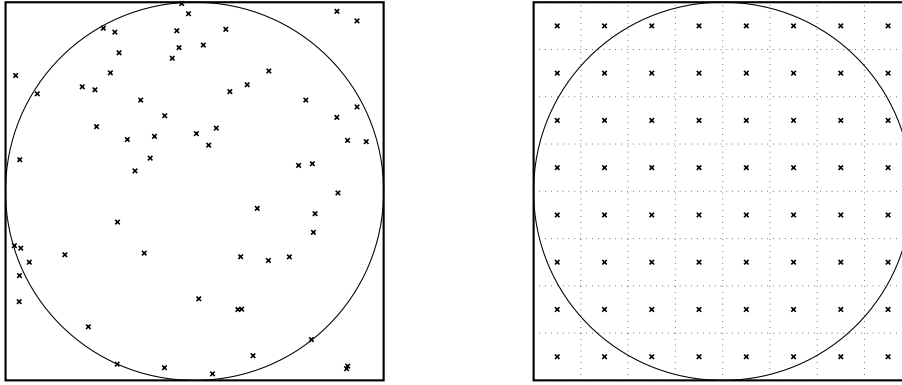


FIGURE 1 – Échantillonnage de Monte-Carlo (à gauche) contre échantillonnage régulier (à droite)

- ☛ On doit pouvoir calculer la densité de la loi d'échantillonnage ;
- ☛ Si on souhaite donner des intervalles de confiance gaussiens, on doit veiller à ce que la variable aléatoire qu'on se ramène à intégrer soit L^2 . [¶]

16.a Remarque. Évidemment, tout ce que nous avons dit dans le cas des fonctions à support non bornée s'applique aussi aux fonctions à support borné : le mémorandum 15.a reste entièrement valable pour ces fonctions-là, et le théorème 13.a n'est donc qu'une option parmi d'autres.

I.4.c Comparaison avec les méthodes d'intégration déterministes

Reprenons l'exemple de la § I.4.a sur le calcul du volume de la 6-boule. Que se serait-il passé si, au lieu de tirer nos points au hasard, nous les avions répartis régulièrement ? En d'autres termes, nos 1 000 000 points auraient constitué une « grille » hypercubique de $\sqrt[6]{1\,000\,000} = 10$ points de côté, chaque point de la grille étant au centre d'un petit hypercube pavant Q (voir la figure I.4.c pour une illustration de cette idée en dimension 2). La valeur de $\text{Vol}(B)$ que nous aurions ainsi évaluée aurait correspondu au calcul de

$$64 \times \sum_{(n_1, \dots, n_6) \in \{1, \dots, 10\}^6} \chi \left(\left(-1 + 2 \times \frac{n_1 - 1/2}{10} \right)^2 + \dots + \left(-1 + 2 \times \frac{n_6 - 1/2}{10} \right)^2 \leq 1 \right) / 1\,000\,000, \quad (16)$$

où $\chi(A)$ est une notation alternative pour $\mathbf{1}_A$. Cette valeur est égale à environ 4,891, ce qui est nettement moins bon qu'avec la méthode de Monte-Carlo. Pire : si nous essayons de calculer un intervalle d'erreur, en encadrant l'intégrale de $\mathbf{1}_Q$ sur chaque petit cube de notre maillage à partir de ses infimum et supremum sur ce cube, nous obtenons seulement que

$$\text{Vol}(B) \in [1,355; 15,619], \quad (17)$$

[¶]. À noter que le fait que la v.a. soit L^1 (qui est la condition pour que la méthode de Monte-Carlo converge, sans spécification sur la vitesse) est automatiquement satisfait dès que l'intégrale à calculer est bien définie.

ce qui est certes juste, mais terriblement décevant !

Comment se fait-il que la méthode de Monte-Carlo soit meilleure que la méthode déterministe ? Le point est le suivant : quand on utilise la méthode de Monte-Carlo, l'erreur d'évaluation en chaque point est importante, mais comme ces erreurs sont centrées et *indépendantes*, leur accumulation croît seulement en $N^{1/2}$ quand il y a N termes à sommer. Avec la méthode déterministe, par contre, l'erreur commise à chaque terme est certes plus petite, mais rien ne permet d'exclure que toutes ces erreurs aillent dans le même sens (et de fait, c'est le cas dans certaines situations), et du coup l'accumulation des erreurs fait cette fois apparaître un facteur N .

Ainsi, dans le cas de l'intégration déterministe d'une fonction régulière par la méthode du point médian,^[III] l'erreur d'évaluation en chaque point décroît en $O(N^{-2/d})$, ce qui donne également une erreur en $O(N^{-2/d})$ après sommation ; tandis qu'avec la méthode de Monte-Carlo, l'erreur d'évaluation en chaque point est en $O(1)$, mais après sommation on obtient du $O(N^{-1/2})$. À partir de la dimension 5, c'est donc la méthode de Monte-Carlo qui converge le plus vite.

De plus, quand on intègre une fonction peu régulière (comme par exemple notre fonction $\mathbf{1}_{|x| \leq 1}$, qui était discontinue), il peut arriver que la convergence de la méthode déterministe soit encore plus lente,^[*] au point que la méthode de Monte-Carlo (dont l'efficacité ne dépend pas de la régularité de la fonction) l'emporte même en dimension inférieure à 4.

On pourra donc retenir le mémorandum suivant :

17.a Mémorandum.

- ☛ À coût de calcul fixé, les méthodes d'intégration déterministes sont d'autant moins efficaces que la dimension de l'espace ambiant est élevée.
- ☛ La méthode de Monte-Carlo, en revanche, converge toujours à vitesse $O(N^{-1/2})$ quand on a N points à simuler (à condition toutefois que la variable à intégrer soit L^2), quelle que soit la dimension.
- ☛ À partir de la dimension 5, il est recommandé d'utiliser systématiquement la méthode de Monte-Carlo, sauf cas très particulier.
- ☛ Les méthodes déterministes requièrent en outre, pour fonctionner efficacement, que la fonction à intégrer soit suffisamment régulière.
- ☛ À l'inverse, la méthode de Monte-Carlo est indifférente à la régularité de la fonction : on la privilégiera donc pour intégrer les fonctions peu régulières, même en petite dimension.

[III]. Il peut toutefois exister d'autres méthodes déterministes plus efficaces quand la fonction à intégrer est de classe au moins C^3 .

[*]. Dans le cas de la fonction $\mathbf{1}_{|x| \leq 1}$, par exemple, la méthode déterministe avec intervalle de certitude converge seulement en $O(N^{-1/6})$. (Cela dit, on pourrait en fait montrer que la valeur estimée du volume converge pour sa part à vitesse $O(N^{-1/3})$, soit aussi vite que pour une fonction régulière).

I.5 Choix de l'échantillonnage

I.5.a Notion d'efficacité

Nous avons souligné dans la dernière section ci-dessus qu'il existait plusieurs façons d'appliquer la méthode de Monte-Carlo pour le calcul d'une même intégrale. Une question naturelle se pose alors : y en a-t-il qui soient meilleures que d'autres, et si oui lesquelles ?

Dans un premier temps, nous allons étudier cette problématique dans un cadre abstrait. Nous supposons donnée une mesure de référence notée dx (dont la masse totale peut être aussi bien finie qu'infinie) telle que l'intégrale à calculer s'écrive

$$I := \int_{\Omega} f(x) dx \quad (18)$$

pour une certaine fonction dx -intégrable f . Pour calculer I par la méthode de Monte-Carlo, nous utilisons une loi d'échantillonnage \mathbb{P}_m ayant une densité m par rapport à dx , i.e. $d\mathbb{P}_m(X = x) = m(x)dx$; comme \mathbb{P}_m est une loi de probabilité, cela signifie qu'on doit avoir $m(x) \geq 0 \forall x$ et $\int_{\Omega} m(x) dx = 1$. On suppose en outre que m est telle que $\{m(x) > 0 \text{ ou } f(x) = 0\}$ dx -presque-partout ; on peut alors écrire $I = \mathbb{E}_m[f/m]$ et donc appliquer le théorème 8.a. On aura convergence en $O(N^{-1/2})$ si f/m est L^2 pour la loi \mathbb{P}_m , c.à.d. si $\int_{\Omega} f(x)^2 m(x)^{-1} dx < \infty$.

Ce critère L^2 est donc la première condition pour avoir une bonne vitesse de convergence. Toutefois, il existe une infinité de fonctions m qui le vérifient, et si toutes ces fonctions convergent certes en $O(N^{-1/2})$, la constante cachée dans la notation « grand O » peut être très différente de l'une à l'autre ! Quelle est cette constante ? La réponse est apportée par le théorème suivant, qui est un simple raffinement du théorème 10.b en utilisant que l'estimateur $\hat{\sigma}_N$ converge vers σ :

18.a Théorème. *Soient X une v.a. réelle L^2 dont on cherche à déterminer l'espérance par la méthode de Monte-Carlo, et L un niveau de confiance. Alors l'intervalle de confiance de niveau asymptotique L donné par la méthode du théorème 10.b a une longueur équivalente, quand le nombre d'expériences N tend vers l'infini, à*

$$2\sqrt{2}erf^{-1}(L) \text{Var}(X)^{1/2} N^{-1/2}, \quad (19)$$

où erf^{-1} désigne la fonction d'erreur inverse, c.à.d. la fonction réciproque de erf .

En outre, si X est maintenant une v.a. L^1 mais pas L^2 , alors la convergence de la méthode de Monte-Carlo est strictement plus lente que $O(N^{1/2})$, au sens où $N^{1/2}|\hat{m}_N - \mathbb{E}(X)| \xrightarrow{pba} +\infty$ quand $N \rightarrow \infty$.

Le théorème 18.a nous donne la vitesse de convergence d'une méthode de Monte-Carlo en fonction du nombre de simulations effectuées. En pratique, ce qui nous limite n'est pas le nombre de simulations, mais le *temps de calcul*. Ce temps de calcul est manifestement proportionnel au nombre de simulations, mais le facteur de proportionnalité peut varier d'une méthode à l'autre, en fonction :

- De la complexité de la simulation de la loi \mathbb{P}_m ;

- De la complexité du calcul de la densité m ;^[†]
- Dans une moindre mesure, du fait que la mesure \mathbb{P}_m privilégie des zones où les calculs sont plus ou moins compliqués.

Si nous notons τ_m le temps moyen d'une étape de calcul (chaque étape incluant non seulement la simulation de la loi \mathbb{P}_m , mais aussi le calcul de f/m) pour la loi d'échantillonnage m , on voit donc que si on dispose d'un temps de calcul t , le temps de calcul pour obtenir un intervalle de confiance de niveau L de longueur ε est équivalent, quand $\varepsilon \rightarrow 0$, à

$$8erf^{-1}(L)^2 \tau_m \text{Var}_m(X) \varepsilon^{-2}. \quad (20)$$

19.a Définition. On définit l'*efficacité* d'une méthode de Monte-Carlo correspondant à la loi d'échantillonnage \mathbb{P}_m comme $\mathcal{E}ff_m := 1/(\tau_m \text{Var}_m(X))$. L'efficacité est donc définie de sorte qu'à niveau de confiance et précision égaux, le temps de calcul nécessaire à l'évaluation d'une intégrale par la méthode de Monte-Carlo soit directement inversement proportionnel à l'efficacité de cette méthode.

Par conséquent, pour choisir entre deux méthodes de Monte-Carlo, la règle est simple : il faut toujours opter pour celle dont l'efficacité est la plus grande !

19.b Remarque. L'efficacité est une grandeur *dimensionnée* : quand la quantité à évaluer est adimensionnée, elle est homogène à l'inverse d'un temps, et plus généralement si la quantité à évaluer est homogène à un *truc*, l'efficacité est homogène à un $\text{truc}^{-2} \cdot \text{temps}^{-1}$.

19.c Remarque. L'efficacité dépend des détails de la programmation et de la machine utilisée. Il se peut même qu'en fonction de ces paramètres, une méthode soit tantôt moins efficace qu'une autre et tantôt plus !

Et comment évaluer l'efficacité d'une méthode ? Très simplement, en lançant une série de simulations puis en regardant le temps mis et la variance estimée. Le nombre de simulations effectuées sera choisi suffisamment grand pour que l'évaluation de l'efficacité soit assez précise, mais pas trop afin que le temps de calcul consacré à celle-ci reste faible. 1 000 simulations semble un choix raisonnable.

19.d Astuce. Calculer l'efficacité d'une méthode de Monte-Carlo prend peu de temps par rapport à la simulation finale ; n'hésitez donc pas le faire chaque fois que vous hésitez entre deux méthodes !

I.5.b Échantillonnage idéal

Dans cette sous-section, nous allons déterminer quelle est la loi d'échantillonnage la plus efficace à *nombre de simulations fixé*. L'intérêt de cette

[†]. Notez que le fait qu'une loi soit simple à simuler ne garantit aucunement que sa densité soit simple à calculer : ainsi, la somme indépendante d'une loi normale et d'une loi de Cauchy n'a pas d'expression analytique... À l'inverse, la plupart des lois dont la densité est simple à calculer peuvent être simulées assez facilement grâce à la méthode de rejet, encore qu'il y ait des exceptions — par exemple, je ne vois pas comment simuler par rejet la loi sur \mathbb{R} dont la densité en x vaut 2^n si x est à distance moins de $1/4^n$ d'un entier $n \geq 2$ et 0 sinon...

étude sera évidemment limitée par le fait qu'elle ne prenne pas en compte le temps de calcul par simulation, mais, comme nous l'avons observé en 19.c, ce dernier dépend des détails d'implémentation et d'architecture de l'ordinateur et ne pourrait donc pas être abordé d'un point de vue rigoureusement mathématique.

Si nous reprenons le formalisme de la § I.5.a, le nombre de simulation requis pour atteindre une précision donnée, d'après le théorème 18.a est d'autant court que $\text{Var}_m(f/m)$ est petit. Or

$$\text{Var}_m(f/m) = \mathbb{E}_m(f^2/m^2) - \mathbb{E}_m(f/m)^2 = \int_{\Omega} \frac{f(x)^2}{m(x)} dx - I^2. \quad (21)$$

Pour optimiser notre méthode de Monte-Carlo, il faut donc minimiser $\int_{\Omega} f(x)^2 m(x)^{-1} dx$ sous la contrainte que m est une densité de probabilité.

Comme $|f(x)| = [f(x)^2/m(x)]^{1/2} \times m(x)^{1/2}$, d'après l'inégalité de Cauchy-Schwarz,

$$\int_{\Omega} |f(x)| dx \leq \left(\int_{\Omega} \frac{f(x)^2}{m(x)} dx \right)^{1/2} \times \left(\int_{\Omega} m(x) dx \right)^{1/2}. \quad (22)$$

Mais comme m est une densité de probabilité, la troisième intégrale de l'expression ci-dessus vaut 1, d'où

$$\int_{\Omega} \frac{f(x)^2}{m(x)} dx \geq \left(\int_{\Omega} |f(x)| dx \right)^2. \quad (23)$$

Si nous arrivons à trouver un cas d'égalité pour (23), celui-ci optimisera donc la vitesse de convergence. Or pour avoir cas d'égalité dans Cauchy-Schwarz, il suffit (et il faut) que $f(x)^2/m(x)$ soit proportionnelle à $m(x)$, c.à.d. que $m(x)^2 \propto f(x)^2$, soit $m(x) \propto |f(x)|$. Ainsi,

20.a Théorème. *L'unique loi qui minimise $\text{Var}_m(f/m)$ correspond à la densité*

$$m_{\text{idéale}}(x) = \frac{|f(x)|}{\int_{\Omega} |f(x)| dx}. \quad (24)$$

20.b Remarque. Pour une telle loi, la fonction f/m dont on prend l'espérance est alors de valeur absolue constante avec le même signe que f . En particulier, si f est de signe constant, on se retrouve alors avec f/m constante : la méthode de Monte-Carlo revient alors à calculer l'espérance d'une constante, et la variance en résultant est nulle !

Au vu de la remarque ci-dessus, le théorème 20.a semble trop beau pour être honnête... De fait, il est inapplicable : en effet, pour calculer $m_{\text{idéale}}(x)$, il faudrait connaître $\int_{\Omega} |f(x)| dx$; or quand f est positive, c'est précisément l'intégrale que nous cherchons à calculer depuis le départ ! Et de manière générale, il est peu plausible qu'on connaisse $\int |f|$ si on ignore la valeur de $\int f$. Sans compter qu'il faudrait aussi savoir simuler $\mathbb{P}_{m_{\text{idéale}}}$...

I.5.c Principe d'échantillonnage préférentiel

Malgré ses limitations pratiques, le théorème 20.a suggère quand même l'heuristique suivante pour choisir une loi d'échantillonnage :

20.c Mémorandum.

- ☛ *Pour obtenir une méthode de Monte-Carlo efficace, il faut essayer de prendre une loi d'échantillonnage dont la densité soit « le plus proportionnelle possible » à $|f|$.*
- ☛ *Un bon critère est souvent de choisir m de sorte que $m / |f|$ soit bornée à la fois inférieurement (par une quantité > 0 s'entend) et supérieurement. Si on ne peut satisfaire qu'une seule de ces conditions, il faut toujours favoriser la première.*

Expliquons pourquoi il vaut toujours mieux que $m / |f|$ s'écarte vers de grandes valeurs plutôt que vers de petites. S'il y a un endroit où m est très grande par rapport à $|f|$ (par exemple, où m serait non nulle alors que f est nulle), tout ce que cela signifie est qu'on va perdre un peu de temps à échantillonner une zone qui ne contribue presque pas à l'intégrale I . Mais si à l'inverse il y a un endroit où m est trop petite, c'est beaucoup plus grave : en effet, c'est une zone dans laquelle $|f|/m$ sera gigantesque, de sorte que l'estimateur de I sera extrêmement sensible à la moindre visite en plus ou en moins dans cette zone (même si ces visites seront rares) et que la variance risque de devenir très grande.

Pour illustrer cette idée, regardons les deux densités de probabilité f et g sur $[-\varepsilon, +1]$ définies par

$$\begin{cases} f(x) := M^{-1}, & g(x) := M & \text{pour } -\varepsilon \leq x < 0; \\ f(x) := 1 - \varepsilon M^{-1}, & g(x) := 1 - \varepsilon M & \text{pour } 0 \leq x \leq 1, \end{cases} \quad (25)$$

où $M \gg 1$ est destiné à tendre vers ∞ et $\varepsilon \ll M^{-1}$ est destiné à tendre vers 0. En échantillonnant selon la densité g pour calculer $\int f$, on trouve une variance équivalente à $M\varepsilon$, tandis qu'en échantillonnant selon la densité f pour calculer $\int g$, la variance est équivalente à $M^3\varepsilon$. Cela montre bien que le cas où la densité est par endroits beaucoup plus grande que la fonction à intégrer est moins défavorable que celui où c'est l'inverse.

Dans le jargon des méthodes de Monte-Carlo, le choix d'une loi d'échantillonnage privilégiant les endroits où la valeur de la fonction à intégrer est grande est appelé *échantillonnage préférentiel* [*importance sampling* en anglais].

I.5.d Exemples

Pour illustrer les idées du mémorandum 20.c, voyons comment nous pouvons les appliquer au calcul de l'intégrale (10). On aimerait trouver une loi facile à simuler, dont la densité soit facile à calculer et ressemble le plus possible à $1/(1+|x|^3)$. Nous avons déjà vu dans la § I.4.b que le densité m_1 correspondant à la loi *Cauchy*(1) semblait plus efficace que la loi *DoubleExp*(1), ce qui ne nous étonne pas au vu des critères que nous avons donné pour trouver une bonne densité m : en effet, le rapport m_1 / f (où f note $1/(1+|x|^3)$) est borné inférieurement alors que le rapport m_2 / f ne l'est pas — au point, même, que f/m_2 n'est même pas $L^2(\mathbb{P}_2)$ en l'occurrence. Essayons maintenant de trouver une loi d'échantillonnage encore plus efficace que m_1 . Je propose une loi de Cauchy de paramètre 0,65 :

$$m_3(x) := \frac{0,65}{\pi(0,65^2 + x^2)}, \quad (26)$$

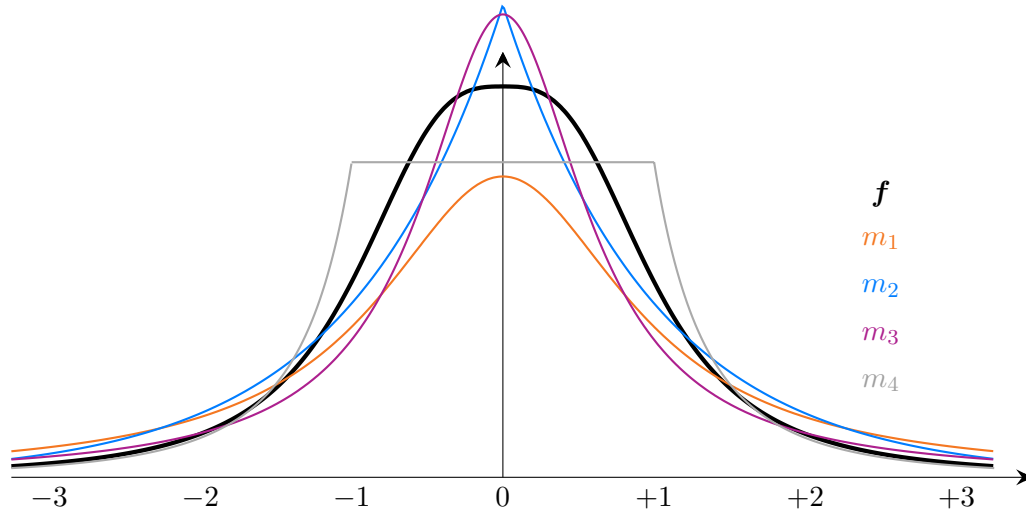


FIGURE 2 – La courbe de la fonction $f(x) = 1/(1+|x|^3)$ et celles des densités m_1 , m_2 , m_3 et m_4 . La fonction f a été mise à l'échelle pour que son intégrale vaille 1, de façon à faciliter la comparaison avec les différentes densités.

dont la densité ressemble beaucoup plus au profil de la fonction à intégrer, au moins visuellement (cf. figure 2). L'algorithme de Monte-Carlo utilisant cette loi d'échantillonnage est présenté en III.4.c. Quand je compare l'efficacité de cette méthode avec celle utilisant la densité de Cauchy m_1 , je trouve les efficacités suivantes :^[‡]

Densité m_1 : $\simeq 85\,000\text{ s}^{-1}$;

Densité m_3 : $\simeq 160\,000\text{ s}^{-1}$.

On voit qu'effectivement, le calcul avec la densité m_3 est plus efficace qu'avec la densité m_1 : presque deux fois plus !

Une autre idée pour améliorer la proportionnalité entre m et f est de carrément changer le type de loi utilisé. En effet, un défaut des densités de Cauchy (quel que soit leur paramètre) pour notre objectif est qu'elles ont une décroissance en $\Theta(x^{-2})$ à l'infini, ce qui fait qu'elles ne sont jamais bornées inférieurement par rapport à la fonction à intégrer dont la décroissance est en $\Theta(x^{-3})$. Nous allons donc essayer de trouver une loi dont la densité soit bornée à la fois inférieurement *et supérieurement* par rapport à f , tout en étant facile à simuler et en ayant une densité facile à calculer. Je propose la densité suivante :

$$m_4(x) := \begin{cases} 1/3 \times |x|^{-3} & \text{si } |x| > 1 ; \\ 1/3 & \text{si } |x| \leq 1, \end{cases} \quad (27)$$

qui effectivement ressemble visuellement beaucoup plus à la vraie densité sur toute la zone $|x| \geq 2$ [voir figure 2]. Un algorithme simple pour simuler cette loi est le suivant :

1. Tirer une v.a. Y uniforme sur $(0, 1)$;

[‡]. J'ai fait mes calculs d'efficacité avec $N = 15\,000$, car pour $N = 1\,000$ le calcul était tellement rapide que l'évaluation du temps de calcul par *MATLAB* manquait de précision.

2. Avec probabilité $1/6$, renvoyer $X = -Y^{-1/2}$; avec probabilité $1/6$, renvoyer $X = Y^{1/2}$; sinon, renvoyer $X = -1 + 2Y$.

Si nous appliquons l'algorithme de Monte-Carlo avec cette loi [voir code en III.4.d], on trouve une efficacité d'environ $450\,000\text{ s}^{-1}$: la calcul est encore presque trois fois plus rapide !

23.a Remarque. Essayer d'avoir une densité d'échantillonnage à peu près proportionnelle à l'intégrande est un bon précepte ; cependant il faut toujours mettre cette recommandation en regard du temps de calcul nécessaire à la simulation de la variable aléatoire et au calcul de la densité !

Dans les exemples ci-dessus, quand nous passons de la loi d'échantillonnage \mathbb{P}_1 à la loi \mathbb{P}_3 , seuls les paramètres changeaient, de sorte que nous pouvions légitimement nous attendre à ce que le temps de calcul par étape de simulation reste pratiquement inchangé [§] et donc que seul joue, pour comparer l'efficacité des deux lois, l'efficacité par étape [¶] — celle que le théorème 20.a optimise. Quand nous passons de la loi \mathbb{P}_3 à la loi \mathbb{P}_4 , en revanche, il n'est pas évident de deviner si la complexité des calculs pour la nouvelle loi sera plus grande ou plus petite (éventuellement de beaucoup !) que celle pour l'ancienne... [||] On aurait très bien pu imaginer que la vitesse de calcul de chaque étape devienne tellement plus lente avec la loi \mathbb{P}_4 que cela aurait complètement annulé le gain au niveau de l'efficacité par étape !

23.b Remarque. Pour continuer sur la remarque précédente, il ne faut pas perdre de vue non plus que l'optimisation de la programmation peut aussi conduire à des gains de vitesse tout-à-fait conséquents ! On pensera donc à ne pas seulement comparer l'efficacité entre deux méthodes de Monte-Carlo différentes, mais aussi entre deux implémentations d'une même méthode de Monte-Carlo...

23.c Remarque. Même sans tenir compte du critère du temps de calcul par étape, il n'est pas toujours facile de deviner *a priori* quelle densité aura la meilleure efficacité par étape... Qui saurait dire, juste en regardant la figure 2, que $\text{Var}_4(f/m_4)$ est trois fois plus petit que $\text{Var}_3(f/m_3)$, lui-même deux fois plus petit que $\text{Var}_1(f/m_1)$, et que $\text{Var}_2(f/m_3)$ est infini ? Les critères du mémorandum 20.c peuvent nous guider, mais eux-mêmes ne marchent pas toujours comme prévu... Il faut donc savoir rester ouvert d'esprit et ne pas dénigrer trop vite un choix de densité : bien souvent, rien ne remplace l'expérience.

*
**

Dans les deux exemples précédents, la mesure de référence que nous avions était une mesure de Lebesgue (resp. sur \mathbb{R} et \mathbb{R}^{16}). Mais le principe d'échantillonnage préférentiel marche en fait aussi bien par rapport à n'importe quel type de mesure de référence, y compris sur des espaces de probabilité abstraits ! Voici un exemple illustrant cette idée.

[§]. De fait, c'était effectivement le cas.

[¶]. Ce que j'appelle « efficacité par étape », en regard du théorème 18.a, est la quantité $1/\text{Var}_m(f/m)$: avec cette définition, l'efficacité d'une méthode est le produit efficacité par étape \times nombre d'étapes calculées par unité de temps.

[||]. En l'occurrence, il se trouve que le temps que calcul par étape est pratiquement le même dans les deux cas.

On considère ici la loi de probabilité \mathbb{P} correspondant à 100 lancers successifs d'une pièce de monnaie équilibrée, et on introduit la variable aléatoire X comptant le nombre de "pile" à l'issue de ces lancers. On cherche à évaluer la valeur de $\mathbb{E}((X - 60)_+) =: R$, où $a_+ := \max(x, 0)$ désigne la partie positive d'un nombre $a \in \mathbb{R}$. D'après le théorème-limite central, X prendra l'essentiel de ses valeurs entre 40 et 60, de sorte que $(X - 60)_+$ sera le plus souvent nul : la loi \mathbb{P} elle-même est donc un mauvais échantillonnage ! Nous cherchons donc une loi \mathbb{P}' respectant mieux le principe d'échantillonnage préférentiel, qui soit facile à simuler et donc on connaisse facilement la densité *par rapport* à \mathbb{P} . Un exemple de telle loi est la loi correspondant à 100 lancers successifs d'une pièce pipée qui a une probabilité de 60 % de tomber sur "pile" à chaque lancer : en effet, si ω est une série possible de 100 lancers dans laquelle il y a $X(\omega)$ "pile" et $100 - X(\omega)$ "face", on a $\mathbb{P}(\omega) = 1/2^{100}$ contre $\mathbb{P}'(\omega) = 0,6^X(\omega) \times 0,4^{100-X(\omega)}$, d'où

$$\frac{d\mathbb{P}'}{d\mathbb{P}} = \left(\frac{0,6}{0,5}\right)^X \left(\frac{0,4}{0,5}\right)^{100-X}. \quad (28)$$

Le rôle de m est ainsi joué par le membre de droite de (28), et on peut finalement écrire la formule d'échantillonnage préférentiel :

$$R = \mathbb{E}' \left(\frac{(X - 60)_+}{\left(\frac{0,6}{0,5}\right)^X \left(\frac{0,4}{0,5}\right)^{100-X}} \right). \quad (29)$$

✱

Pour finir, voyons un exemple où l'échantillonnage préférentiel est pratiquement indispensable. Dans cet exemple, nous allons essayer de calculer le volume de la boule unité de dimension 16. Naturellement, la méthode présentée à la § I.4.a pour le volume de la 6-boule s'adapte immédiatement à ce cadre ; hélas, les résultats qu'elle donne sont cette fois-ci catastrophiques : en effet, en lançant les calculs pour $N = 100\,000$, je trouve un estimateur égal à... 0 (et avec un intervalle de confiance de largeur nulle, donc complètement faux) !! En d'autres termes, nos 100 000 simulations sont *toutes* tombées en-dehors de la boule... En poussant les calculs jusqu'à $N = 10\,000\,000$ (ce qui commence à prendre du temps, près d'une minute chez moi), je trouve que 33 simulations seulement tombent dans la sphère — ce qui me donne un intervalle de confiance à 95 % égal à $[0,14; 0,30]$, soit guère qu'un ordre de grandeur. Il se trouve en effet qu'en grande dimension, « la sphère a une forme très différente du cube » (dans un sens qui pourrait être rendu rigoureux), ce qui explique la faillite de notre méthode.

Du coup, on cherche plutôt une loi d'échantillonnage dont la densité ressemble suffisamment à $\mathbf{1}_B$... Justement, les gaussiennes isotropes ont une symétrie sphérique, aussi allons-nous essayer de prendre comme loi d'échantillonnage $\mathcal{N}(\frac{1}{16}\mathbf{I}_{16})$ — le facteur d'échelle $\frac{1}{16}$ est choisi de sorte que les points que nous échantillonnerons auront une distance à l'origine moyenne (en moyenne quadratique^[*]) de 1, et seront donc proches du bord de la

[*]. Ce qu'on appelle *moyenne quadratique* d'une v.a. positive X est $\mathbb{E}[X^2]^{1/2}$.

boule, ce qui paraît raisonnable. ^[†] Cette loi est facile à simuler comme 16-uplet de variables $\mathcal{N}(\frac{1}{16})$ indépendantes, et sa densité est tout aussi facile à calculer : $m(\vec{x}) = (8/\pi)^8 \times \exp(-8\|\vec{x}\|_2^2)$. Ainsi programmé avec $N = 10\,000$, je trouve en moins d’une seconde de calcul un intervalle de confiance à 95 % de $[0,232; 0,238]$, soit déjà une précision très acceptable. ^[‡] En termes d’efficacité, le gain sur ma machine est d’un facteur environ 50 000 !

25.a Astuce. L’échantillonnage préférentiel est particulièrement crucial quand la dimension est très grande (donc en particulier en dimension infinie, comme c’est généralement le cas avec les processus) : en très grande dimension, un mauvais échantillonnage peut causer une perte d’efficacité dramatique au point de rendre le calcul non concluant !

I.5.e Manipulations préliminaires à l’échantillonnage préférentiel

Le principe d’échantillonnage préférentiel, nous venons de le voir, peut permettre des gains d’efficacité spectaculaires, en particulier quand la fonction qu’on échantillonne est très hétérogène. Dans cette sous-section, nous allons voir qu’il est parfois possible de procéder à certaines manipulations préliminaires afin de transformer une espérance de fonction peu hétérogène (donc se prêtant mal à l’échantillonnage préférentiel) en une fonction sur laquelle l’échantillonnage préférentiel est beaucoup plus efficace.

Dans la mesure où il n’y pas de véritable théorie derrière cette astuce, nous allons simplement en donner deux exemples pour la présenter.



Dans notre premier exemple, nous reprenons la loi \mathbb{P} du tirage de 100 lancers à pile ou face et la variable X comptant le nombre de “pile”, et notre objectif est d’évaluer $\mathbb{E}(|X - 60|)$. L’intégrande $|X - 60|$ n’est pas particulièrement hétérogène, de sorte qu’on voit mal quel échantillonnage préférentiel on pourrait appliquer à ce cas. Mais maintenant, nous observons que

$$|X - 60| = 2(X - 60)_+ - (X - 60). \quad (30)$$

Or, comme nous connaissons exactement $\mathbb{E}(X) = 50$, nous pouvons en déduire que

$$\mathbb{E}(|X - 60|) = 10 - 2\mathbb{E}(X - 60)_+, \quad (31)$$

ce qui nous ramène à un des exemples de la § I.5.d, pour lequel nous avons vu que l’échantillonnage préférentiel était très efficace !

25.b Remarque. L’idée ci-dessus est proche de la technique de la variable de contrôle qui sera présentée à la § ??, puisque dans les deux cas on soustrait à la fonction à intégrer une fonction auxiliaire d’espérance commune pour améliorer les calculs. Toutefois, il y a une différence importante entre les deux méthodes. Dans la technique de la variable de contrôle, en effet, on utilise la fonction auxiliaire pour réduire la variance de la fonction à intégrer *sans* changer la loi d’échantillonnage. Dans le cas que nous venons de présenter,

[†]. En fait, c’est le *seul* choix raisonnable, car en grande dimension l’essentiel de la masse des boules se situe à proximité immédiate de leurs bords.

[‡]. La valeur exacte est $\pi^8/40\,320 \simeq 0,235\,331$.

en revanche, la fonction auxiliaire *ne réduit pas forcément* la variance de l'intégrande (par exemple, la fonction auxiliaire peut très bien être constante), mais le point est que la nouvelle fonction en laquelle elle la transforme se prête bien à changer la loi pour appliquer l'échantillonnage préférentiel *a posteriori*. Vous observerez en particulier que la technique que nous venons de décrire s'utilise *avant* le choix de la loi d'échantillonnage, alors que la technique de la variable de contrôle s'utilise *après* — et du reste, les deux techniques peuvent très bien être utilisées ensemble dans certains cas !



Dans notre second exemple, nous reprenons le problème du yahtzee de la § I.2.a. On cherche un échantillonnage préférentiel qui favorise les yahtzees. Une idée est de piper les dés pour qu'ils retombent plus souvent sur "6", ce qui augmente effectivement la fréquence d'occurrence des yahtzees de "6", mais cet échantillonnage a en fait un gros écueil car il contribue aussi à notablement sous-représenter des yahtzees de "1", "2", "3", "4" ou "5", de sorte qu'en fait la variance va exploser au lieu de diminuer ! D'où l'idée suivante : on observe que par des arguments de symétrie, la proportion de yahtzees de "6" est en fait un sixième de la proportion de yahtzee tout court,^[§] donc on peut utiliser notre échantillonnage préférentiel pour évaluer la proportion de yahtzee de "6" (cette fois-ci, l'échantillonnage est effectivement adapté), puis multiplier le résultat obtenu par 6 pour évaluer p . Le code correspondant est donné en ??.

I.5.f Échantillonnage interstrates

26.a Remarque. Cette sous-section est intimement liée à la § II.5, qu'il peut donc être utile de lire en même temps que celle-ci.

Supposons qu'on nous demande de calculer une somme $\mathbb{E}_1(f_1) + \mathbb{E}_2(f_2)$ par la méthode de Monte-Carlo, et qu'on ait déjà choisi les lois \mathbb{P}_1 et \mathbb{P}_2 qui serviront à échantillonner respectivement f_1 et f_2 (par exemple d'après le principe d'échantillonnage préférentiel). Il nous reste encore un paramètre à régler : quelle proportion de notre temps de calcul allons-nous consacrer à l'évaluation de $\mathbb{E}_1(f_1)$, et quelle proportion à celle de $\mathbb{E}_2(f_2)$?

Pour clarifier notre présentation, prenons un exemple. Imaginons que le Pape désire connaître avec précisions nombre de catholiques sur l'île d'Irlande. Cette île est divisée en deux parties assez homogènes : l'Irlande du Sud, qui comporte une population totale d'exactement $M_1 = 4\,500\,000$ hab (ce qu'on sait d'après le recensement) et où il y a une proportion p_1 de Catholiques valant *grosso-modo* 90 %, et l'Irlande du Nord, qui comporte exactement $M_2 = 1\,800\,000$ hab et où la proportion p_2 de Catholiques vaut environ 50 %.^[¶] On suppose qu'échantillonner l'Irlande du Sud, resp. du

[§]. Attention, ce n'est pas tout-à-fait vrai dans le programme de l'annexe III.1.a, car celui-ci ne traite pas tous les chiffres de la même façon : en cas d'égalité, il retient en effet le plus petit des chiffres les plus présents, ce qui augmente la proportion de yahtzees de "1" au détriment de ceux de "6". Il faudra donc modifier cet algorithme pour le rendre symétrique : voir annexe ??.

[¶]. NdA : Bien qu'inspirés de la réalité pour des raisons pédagogiques, ces chiffres n'ont de valeur qu'illustrative et ne prétendent aucunement refléter la situation de la véritable Irlande.

Nord, est facile, car il suffit de prendre la première route venue et d'interroger les personnes qu'on y croise [III]. Le temps requis pour procéder au sondage est proportionnel au nombre de personnes interrogées dans les deux régions, mais un peu lent en Irlande du Sud (où l'on croise une personne toutes les $\tau_1 = 3$ min) qu'en Irlande du Nord (où on croise une toutes les $\tau_2 = 2$ min). Dans cet exemple, \mathbb{P}_1 est la loi d'échantillonnage uniforme au Sud, \mathbb{P}_2 est l'échantillonnage uniforme au Nord, et f_1 et f_2 sont respectivement " M_1 fois l'indicatrice d'être catholique" et " M_2 fois l'indicatrice d'être catholique"

En se rappelant que la variance de l'indicatrice d'un événement de probabilité p vaut $p(1-p)$, on calcule alors que l'efficacité de l'évaluation de $\mathbb{E}_1(f_1)$ par la méthode de Monte-Carlo est alors égale à $\mathcal{E}ff_1 = 1/(\tau_1 M_1^2 p_1(1-p_1)) \simeq 1,83 \cdot 10^{-13} \text{ hab}^{-2} \cdot \text{min}^{-1}$ au Sud, contre $\mathcal{E}ff_2 = 1/(\tau_2 M_2^2 p_2(1-p_2)) \simeq 6,17 \cdot 10^{-13} \text{ hab}^{-2} \cdot \text{min}^{-1}$ au Nord. Ce qui nous intéresse, toutefois, est de comprendre l'efficacité *globale* d'un échantillonnage où l'on passerait une proportion α_1 du temps au Nord et une proportion $\alpha_2 = 1 - \alpha_1$ au Sud. Dans ce cas, si on consacre un temps total T à notre sondage, la variance sur l'estimateur de $\mathbb{E}_1(f_1)$ est égale à $1/(\alpha_1 T) / \mathcal{E}ff_1$ et celle sur l'estimateur de $\mathbb{E}_2(f_2)$ est égale à $1/(\alpha_2 T) / \mathcal{E}ff_2$. Comme les deux estimateurs sont indépendants, leurs variances s'ajoutent et la variance totale est ainsi $1/(\alpha_1 T) / \mathcal{E}ff_1 + 1/(\alpha_2 T) / \mathcal{E}ff_2$, d'où une efficacité globale de

$$\mathcal{E}ff = \left((\alpha_1 \mathcal{E}ff_1)^{-1} + ((1 - \alpha_1) \mathcal{E}ff_2)^{-1} \right)^{-1}. \quad (32)$$

Optimiser cette expression en α_1 est un exercice facile : on trouve qu'il faut choisir les α_i proportionnels aux $\mathcal{E}ff_i^{-1/2}$, soit

$$\alpha_1 = \frac{\mathcal{E}ff_1^{-1/2}}{\mathcal{E}ff_1^{-1/2} + \mathcal{E}ff_2^{-1/2}} \simeq 64,8 \%, \quad (33)$$

et donc $\alpha_2 \simeq 35,2 \%$. On peut aussi calculer que l'efficacité globale associée est de $7,67 \cdot 10^{-14} \text{ hab}^{-2} \cdot \text{min}^{-1}$. Ainsi, bien que les Irlandais du Sud constituent 71,4 % de la population (et 81,8 % de la population catholique), on ne passe que 64,8 % du temps à les interroger — et c'est pire encore en termes de nombre de personnes interrogées : les Irlandais du Sud ne constituent que 55,1 % de notre échantillon !

✱

La conclusion ci-dessus est paradoxale, car elle semble contrevenir au principe d'échantillonnage préférentiel. Je m'explique. Notant $M := M_1 + M_2 = 6\,300\,000$ hab la taille totale de la population f la fonction " M fois indicatrice d'être catholique" et \mathbb{P} la loi uniforme sur la population de l'île d'Irlande, la quantité que nous cherchons à calculer peut s'écrire $\mathbb{E}(f)$. L'échantillonnage que nous venons de décrire, à savoir de constituer notre échantillon avec des fractions déséquilibrées de sudistes et de nordistes, est simplement un échantillonnage préférentiel : si nous notons β_1 la fraction du panel constitué par les Sudistes et $\beta_2 = 1 - \beta_1$ celle constituée par les Nordistes, la densité d'échantillonnage est $\beta_1 M/M_1$ au Sud et $\beta_2 M/M_2$ au

[III]. Dans la vraie vie, un tel échantillonnage ne serait évidemment absolument pas indépendant, ni même uniforme !

Sud, ce qui correspond à une loi d'échantillonnage que nous noterons \mathbb{P}_β . La formule de l'efficacité (définition 19.a) donne alors :

$$\mathcal{E}ff_\beta = (\mathbb{E}_\beta(\tau) \times \text{Var}_\beta(f))^{-1}. \quad (34)$$

L'optimisation de cette efficacité en β (sous la contrainte $\beta_1 + \beta_2 = 1$) donne, d'après le théorème 29.a ci-après, un échantillonnage optimal correspondant à $\beta_1 \simeq 76,4$ % de l'échantillon pris au Sud, ce qui est donc au contraire un suréchantillonnage du Sud — et c'est pire encore en termes de temps consacré : 82,9 % au Sud ! Quant à l'efficacité « optimale » que nous obtenons, elle n'est cette fois-ci que de $5,82 \cdot 10^{-14} \text{ hab}^{-2} \cdot \text{min}^{-1} \dots$

Où est l'erreur ? En fait, la situation que nous venons de décrire n'est *pas* équivalente à la première, car nous n'y avons pas tenu compte de l'information dont nous disposions sur la stratification, à savoir les données exactes des recensements. En d'autres termes, dans notre nouvelle situation, la proportion d'habitants du Sud et du Nord dans l'échantillon peut fluctuer aléatoirement autour de sa valeur moyenne, *et on ne rectifie pas ces fluctuations en fonction des données du recensement*. Comme les proportions de Catholiques au Sud et au Nord sont très différentes, ces fluctuations se répercutent sur la proportion globale de Catholiques dans notre échantillon, et on a donc intérêt à les limiter en exagérant l'échantillonnage du Sud. Cela explique aussi que l'efficacité qu'on obtienne soit moins bonne que dans la première situation : ne tenant pas compte de l'information de stratification, on perd nécessairement en variance !

Le paradoxe est donc résolu. On en retiendra que le fait de tenir compte de la stratification change radicalement les critères de l'échantillonnage préférentiel !



On peut immédiatement généraliser l'exemple que nous venons de traiter en la règle suivante :

28.a Théorème. *Soit $\mathbb{E}_1(f_1) + \dots + \mathbb{E}_k(f_k)$ une somme d'espérances qu'on souhaite calculer par la méthode de Monte-Carlo associée aux échantillonnages respectifs $\mathbb{P}_1, \dots, \mathbb{P}_k$, ces échantillonnages étant réalisés indépendamment. Alors l'allocation optimale des ressources de calcul consiste à consacrer au calcul de $\mathbb{E}_i(f_i)$ un temps proportionnel à $\mathcal{E}ff_i^{-1/2}$, et l'efficacité globale vaut alors $(\mathcal{E}ff_1^{-1/2} + \dots + \mathcal{E}ff_k^{-1/2})^{-2}$.*

28.b Remarque.

- (i). Remarquez que nous avons bien précisé que nous nous plaçons dans une situation où les échantillonnages étaient réalisés indépendamment : si on a la possibilité d'appliquer la technique des variables antithétiques (§ II.3), il n'y a donc pas lieu d'utiliser ce théorème.
- (ii). Remarquez également que c'est le *temps de calcul* consacré à chaque échantillonnage qui est proportionnel à $\mathcal{E}ff_i^{-1/2}$, et non pas le nombre de simulations associées à chaque calcul.

28.c Corollaire. Dans le cas particulier où les temps de calcul par simulation sont les mêmes pour tous les $\mathbb{E}_i(f_i)$, le théorème 28.a signifie que le nombre de simulations pour chaque $\mathbb{E}_i(f_i)$ doit être inversement proportionnel à l'écart-type de f_i .

I.5.g Échantillonnage idéal généralisé

Dans cette sous-section nous donnons une généralisation du théorème 20.a qui établit la loi d'échantillonnage optimale en tenant compte :

- D'une contrainte de type stratification, au sens où on ne s'intéresse qu'à un sous-espace vectoriel de l'ensemble des lois d'échantillonnage possibles (ce qui n'est *pas* de la stratification *stricto sensu* dans la mesure où on ne prétend pas connaître la taille exacte de chaque strate) ;
- De l'hétérogénéité éventuelle du temps de calcul par étape en fonction de la strate dans laquelle on tombe (l'effet est rarement marqué pour les simulations par méthode de Monte-Carlo, mais peut devenir important en théorie des sondages).

29.a Théorème. Soit $(\Omega, \mathcal{A}, \mathbb{P})$ un espace probabilisé et \mathcal{B} une sous-tribu de \mathcal{A} . Soit $f: \Omega \rightarrow \mathbb{R}$ une fonction de $L^1(\mathcal{A}, \mathbb{P})$ dont on cherche à déterminer l'espérance par une méthode de Monte-Carlo, où l'on choisit un échantillonnage selon une densité m supposée \mathcal{B} -mesurable.^[*] On suppose qu'une évaluation en ω prend un temps $\dot{\tau}(\omega)$ et on note $\tau := \mathbb{E}(\dot{\tau} \mid \mathcal{B})$; on note également $q := \mathbb{E}(f^2 \mid \mathcal{B})$. On note enfin $I = \mathbb{E}(f)$. Alors l'optimisation de l'efficacité

$$\mathcal{E}ff_m = \left((\mathbb{E}(q \mid m) - I^2) \times \mathbb{E}(\tau m) \right)^{-1} \quad (35)$$

(sous la contrainte $\mathbb{E}(m) = 1$) est atteinte pour

$$m = \left(\frac{q}{I^2 + \alpha \tau} \right)^{1/2}, \quad (36)$$

où α est l'unique constante assurant $\mathbb{E}(m) = 1$.

Démonstration (esquisse). On écrit la dérivée de l'efficacité en utilisant les dérivées logarithmiques, laquelle doit être nulle dès lors que la perturbation est d'espérance nulle : cela impose que le m optimal doit vérifier une formule de type $m = (q / (\beta + \alpha \tau))^{1/2}$. Les calculs donnent par ailleurs une relation entre α et β (relation qui dépend de m elle-même), relation qui se simplifie pour donner $\beta = I^2$ quand on tient compte de l'égalité $\mathbb{E}(m) = 1$. \square

[*]. C'est la tribu \mathcal{B} qui exprime la stratification.

Chapitre II

Techniques de réduction de la variance

II.1 Généralités

Comme nous l'avons vu dans la section précédente, il importe d'avoir la meilleure efficacité possible pour économiser le temps de calcul et obtenir des résultats précis par la méthode de Monte-Carlo. À cette fin, l'échantillonnage préférentiel de la § I.5.c est une première technique tout-à-fait fondamentale. Dans cette section, nous allons présenter d'autres méthodes permettant elles aussi d'améliorer l'efficacité, qui cette fois-ci ne jouent pas sur le choix de la loi d'échantillonnage mais plutôt sur une façon astucieuse de poser les calculs.

L'ensemble de ces techniques (incluant l'échantillonnage préférentiel) sont appelées dans le jargon de Monte-Carlo *techniques de réduction de variance*, puisque l'efficacité d'une méthode de Monte-Carlo est inversement proportionnelle, à temps de calcul fixé, à la variance de l'estimateur fourni par celle-ci.

Précisons d'emblée que rien n'empêche d'utiliser conjointement les diverses techniques de réductions de variance : ces techniques ne sont donc pas concurrentes, mais complémentaires !

II.2 Conditionnement

Pour expliquer la méthode de réduction de variance par conditionnement, nous allons revenir au problème du yahtzee traité dans la § I.2.a. Notons \mathbb{P} la loi décrivant les résultats successifs obtenus par le joueur au cours de ses trois lancers (étant entendu que le joueur suit la stratégie optimale), et X est la variable aléatoire qui vaut 1 si le joueur a obtenu un yahtzee à l'issue de ses trois lancers et 0 sinon, de sorte que p s'écrive $\mathbb{E}(X)$. Maintenant, notons \mathcal{B} la tribu engendrée par les résultats des deux premiers lancers. On peut toujours écrire

$$p = \mathbb{E}(X) = \mathbb{E}(\mathbb{E}(X|\mathcal{B})) \quad (37)$$

d'après les règles sur les espérances conditionnelles. *A priori*, cette réécriture n'apporte aucune information nouvelle. Cependant, il se trouve que nous sommes dans un cas où nous savons calculer facilement $\mathbb{E}(X | \mathcal{B}) =: Y$! En

effet, à l'issue de ses deux premiers lancers, le joueur peut facilement calculer la probabilité conditionnelle qu'il a d'obtenir un yahtzee : s'il a obtenu k dés identiques, il va relancer les $(5 - k)$ autres dés en espérant que ceux-ci tomberont tous sur la valeur des dés qu'il n'a pas relancés, ce qui se produit avec probabilité $(\frac{1}{6})^{5-k}$; on a donc $Y = 6^{K-5}$, où K est la variable aléatoire « nombre de dés identiques obtenus à l'issue des deux premiers lancers », laquelle est bien \mathcal{B} -mesurable.

Puisque $p = \mathbb{E}(Y)$ et que nous savons simuler Y facilement, nous pouvons évaluer p par une méthode de Monte-Carlo que suggère cette écriture. Le code correspondant est donné à l'annexe III.1.b. On constate que l'efficacité de ce nouvel algorithme est bien meilleure que celle de l'algorithme « primitif » : sur ma machine, environ $800\,000\text{ s}^{-1}$ contre $160\,000\text{ s}^{-1}$, soit un gain d'un facteur cinq !

Du point de vue théorique, il y a deux raisons pour lesquelles la méthode évaluant Y est plus efficace que la méthode évaluant X :

- D'une part, la variance de Y est plus petite que la variance de X (et donc, d'après le théorème 18.a, l'efficacité par simulation est améliorée) : cela est assuré par le théorème 31.a sur l'espérance conditionnelle rappelé ci-dessous.
- D'autre part (et dans une moindre mesure), les simulations à effectuer sont plus simples puisqu'on n'a plus que deux lancers à modéliser au lieu de trois.

Ces principes marchent de manière générale dans toute situation de conditionnement. Aussi la méthode de conditionnement améliore-t-elle presque systématiquement l'efficacité (même si pas forcément de façon foudroyante) quand on peut l'appliquer. Nous rappelons ci-dessous le théorème qui justifie que l'efficacité par simulation est en tout cas *toujours* améliorée :

31.a Théorème. *Si X est une v.a. réelle L^2 sur un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$ et que \mathcal{B} une sous-tribu de \mathcal{A} , alors $\text{Var}(\mathbb{E}(X|\mathcal{B})) \leq \text{Var}(X)$, avec égalité (si et) seulement si X est \mathcal{B} -mesurable (quitte à la modifier sur un ensemble de mesure nulle).*

31.b Remarque. Quand l'expression à calculer s'écrit sous forme intégrale, la technique de conditionnement peut encore intervenir de façon déguisée : si on souhaite intégrer une fonction $f(x, y)$ et qu'on connaît $\int f(x, y) dy$ pour tout x , le « conditionnement » consiste alors à écrire la formule de Fubini :

$$\iint f(x, y) dx dy = \int \left(\int f(x, y) dy \right) dx. \quad (38)$$

Il s'agit bien de la même idée, car le conditionnement est en fait une formule du Fubini où le rôle de $dx dy$ est joué par \mathbb{P} , celui de dx par la restriction de \mathbb{P} à \mathcal{B} et celui de dy par la loi conditionnelle de \mathbb{P} connaissant \mathcal{B} , f correspondant alors à X et $\int f(x, y) dy$ à $\mathbb{E}(f | \mathcal{B})$.

II.3 Variables antithétiques

II.3.a Exemple : dilemme rugbystique

Imaginons un match de rugby entre les équipes d'Agen (A) et Brive (B).^[*] Dans le modèle simplifié du rugby que nous prendrons ici, chaque équipe dispose de 12 séquences de jeu au cours du match. À l'issue de chacune de ces séquences, il se peut soit que l'équipe soit repoussée par son adversaire (elle ne marque alors aucun point), soit qu'elle prenne l'avantage. Quand une équipe a pris l'avantage, elle a le choix entre deux options : soit inscrire une pénalité et marquer systématiquement 3 points, soit tenter d'inscrire un essai. Dans ce dernier cas, l'équipe marque 7 points si elle parvient à inscrire l'essai, et aucun sinon.

Dans le match qui nous intéresse, l'équipe B sait qu'elle prendra l'avantage, en moyenne, dans 40 % de ses séquences de jeu. Elle sait aussi que, quand elle choisira de tenter l'essai, elle y réussira dans 35 % des cas. En moyenne, tenter l'essai rapporte donc moins que prendre la pénalité, mais comme la variance est plus importante,^[†] cela introduit un facteur chance plus important qui peut être intéressant face à une équipe plus forte. Justement, l'équipe A est un peu plus forte que l'équipe B , puisqu'elle prend l'avantage dans 60 % de ses séquences de jeu ; quant à sa stratégie, elle est de systématiquement prendre la pénalité dans ce cas (étant donné que pour sa part elle n'a aucun intérêt à augmenter la variance). L'entraîneur de l'équipe B se pose dans la question suivante : quelle stratégie adopter ? Faut-il systématiquement prendre la pénalité ? Tenter l'essai ? Panacher les deux stratégies ?

Pour $q \in [0, 1]$, notons S_q la stratégie consistant, lorsque l'équipe B a pris l'avantage, à tenter l'essai avec probabilité q et à prendre la pénalité sinon ; et notons \mathbb{P}_q la loi de probabilité décrivant le score d'un match où l'équipe B a suivi la stratégie S_q . On note B l'événement "l'équipe B gagne", en supposant qu'en cas d'égalité de points le vainqueur est tiré à la pièce. L'entraîneur cherche donc à maximiser $\mathbb{P}_q(B)$ en fonction de q — disons qu'il veut au moins comparer les cinq stratégies S_q pour $q \in \{0; 0,25; 0,5; 0,75; 1\}$.

Dans un premier temps, l'entraîneur évalue les $\mathbb{P}_q(B)$ par la méthode de Monte-Carlo (les codes sont donnés en ??). Hélas, son ordinateur n'est pas très puissant (on va dire que c'est à cause des restrictions budgétaires...), ce qui fait qu'il ne peut guère monter au-delà de 60 000 simulations. Les intervalles de confiance à 2σ qu'il obtient sont alors les suivants : $\mathbb{P}_0(B) \in [0,162, 0,169]$, $\mathbb{P}_{0,25}(B) \in [0,163, 0,170]$, $\mathbb{P}_{0,5}(B) \in [0,160, 0,167]$, $\mathbb{P}_{0,75}(B) \in [0,154, 0,160]$, $\mathbb{P}_1(B) \in [0,154, 0,161]$. Si cela lui permet d'exclure les stratégies pour $q \geq 0,75$, il n'est en revanche guère avancé pour comparer les valeurs de $q \leq 0,5$, vu que les intervalles de confiance se recouvrent largement...

Mais en fait, réfléchit l'entraîneur, ce qui est intéressant n'est pas tant de calculer les $\mathbb{P}_q(B)$ que leurs *différences* ! N'y aurait-il donc pas un moyen,

[*]. Tout lien avec la ville natale de l'auteur de ce polycopié ne saurait être que purement fortuit...

[†]. Si l'équipe B choisit de systématiquement prendre la pénalité, son espérance de gain en à chaque séquence de jeu est de 1,2 points et sa variance de 2,16 points², alors que si elle tente systématiquement l'essai l'espérance devient 0,98 et la variance environ 5,90.

pour deux stratégies S_q et $S_{q'}$, d'évaluer directement $\mathbb{P}_{q'}(B) - \mathbb{P}_q(B)$ de façon plus efficace ?

✱

L'idée de l'entraîneur est la suivante. Soit \mathbb{P} une loi de probabilité sous laquelle on suppose qu'un certain événement B se produit avec la probabilité $\mathbb{P}_q(B)$ tandis qu'un autre événement B' se produit avec probabilité $\mathbb{P}_{q'}(B)$. Par linéarité de l'espérance, on a alors

$$\mathbb{P}_{q'}(B) - \mathbb{P}_q(B) = \mathbb{E}(\mathbf{1}_{B'}) - \mathbb{E}(\mathbf{1}_B) = \mathbb{E}(\mathbf{1}_{B'} - \mathbf{1}_B). \quad (39)$$

Réaliser ainsi les lois \mathbb{P}_q et $\mathbb{P}_{q'}$ à partir d'une même troisième loi \mathbb{P} est ce qu'on appelle en mathématiques un *couplage* de ces deux lois. L'avantage de cette nouvelle formulation, c'est qu'il y a plusieurs façons de procéder à un tel couplage... Une première façon, la plus évidente, est réaliser les deux lois indépendamment, mais cela reviendrait au même que de calculer séparément $\mathbb{P}_q(B)$ et $\mathbb{P}_{q'}(B)$. L'approche de l'entraîneur consiste plutôt à regarder, sur le scénario d'un *même* match, ce que cela changerait d'utiliser la stratégie S_q ou $S_{q'}$. Autrement dit, on imagine deux scénarios, l'un avec une équipe B et l'autre avec une équipe B' , qui se déroulent de la façon suivante (on supposera ici $q' > q$) :

- Le score de l'équipe A est le même, qu'elle joue contre B ou B' .
- L'équipe B' est repoussée à l'issue d'une séquence de jeu si et seulement si l'équipe B l'est.
- Quand l'équipe B (et donc aussi B') a pris l'avantage à l'issue d'une séquence de jeu :
 - Chaque fois que l'équipe B tente d'aller à l'essai (ce qui a lieu avec une probabilité q), l'équipe B' le tente aussi ; en outre, le succès ou l'échec de cette tentative d'essai est identique pour B et pour B' .
 - Avec probabilité $(q' - q)$, l'équipe B' tente d'aller à l'essai (et le réussit alors dans 35 % des cas, comme d'habitude) tandis que l'équipe B prend la pénalité.
 - Enfin, avec probabilité $(1 - q')$, les équipes B et B' prennent toutes les deux la pénalité.
- Enfin, si à la fin d'un match, aussi bien les équipes B que B' sont à égalité avec l'équipe A , le résultat du tirage à la pièce est le même pour les deux.

Avec ce couplage, se dit l'entraîneur, les phénomènes aléatoires susceptibles d'influencer le résultat seront aussi similaires que possibles entre les scénarios S_q et $S_{q'}$, et par conséquent, le seul niveau où l'aléa jouera réellement concernera uniquement la *différence* entre les deux scénarios : on s'attend donc à ce que la méthode de Monte-Carlo soit plus efficace dans ce cas-là ! De fait, pour 60 000 simulations (voir les codes en ??), la méthode couplée selon l'idée de l'entraîneur donne les résultats suivants : $\mathbb{P}_{0,25}(B) - \mathbb{P}_0(B) \in [+0,001; +0,007]$, $\mathbb{P}_{0,5}(B) - \mathbb{P}_{0,25}(B) \in [-0,008; -0,002]$, $\mathbb{P}_{0,75}(B) - \mathbb{P}_{0,5}(B) \in [-0,010; -0,005]$, $\mathbb{P}_1(B) - \mathbb{P}_{0,75}(B) \in [-0,003; +0,003]$. Cette fois-ci, les résultats sont nets : la meilleure stratégie est $S_{0,25}$! Comme prévu, le couplage a permis de réduire la variance dans des proportions significatives. ^[‡]

[‡]. Pour parler de la variance avant couplage, une remarque s'impose. Déjà, il serait

II.3.b Approche générale

L'argument théorique derrière l'exemple de la sous-section précédente est le théorème immédiat suivant :

34.a Théorème. Soient f et g deux variables aléatoires L^2 sur un même espace probabilisé. Si f et g sont négativement corrélées^[§] (autrement dit si $\text{Cov}(f, g) < 0$), alors il est plus rapide de calculer directement $\mathbb{E}(f + g)$ par la méthode de Monte-Carlo que de calculer $\mathbb{E}(f) + \mathbb{E}(g)$. Plus précisément, la variance par étape est $\text{Var}(f) + \text{Var}(g) - 2|\text{Cov}(f, g)|$ en utilisant le couplage, contre $\text{Var}(f) + \text{Var}(g)$ si on calcule indépendamment $\mathbb{E}(f)$ et $\mathbb{E}(g)$.

34.b Remarque. Outre la gain en variance, la méthode des variables antithétiques permet de gagner en temps de calcul par étape. En effet, quand on fait les évaluations indépendamment chaque étape demande *deux* simulations et deux évaluations de fonctions. Avec les variables antithétiques, on a toujours deux évaluations de fonctions à faire par étape, mais plus qu'une seule simulation ! Par conséquent, on peut gagner jusqu'à un facteur 2 en temps quand c'est la simulation qui prend l'essentiel du temps de calcul plutôt que l'évaluation.

34.c Remarque. En pratique, le plus difficile n'est évidemment pas d'appliquer le théorème 34.a, mais bien de repérer qu'on est dans une situation permettant de s'y ramener !

34.d Exemple. Voici un exemple où les variables antithétiques interviennent de manière inattendue. Supposons qu'on veuille calculer par la méthode de Monte-Carlo le moment exponentiel $M := \mathbb{E}_{\mathcal{N}(1)}(e^X)$.^[¶] Sous la loi $\mathcal{N}(1)$, $-X$ suit la même loi que X par symétrie, donc on a aussi $M = \mathbb{E}_{\mathcal{N}(1)}(e^{-X})$, et par conséquent $M = (\mathbb{E}(e^X) + \mathbb{E}(e^{-X}))/2$. Or les variables e^X et e^{-X} sont clairement négativement corrélées puisqu'elles varient en sens inverses, de sorte qu'il est plus rapide d'évaluer M à partir de l'expression $\mathbb{E}(e^X + e^{-X})/2$ qu'à partir de son expression initiale.^[||]

34.e Remarque. Dans le cas du dilemme rugbystique de la sous-section précédente, la quantité à calculer se présentait sous la forme d'une différence de deux termes très similaires, ce qui suggérait immédiatement l'approche par couplage. Dans ce genre de cas, la méthode des variables antithétique est plus volontiers appelée « méthode de l'aléa commun ».

faux de dire que l'intervalle de confiance sans couplage était de largeur 0,006, puisque cela n'est la largeur de l'intervalle que pour *un* des $\mathbb{P}_q(B)$! Mais *a contrario*, il serait excessif de soustraire naïvement les deux intervalles de confiance et de dire qu'on avait une largeur de 0,012 pour la différence : en effet, quand on ajoute deux v.a. indépendantes ce sont leurs *variances* qui s'ajoutent et non leurs écarts-types, de sorte que l'intervalle de confiance correct associé à la méthode naïve avait une largeur de $(2 \times 0,006^2)^{1/2} \simeq 0,009$.

[§]. D'où l'adjectif *antithétiques*.

[¶]. C'est juste un exemple pédagogique, vu que cette quantité s'évaluerait mieux par intégration déterministe, et qu'on peut même calculer qu'elle vaut exactement \sqrt{e} .

[||]. J'ai implicitement utilisé qu'évaluer M à partir de l'expression $(\mathbb{E}(e^X) + \mathbb{E}(e^{-X}))/2$ prend le même temps qu'à partir de son expression initiale, vu que cela prend pratiquement le même temps d'évaluer e^{-X} que e^X .

34.f Remarque. Dans la méthode de l'aléa commun, la quantité qu'on cherche à évaluer est de la forme $\mathbb{E}(f_\varepsilon) - \mathbb{E}(f_0)$, où f_ε est une fonction proche de f_0 . Si f_ε varie en ε de façon régulière, sous la forme $f_\varepsilon = f_0 + \varepsilon g + o(\varepsilon)$ (on laissera tomber le " $o(\varepsilon)$ " dans la suite), on constate alors que l'efficacité relative de la méthode reste bonne quand $\varepsilon \searrow 0$, puisque l'aléa commun conduit alors à évaluer $\varepsilon \mathbb{E}(g)$ à partir de l'expression $\mathbb{E}(\varepsilon g)$, d'où une efficacité relative constante. (Ce que j'appelle l'*efficacité* relative est l'efficacité multipliée par le carré de l'ordre de grandeur de la quantité à calculer, de façon à avoir une mesure de l'efficacité qui soit invariante quand on multiplie la quantité à évaluer par une constante. Ici on a une efficacité absolue qui croît en ε^{-2} pour une quantité à évaluer proportionnelle à ε , ce qui correspond bien à une efficacité relative constante). *A contrario*, l'efficacité relative de la méthode naïve décroîtrait vers 0 en ε^2 ... On en retiendra que dans ce genre de cas, la méthode de l'aléa commun permet donc de comparer efficacement deux espérances $\mathbb{E}(f_\lambda)$ pour des paramètres λ arbitrairement proches, ce qui est utile pour calculer avec précision le paramètre qui optimise $\mathbb{E}(f_\lambda)$.^[*]

35.a Remarque. Théoriquement, il est possible, de manière symétrique, d'imaginer une méthode des « variables prothétiques » : si f et g sont *positivement* corrélées, il y a moins de variance à calculer séparément $\mathbb{E}(f) + \mathbb{E}(g)$ qu'à calculer directement $\mathbb{E}(f + g)$... En pratique cependant, cette idée est presque toujours mauvaise, car la remarque 34.b nous dit que le gain en variance d'une telle manipulation a toutes les chances d'être compensé par la perte de temps par étape de calcul causée par une telle division des tâches dès lors que la simulation prend une part non négligeable du temps de calcul par rapport à l'évaluation de f et g . En outre, le gain en variance ne saurait de toutes façons excéder un facteur 2 dans ce sens-là, alors qu'il pouvait être arbitrairement grand dans l'autre sens.

II.4 Variables de contrôle

II.4.a Variable de contrôle unique

L'idée de la méthode de réduction de variance par utilisation de variables de contrôle est la suivante. *In fine*, une méthode de Monte-Carlo se ramène au calcul d'une espérance $\mathbb{E}(f(X))$ par échantillonnage de X selon la loi \mathbb{P} , l'efficacité de la méthode étant inversement proportionnelle à (entre autres) $\text{Var}(f)$. Maintenant, supposons qu'il existe une autre fonction g pour laquelle nous sachions calculer exactement $\mathbb{E}(g(X))$. En écrivant $\mathbb{E}(f) = \mathbb{E}(f - \beta g) + \beta \mathbb{E}(g)$, où β est un paramètre à fixer ultérieurement, on voit alors qu'on peut ramener l'évaluation $\mathbb{E}(f)$ par la méthode de Monte-Carlo à l'évaluation de $\mathbb{E}(f - \beta g)$, le passage de l'une à l'autre se faisant en temps négligeable. Pour peu qu'une étape de simulation pour $(f - \beta g)$ ne prenne guère plus de temps que pour f (ce qui est généralement le cas si l'expression de g n'est pas trop compliquée) et que la variance de $(f - \beta g)$ soit sensiblement plus petite que celle de f , cette nouvelle méthode sera plus rapide que la

[*]. Par contre, cela ne marche pas dans les cas où l'évolution de f_λ se fait par sauts comme dans l'exemple rugbystique de la sous-section précédente : dans ce cas, l'efficacité relative décroît en effet vers 0 en ε — ce qui est tout de même mieux que le ε^2 de la méthode naïve.

précédente. Dans le jargon des méthodes de Monte-Carlo, la v.a. $g(X)$ est alors appelée de *variable de contrôle*. Pour trouver la valeur de β qui minimise $\text{Var}(f - \beta g)$,^[†] on dispose du théorème suivant, dont la démonstration est immédiate :

36.a Théorème. $\text{Var}(f - \beta g)$ est minimal pour

$$\beta = \frac{\text{Cov}(f, g)}{\text{Var}(g)}, \quad (40)$$

et vaut alors

$$(1 - |\text{Corr}(f, g)|^2) \times \text{Var}(f), \quad (41)$$

où $\text{Corr}(f, g)$ est le coefficient de corrélation de Pearson entre f et g , c.à.d. $\text{Corr}(f, g) := \text{Cov}(f, g) / \text{Var}(f)^{1/2} \text{Var}(g)^{1/2}$.

36.b Remarque. Rappelons que le coefficient de corrélation de Pearson est, en valeur absolue, toujours compris entre 0 et 1, qu'il vaut 0 entre autres si f et g sont indépendantes, et qu'il vaut 1 si et seulement s'il existe une relation déterministe affine entre f et g .

36.c Remarque. On voit dans la formule (41) que la réduction de variance par variable de contrôle n'est intéressante que si la corrélation de cette variable avec la fonction à intégrer est (en valeur absolue) assez proche de 1 : ainsi, pour une corrélation de 1/2, le gain de variance n'est déjà plus que d'un quart, et pour une corrélation de 1/4, il n'est que d'un seizième et ne compense alors sans doute même plus l'augmentation du temps de calcul par étape...

36.d Astuce. D'après le théorème 36.a, la détermination du β optimal, noté ici β^{opt} , demande d'évaluer $\text{Cov}(f, g)$ ainsi que $\text{Var}(g)$: pour ce faire, on peut procéder... encore et toujours par la méthode de Monte-Carlo ! De même que pour l'évaluation de l'efficacité (cf. fin de la § I.5.a), cela peut se faire rapidement car il n'est pas nécessaire d'avoir une valeur très précise de β^{opt} en pratique : en prenant pour β un estimateur de β^{opt} obtenu par quelques milliers de simulations, la valeur de $\text{Var}(f - \beta g)$ sera en général très proche de celle obtenue pour le vrai β^{opt} .



36.e Exemple.

II.4.b Variables de contrôle multiples

D'un point de vue abstrait, la technique de la variable de contrôle du théorème 36.a peut être reformulée ainsi : on réduit la variance de f en projetant cette variable sur l'orthogonal de la droite $\mathbb{R}g$ dans $L^2(\mathbb{P})$. Ainsi réécrit, on s'aperçoit qu'il n'y a aucune raison pour projeter f sur l'orthogonal d'une *droite*, et qu'on peut prendre en fait à la place de $\mathbb{R}g$ n'importe quel sous-espace de dimension finie. Cela donne la procédure suivante :

36.f Mémorandum.

[†]. Le temps de calcul ne dépend pratiquement pas de β dès lors que β est non nul, ce qui fait que l'optimisation de l'efficacité équivaut à la minimisation de $\text{Var}(f - \beta g)$.

- ☛ Pour réduire la variance d'une v.a. $f \in L^2$ (dont on cherche l'espérance) à partir d'une ou plusieurs variables de contrôle g_1, \dots, g_k dont on connaît déjà exactement l'espérance, le principe est d'écrire :

$$\mathbb{E}(f) = \mathbb{E}(f - \beta_1 g_1 - \beta_2 g_2 - \dots - \beta_k g_k) + \beta_1 \mathbb{E}(g_1) + \beta_2 \mathbb{E}(g_2) + \dots + \beta_k \mathbb{E}(g_k). \quad (42)$$

- ☛ Il faut choisir les β_i de façon à minimiser la variance de $(f - \beta_1 g_1 - \dots - \beta_k g_k)$. Les valeurs optimales des β_i sont données par le théorème ?? ci-dessous.
- ☛ Il n'est pas nécessaire de connaître la valeur exacte des β_i optimaux, mais juste de les évaluer avec une précision suffisante : cela se fait en calculant (43) par la méthode de Monte-Carlo avec quelques milliers de simulations.

37.a Théorème. Supposons que la famille $(1, g_1, \dots, g_k)$ est libre dans $L^2(\mathbb{P})$ et notons Σ_g la matrice de covariance $((\text{Cov}(g_i, g_j)))_{1 \leq i, j \leq k}$, qui est alors inversible. Notons également R_{fg} le vecteur $(\text{Cov}(f, g_i))_{1 \leq i \leq k}$. Alors la valeur de $\vec{\beta} := (\beta_1, \dots, \beta_k)$ qui minimise $\text{Var}(f - \beta_1 g_1 - \dots - \beta_k g_k)$ est donnée par

$$\vec{\beta} = \Sigma_g^{-1} R_{fg}, \quad [4] \quad (43)$$

et la variance correspondante est

$$\text{Var}(f) - R_{fg}^T \Sigma_g^{-1} R_{fg} = \left(1 - \frac{R_{fg}^T \Sigma_g^{-1} R_{fg}}{\text{Var}(f)}\right) \times \text{Var}(f). \quad (44)$$

37.b Remarque. La quantité $R_{fg}^T \Sigma_g^{-1} R_{fg} / \text{Var}(f)$ qui apparaît dans (44) peut être vue comme le carré du “coefficient de corrélation multidimensionnel” entre f et les g_i : plus précisément, $(R_{fg}^T \Sigma_g^{-1} R_{fg} / \text{Var}(f))^{1/2}$ est le cosinus de l'angle entre la droite $\mathbb{R}f$ et le sous-espace $\mathbb{R}g_1 + \dots + \mathbb{R}g_k + \mathbb{R}\mathbf{1}$ dans $L^2(\mathbb{P})$. En particulier, cette quantité est toujours comprise entre 0 et 1, elle vaut 0 entre autres si f et g sont indépendantes, et elle vaut 1 si et seulement s'il existe une relation déterministe affine entre f et les g_i .

II.4.c Exemple

Voyons comment nous pouvons appliquer la technique des variables de contrôle pour le problème de la course à pied. On rappelle qu'on y considère 8 chronos X_1, \dots, X_8 i.i.d. $\text{ln}\mathcal{N}(4; 0,03)$ et qu'on cherche à évaluer $\mathbb{E}(\max_i X_i)$ — on notera dans la suite $\max_i X_i =: X_*$. Je propose d'introduire les trois variables de contrôle suivantes :

$$Y_1 := \mathbf{1}_{X_* \leq 65}; \quad (45)$$

$$Y_2 := \mathbf{1}_{X_* \leq 70}; \quad (46)$$

$$Y_3 := \mathbf{1}_{X_* \leq 75}. \quad (47)$$

[4]. Attention : dans la formule (43), j'ai pris le formalisme usuel en mathématiques consistant à dire qu'un “vecteur” est un vecteur-colonne ; mais *MATLAB* a plutôt tendance à manipuler des vecteurs-lignes : avec ce formalisme-là, la formule deviendrait « $\vec{\beta} = R_{fg} \Sigma_g^{-1}$ ».

Il est clair que ces deux variables de contrôle seront fortement (négativement) corrélées à X_* . Ce qui est moins évident, c'est de comprendre en quoi les espérances de celles-ci sont faciles à calculer : le point est qu'on a

$$\begin{aligned}\mathbb{E}(\mathbf{1}_{X_*} \leq t) &= \mathbb{P}(\max_i X_i \leq t) = \prod_{i=1}^8 \mathbb{P}(X_i \leq t) \\ &= \mathbb{P}(\ln \mathcal{N}(4; 0,03) \leq t)^8 = \mathbb{P}(\mathcal{N}(1) \leq (\ln t - 4)/0,03^{1/2})^8 \\ &= \left[\frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{\ln t - 4}{\sqrt{2} \times 0,03^{1/2}} \right) \right]^8, \quad (48)\end{aligned}$$

et comme les logiciels comme *MATLAB* savent calculer très précisément la fonction *erf*, on en déduit les valeurs exactes (à l'arrondi près) suivantes :

$$\mathbb{E}(Y_1) = 0,255\,024\,592; \quad (49)$$

$$\mathbb{E}(Y_2) = 0,762\,034\,518. \quad (50)$$

Maintenant, à l'aide d'une série de 2 000 simulations (voir annexe ??), je trouve les corrélations suivantes : [§]

$$\operatorname{Cov}(X, \vec{Y}) \simeq \begin{pmatrix} -2,38 \\ -2,54 \end{pmatrix} \text{ et } \operatorname{Var}(\vec{Y}) \simeq \begin{pmatrix} 0,200 & 0,064\,7 \\ 0,064\,7 & 0,178 \end{pmatrix}. \quad (51)$$

D'après (43), on en déduit que $\vec{\beta}^{\text{opt}} \simeq (-8,30, -11,2)$; nous prendrons donc $(\beta_1, \beta_2) = (-8, -11)$. L'algorithme correspondant est donné en ??. Avec l'utilisation des variables de contrôle, l'efficacité passe chez moi d'environ $1,1 \text{ s}^{-1}$ à $3,5 \text{ s}^{-1}$, soit un gain d'un facteur 3 !

II.5 Stratification

38.a Remarque. Toute cette section est intimement liée à la § I.5.f, qu'il peut donc être utile de lire en même temps que celle-ci.

II.5.a Approche théorique

Imaginons qu'on cherche à évaluer par la méthode de Monte-Carlo une expression de la forme $\mathbb{E}(f)$, où \mathbb{P} est la loi d'échantillonnage et $f \in L^2(\mathbb{P})$ la quantité à échantillonner ; et supposons donnée une partition de l'espace de probabilité Ω en un nombre fini d'événements A_1, \dots, A_k (on rappelle que « partition » signifie que les A_i sont deux à deux disjoints et que leur réunion est Ω tout entier) pour laquelle on connaît exactement la valeur de chaque $\mathbb{P}(A_i)$, notée p_i . On suppose en outre que les p_i sont tous non nuls.

Pour chaque $i \in \{1, \dots, k\}$, définissons \mathbb{P}_i comme la loi conditionnelle de \mathbb{P} sous l'événement A_i , c.à.d. $\mathbb{P}_i(B) := \mathbb{P}(B \cap A_i) / \mathbb{P}(A_i)$, de sorte que $\mathbb{P} = \sum_i p_i \mathbb{P}_i$. On peut alors écrire $\mathbb{E}(f) = \sum_i p_i \mathbb{E}_i(f_i)$.

Maintenant, supposons qu'au lieu d'échantillonner « naïvement » f sous la loi \mathbb{P} , on en fasse des échantillonnages séparés conditionnellement à chacun

[§]. Noter qu'en l'occurrence, la forme des variables Y_i permettrait de calculer exactement $\operatorname{Var}(\vec{Y})$ à partir des $\mathbb{E}(Y_i)$, mais nous ferons comme si nous n'avions pas remarqué cela afin de mieux illustrer notre propos.

des A_i . Plus précisément, au lieu d'estimer directement f par l'estimateur \hat{m} issu de N simulations i.i.d. sous \mathbb{P} , on réalise plutôt N_i simulations de chaque \mathbb{P}_i (avec $\sum_i N_i = N$), puis on estime chaque $\mathbb{E}_i(f)$ par l'estimateur \hat{m}_i issu de ces simulations, et on en déduit finalement un estimateur $\hat{m}^{\text{strat}} := \sum p_i \hat{m}_i$ pour $\mathbb{E}(f)$. Les N_i sont choisis de façon que $N_i \stackrel{N \rightarrow \infty}{\sim} \beta_i N \forall i$, où les $\beta_i, i \in \{1, \dots, k\}$ sont une famille de paramètres strictement positifs et de somme 1 — qui peuvent être égaux aux p_i , mais pas forcément. Notez que l'hypothèse de non-nullité des β_i implique que, pour N assez grand, tous les N_i sont non nuls, de sorte que a bien un sens d'« estimer $\mathbb{E}_i(f)$ d'après N_i simulations » : dans la suite, on supposera toujours implicitement que c'est effectivement le cas.

Quelle est la variance de ce nouvel estimateur ? Puisque \hat{m}^{strat} s'obtient comme somme de k estimateurs indépendants, sa variance est la somme des variances de chacun de ces estimateurs :

$$\text{Var}(\hat{m}^{\text{strat}}) = \sum_{i=1}^k \text{Var}(\hat{m}_i) = \sum_i \frac{\text{Var}_i(f)}{N_i}. \quad (52)$$

Cette variance est asymptotiquement proportionnelle à N^{-1} , l'inverse de la constante de proportionnalité définissant l'efficacité par simulation de la méthode stratifiée :

39.a Théorème. *L'efficacité par simulation de la méthode stratifiée est $\text{Eff}^{\text{strat}} = 1 / \text{Var}^{\text{strat}}(f)$, où*

$$\text{Var}^{\text{strat}}(f) = \sum_i \frac{q_i^2}{\beta_i} \text{Var}_i(f). \quad (53)$$

La formule (53) permet de calculer les β_i qui optimisent l'efficacité par simulation : il faut prendre $\beta_i \propto q_i \text{Var}_i(f)^{1/2}$.^[¶] En pratique toutefois, ce n'est pas l'efficacité *par simulation* que nous cherchons à optimiser, mais l'efficacité *tout court*. C'est le problème de l'échantillonnage *interstrates*, qui a déjà été traité par le théorème 28.a dans la § I.5.f. Je rappelle ce théorème par souci de complétude :

39.b Théorème (Reformulation du théorème 28.a). *Les β_i qui optimisent l'efficacité tout court sont donnés par*

$$\beta_i \propto \tau_i^{-1/2} q_i \text{Var}_i(f)^{1/2} \quad (54)$$

(sous la contrainte $\sum_i \beta_i = 1$), soit $\beta_i = \tau_i^{-1/2} q_i \text{Var}_i(f)^{1/2} / \Rightarrow$
 $\Rightarrow (\sum_{j=1}^k \tau_j^{-1/2} q_j \text{Var}_j(f)^{1/2})$, où τ_i est le temps moyen pris par une simulation (incluant l'évaluation de f) sous \mathbb{P}_i .

Mais revenons un instant à l'efficacité par simulation. Comme nous le verrons, dans nombre de situations on est en fait forcé de prendre les β_i égaux aux q_i . Le théorème suivant nous dit que dans ce cas, l'efficacité par simulation est malgré tout toujours meilleure que si on n'a pas stratifié :

[¶]. Je ne donne pas la démonstration de cette formule dans la mesure où, comme on le fait remarquer juste après, le théorème 28.a est plus utile en pratique... Cela dit, pour information, le calcul des β_i « optimaux par nombre de simulation » est complètement similaire à la démonstration du théorème 28.a.

39.c Théorème. Soit \bar{f} la fonction définie par $\bar{f} = \mathbb{E}_i(f)$ sur chaque A_i ; autrement dit \bar{f} est l'espérance de f conditionnellement à la tribu engendrée par les A_i . Alors, pour $\beta_i = q_i \forall i$,

$$\text{Var}(f) = \text{Var}(\bar{f}) + \text{Var}^{\text{strat}}(f); \quad (55)$$

en particulier on a toujours $\text{Var}^{\text{strat}}(f) \leq \text{Var}(f)$, avec égalité si et seulement si \bar{f} est constante.

40.a Remarque. Les deux termes du membre de droite de (55) sont respectivement appelés *variance interstrates* (car c'est la variance qui reste quand on remplace f par sa moyenne sur chaque strate) et *variance intrastrates* (car ce terme ne dépend que des $\text{Var}_i(f)$ et pas de l'écart de \bar{f} d'une strate à l'autre).

Démonstration. On calcule les trois termes impliqués dans (55) :

1. Par définition, $\text{Var}(f) = \mathbb{E}(f^2) - \mathbb{E}(f)^2$.
2. Vu que \bar{f} vaut $\mathbb{E}_i(f)$ sur l'ensemble A_i , on a $\text{Var}(\bar{f}) = \mathbb{E}(\bar{f}^2) - \mathbb{E}(\bar{f})^2 = \sum_i q_i \mathbb{E}_i(f)^2 - (\sum_i q_i \mathbb{E}_i(f))^2 = \sum_i q_i \mathbb{E}_i(f^2) - \mathbb{E}(f)^2$.
3. Comme $\beta_i = q_i$, $\text{Var}^{\text{strat}}(f)$ se réécrit comme $\sum_i q_i \text{Var}_i(f)$, et vu que par définition $\text{Var}_i(f) = \mathbb{E}_i(f^2) - \mathbb{E}_i(f)^2$, on en tire que $\text{Var}^{\text{strat}}(f) = \sum_i q_i \mathbb{E}_i(f^2) - \sum_i q_i \mathbb{E}_i(f)^2 = \mathbb{E}(f^2) - \sum_i q_i \mathbb{E}_i(f)^2$.

Il est alors immédiat de vérifier la formule. \square

40.b Remarque. Ce qui permet à la méthode de stratification ce gain d'efficacité est le fait qu'on connaisse *exactement* la valeur des q_i : il ne faut donc utiliser la méthode que dans ce cas ! On pourrait se dire qu'on pourrait appliquer la stratification en évaluant les q_i par la méthode de Monte-Carlo, mais cette méthode reviendrait en fait à un échantillonnage non stratifié... C'est précisément l'erreur sur l'évaluation des q_i qui explique que l'échantillonnage non stratifié soit moins efficace : pour plus de détails, voir la discussion de la § I.5.f, page 27.

II.5.b La stratification en pratique

Il y a deux types de situations dans lesquelles on peut utiliser la stratification :

- On peut avoir un problème naturellement stratifié, c'est-à-dire que la probabilité d'échantillonnage \mathbb{P} s'écrit naturellement sous la forme $\sum q_i \mathbb{P}_i$ pour des probabilités \mathbb{P}_i deux à deux étrangères — par « naturellement », j'entends ici que la façon la plus efficace de simuler \mathbb{P} consiste à tirer i avec probabilité q_i puis à faire une simulation de \mathbb{P}_i . C'est ce type de situation qu'on avait dans le problème du recensement des Catholiques irlandais de la § I.5.f. Dans ce cas, pas d'hésitation : il faut choisir les q_i selon l'échantillonnage interstrates optimal. Ce genre de cas, à vrai dire, ne peut pas vraiment être qualifié de « stratification » puisque ce serait plutôt de ne pas tenir compte de la structure stratifiée du problème qui demanderait un effort...

- Plus fréquemment, on dispose d’informations sur la taille des strates, mais on n’a pas vraiment d’autre moyen d’échantillonner les \mathbb{P}_i qu’à partir de la simulation de \mathbb{P} ... Cela n’empêche pas d’utiliser la stratification malgré tout, mais la façon de l’utiliser dépendra du rapport de force entre les parties « simulation » et « évaluation » de la méthode de Monte-Carlo :
- Si c’est la simulation qui prend l’essentiel du temps, il faut utiliser toutes les simulations effectuées ; par conséquent, ce qu’on fait est de réaliser N simulations de \mathbb{P} et d’en déduire les N_i en fonction des A_i dans lesquels les simulations sont tombées : [||] contrairement à ce qu’il pourrait sembler, cela reste bien de la stratification car on utilise l’information sur les q_i dans la construction de l’estimateur. Dans ce cas, les β_i sont forcément égaux aux q_i .
- Si au contraire, la simulation (incluant la détermination de la strate dans laquelle on est tombé) prend un temps négligeable par rapport à l’évaluation de f , la situation est en pratique équivalente au cas naturellement stratifié, vu qu’il n’est pas plus coûteux de simuler les \mathbb{P}_i que \mathbb{P} .
- Mais dans la plupart des cas, chaque partie de la méthode prend un certain temps... Dans ce cas, on peut essayer de s’approcher de l’échantillonnage interstrates optimal en décidant de lancer ou non l’évaluation de f et fonction de la strate dans laquelle on se trouve et du nombre d’échantillons déjà disponibles dans cette strate. [*] La différence avec le cas de l’échantillonnage naturellement stratifié est qu’ici l’expérimentateur *ne décide pas* à l’avance dans quelle strate il va tomber.

II.6 Optimisation du code

Cela n’a rien de spécifique aux méthodes de Monte-Carlo, mais rappelons quand même qu’un code mal conçu peut faire perdre un temps considérable quand il s’agit d’implémenter un algorithme, comparé à un autre code exprimant la même idée sémantique mais dont les détails sont pensés de sorte à privilégier les voies de calcul que la machine exécute le plus rapidement.

L’optimisation du code est une science pratique plus que théorique, dans la mesure où dépend en grande partie de la machinerie de programmation utilisée, [†] ainsi que de l’architecture du processeur (ou de l’ensemble de processeurs) sur lequel sont effectués les calculs. Dans le cas d’un langage

[||]. Présenté comme tel, ce protocole a le défaut qu’il se peut qu’un des N_i soit accidentellement nul, auquel cas on ne peut pas évaluer \hat{m}^{strat} . Pour pallier cet inconvénient, on peut, quitte à augmenter la valeur de N , attendre que chaque N_i soit au moins égal à 1 — ou même que chaque N_i soit au moins égal à $q_i N$ si on veut aussi éviter le risque qu’un N_i trop petit crée de grandes fluctuations de l’estimateur. Cela ne diminue pas l’efficacité (en tout cas pas asymptotiquement), puisque les N_i sont équivalents à $q_i N$ pour les grandes valeurs de N par la loi des grands nombres.

[*]. En fait, c’est encore un peu plus compliqué, car l’échantillonnage interstrates optimal dans cette méthode n’est pas exactement celui donné par la formule (54) : en fonction du temps relatif pris par la simulation et par l’évaluation, il va plus ou moins s’approcher de l’échantillonnage « naturel » $\beta_i = q_i$ ou au contraire de l’échantillonnage de (54).

[†]. C’est à dessein que je parle de *machinerie* et non pas de *langage* : le même langage de programmation peut en effet être traduit de façon fort différente au niveau de la machine en

interprété comme *MATLAB*, l'optimisation est particulièrement importante puisqu'aucun compilateur ne fera lui-même les raccourcis faciles que vous auriez oublié de prendre en rédigeant votre code.

Je vous présenterai quelques exemples concrets d'optimisation de code à l'occasion des TP, lorsque nous serons amenés à en rencontrer.

II.7 Synthèse

fonction de l'interpréteur (ou du compilateur) retenu, et les problématiques d'optimisation de code seront alors elles aussi différentes.

Chapitre III

Annexe : Codes *MATLAB*

III.1 Les codes pour le problème du yahtzee

III.1.a Algorithme « naïf »

```
function yahtzee(N)
% La fonction "yahtzee" évalue la probabilité de faire un yahtzee en trois
% lancers par la méthode de Monte-Carlo. Elle prend en paramètre le nombre
% N de simulations demandées.

% La variable "succes" comptera le nombre de fois qu'on a réussi à obtenir
% un yahtzee.
succes = 0;
% Les 2 lignes suivantes servent seulement au calcul de l'efficacité.
sommecarres = 0;
tic;
% On lance une boucle de N expériences. La variable "i" numérote
% l'expérience qu'on est en train de réaliser.
for i = 1:N,
    % On fait jouer le joueur et on compte son succès éventuel. La fonction
    % "troislancers" est une sous-routine qui simule le comportement du
    % joueur effectuant ses trois lancers de dés en cherchant à obtenir un
    % yahtzee. Elle renvoie 1 si l'expérience est un succès et 0 si c'est
    % un échec.
    x = troislancers;
    succes = succes+x;
    % Décommenter la ligne suivante pour calculer l'efficacité %
    sommecarres = sommecarres+x*x; %
end
% La ligne suivante sert seulement au calcul de l'efficacité.
t = toc;
disp('Estimateur de la probabilité de succes :');
disp(succes/N);
% Commentez les 3 lignes suivantes si vous ne souhaitez pas calculer
% l'efficacité.
disp('Efficacité :');
moyenne = succes/N; variance = sommecarres/N-moyenne*moyenne; %
disp(N/t/variance); %
end
```

```

function resultat = troislancers
% La fonction "troislancers" simule le comportement du joueur effectuant
% ses trois lancers de dés en cherchant à obtenir un yahtzee. Elle renvoie
% 1 si l'expérience est un succès et 0 si c'est un échec.

% On fait le premier lancer. La variable "des" est une série de cinq
% éléments représentant le chiffre affiché par chacun des cinq dés.
des = randi([1,6],[1,5]);
% On relance les dés, deux fois. La fonction "relancer" est une
% sous-routine qui, à partir des chiffres affichés par les dés sur la
% table, choisit quels dés le joueur relance et renvoie le nouvel affichage
% des dés après ce lancer.
des = relancer(des);
des = relancer(des);
% On teste si tous les dés ont la même valeur.
resultat = all(des == des(1));
end

function desrelances = relancer(des)
% La fonction "relancer" choisit quels dés le joueur relance et renvoie le
% nouvel affichage des dés après ce lancer. Elle prend en paramètre la
% liste des valeurs affichées.

% La variable "occurrences" est une série de six nombres destinée à
% contenir le nombre d'occurrences de chaque chiffre pour les dés.
% Autrement dit, occurrences(1) comptera le nombre de "1" présents sur la
% table, occurrences(2) comptera le nombre de "2", etc.
occurrences = zeros(1,6);
% On lance une boucle pour remplir la série "occurrences". La variable "i"
% est le numéro du dé qu'on est en train de regarder.
for i = 1:5,
    occurrences(des(i)) = occurrences(des(i))+1;
end
% La variable "recordman" est le chiffre qui apparaît le plus souvent sur
% la table. Pour la trouver, on utilise la syntaxe à deux sorties de la
% fonction "max".
[~, recordman] = max(occurrences);
% On va maintenant donner la série des dés relancés. À l'initialisation,
% tous les dés sont comme ils étaient au départ.
desrelances = des;
for i = 1:5,
    % On relance le dé seulement si le chiffre qu'il affichait n'était pas
    % le recordman.
    if des(i) ~= recordman
        desrelances(i) = randi([1,6]);
    end
end
end
end

```

III.1.b Algorithme avec conditionnement

```

function yahtzeecondi(N)
% La fonction "yahtzeecondi" évalue la probabilité de faire un yahtzee en
% trois lancers par la méthode de Monte-Carlo, en utilisant la méthode de

```

```

% conditionnement à l'issue du deuxième lancer. Elle prend en paramètre le
% nombre N de simulations demandées.

% La variable "succes" de la méthode naïve est remplacée ici par la
% variable "succescondi" qui est son analogue conditionnel.
succescondi = 0;
sommecarres = 0;
% On lance une boucle de N expériences. La variable "i" numérote
% l'expérience qu'on est en train de réaliser.
tic;
for i = 1:N,
    % On fait jouer le joueur et on compte son espérance de succès à
    % l'issue de ses deux premiers lancers. La fonction "deuxlancers" est
    % une sous-routine qui simule le comportement du joueur effectuant ses
    % deux premiers lancers de dés. Elle renvoie la probabilité
    % conditionnelle qu'il obtienne un yahtzee à l'issue du troisième
    % lancer.
    y = deuxlancers();
    succescondi = succescondi+y;
    sommecarres = sommecarres+y*y;
end
t = toc;
disp('Nouvel estimateur de la probabilité de succes :');
disp(succescondi/N);
disp('Nouvelle efficacité :');
moyenne = succescondi/N; variance = sommecarres/N-moyenne*moyenne;
disp(N/t/variance);
end

function resultatcondi = deuxlancers()
% La fonction "troislancers" simule le comportement du joueur effectuant
% ses deux premiers lancers de dés. Elle renvoie la probabilité
% conditionnelle qu'il obtienne un yahtzee à l'issue du troisième lancer.

% On fait le premier lancer. La variable "des" est une série de cinq
% éléments représentant le chiffre affiché par chacun des cinq dés.
des = randi([1,6],[1,5]);
% On relance les dés, une seule fois.
des = relancer(des);
% On compte le plus grand nombre de fois qu'un chiffre apparaît, stocké
% dans la variable "k". Pour ce faire, on commence par établir la série des
% occurrences de chaque chiffre, comme dans la sous-routine "relancer".
occurrences = zeros(1,6);
for i = 1:5,
    occurrences(des(i)) = occurrences(des(i))+1;
end
k = max(occurrences);
% On renvoie la probabilité de succès conditionnellement à l'issue du
% deuxième lancer, soit  $6^{-5+k}$ .
resultatcondi = 6^(-5+k);
end

function desrelances = relancer(des)

```

```

occurrences = zeros(1,6);
for i = 1:5,
    occurrences(des(i)) = occurrences(des(i))+1;
end
[~, recordman] = max(occurrences);
desrelances = des;
for i = 1:5,
    if des(i) ~= recordman
        desrelances(i) = randi([1,6]);
    end
end
end
end

```

III.1.c Algorithme avec échantillonnage préférentiel

```

function yahtzeepref(lambda,N)
% La fonction "yahtzeepref" évalue la probabilité de faire un yahtzee en
% trois lancers par la méthode de Monte-Carlo avec échantillonnage
% préférentiel. lambda est la paramètre de pipage, compris entre 0 et 1,
% tel que "6" soit tiré avec probabilité (1+5*lambda)/6 et les autres
% chiffres avec probabilité (1-lambda)/6. N est le nombre de simulations
% demandé.

somme = 0;
sommecarres = 0;
tic;
% On lance une boucle de N expériences. La variable "i" numérote
% l'expérience qu'on est en train de réaliser.
for i = 1:N,
    six = 0;
    nonsix = 0;
    % On fait jouer le joueur et on compte son succès éventuel. La fonction
    % "troislancers" est une sous-routine qui simule le comportement du
    % joueur effectuant ses trois lancers de dés en cherchant à obtenir un
    % yahtzee. Elle renvoie 1 si l'expérience est un succès et 0 si c'est
    % un échec.
    troislancers;
    densite = (1+5*lambda)^six*(1-lambda)^nonsix;
    x = resultat/densite;
    somme = somme + x;
    sommecarres = sommecarres+x*x; %
end
t = toc;
% Ne pas oublier qu'on vient d'évaluer la probabilité d'un yahtzee de "6",
% donc il faut multiplier le résultat obtenu par 6.
moyenne = 6*somme/N;
variance = 6*6*sommecarres/N-moyenne*moyenne;
disp('Intervalle de confiance à 2 sigmas :');
disp(moyenne+sqrt(variance/N)*[-1,1]);
disp('Efficacité :');
disp(N/t/variance);

% Attention ! Ici il faut mettre les sous-routines /à l'intérieur/ de la
% fonction-mère, car on veut qu'elles agissent sur certaines variables de

```

% la fonction-mère.

```
function troislancers
    % Analogue pipé de la sous-routine "troislancers" du code « naïf ».
    % Noter qu'ici on ne demande pas à la fonction de retourner son
    % résultat ; elle se contente de l'écrire dans la variable
    % "resultat" de la fonction-mère.

    des = [pipe,pipe,pipe,pipe,pipe];
    des = relancer(des);
    des = relancer(des);
    resultat = all(des == 6);
end

function desrelances = relancer(des)
    % Analogue pipé de la fonction "desrelances" du code « naïf ».

    % Pour départager les ex-æquo de façon symétrique, on initialise
    % "occurrences" par des nombres fractionnaires aléatoires.
    occurrences = rand(1,6);
    for j = 1:5,
        occurrences(des(j)) = occurrences(des(j))+1;
    end
    [~, recordman] = max(occurrences);
    desrelances = des;
    for j = 1:5,
        if des(j) ~= recordman
            desrelances(j) = pipe;
        end
    end
end

function chiffre = pipe
    % La sous-routine "pipe" tire un dé pipé. Elle tient en outre le
    % compte des "6" et des "non-6" qu'elle a tiré.

    if(rand<lambda)
        chiffre = 6;
    else
        chiffre = randi([1,6]);
    end
    % La condition suivante permet de compter le nombre total de "6" et
    % de "non-6" qui ont été tirés.
    if (chiffre == 6)
        six = six + 1;
    else
        nonsix = nonsix + 1;
    end
end
```

end

III.2 Le code pour la course à pied

```
function course(N)
% La fonction "course" évalue la durée moyenne de la course par la méthode
% de Monte-Carlo. L'argument "N" est le nombre de simulations effectuées.

% Les variables "somme" et "sommecarres", comme leurs noms l'indiquent,
% sont destinées à contenir la somme, resp. la somme des carrés de toutes
% les variables simulées. On les initialise chacune à 0.
somme = 0;
sommecarres = 0;
% On lance maintenant les N simulations.
for i = 1:N
    % La variable "chronos" est la série des 8 chronos réalisés par les
    % coureurs.
    chronos = exp(4+sqrt(.03)*randn(1,8));
    % La variable "dernier" est le temps du plus lent des coureurs.
    dernier = max(chronos);
    somme = somme + dernier;
    sommecarres = sommecarres + dernier*dernier;
end
% On évalue maintenant la moyenne m de la variable aléatoire "dernier" et
% son écart-type "sigma".
m = somme/N;
disp('Estimateur de m :');
disp(m);
sigma = sqrt(sommecarres/N-m*m);
disp('Estimateur de sigma :');
disp(sigma);
% On calcule l'intervalle de confiance à 80 %.
I = m + 1.29*sigma/sqrt(N)*[-1 1];
disp('Intervalle de confiance à 80 % :');
disp(I);
end
```

III.3 Codes pour les calculs de volume de boule

III.3.a Méthode de Monte-Carlo sur le cube

```
function volboule(d,N)
% La fonction "volboule" évalue la volume de la boule unité de dimension d
% par la méthode de Monte-Carlo en échantillonnant sur le cube la
% contenant. N est le nombre de simulations demandées.

% "positifs" compte le nombre de fois que l'échantillon est tombé dans la
% boule.
positifs = 0;
tic;
for i = 1:N
    % x est le point du cube simulé.
    x = -1+2*rand(1,d);
    % On teste si la distance à l'origine est plus petite que 1.
```



```

        if(sqrt(sum(x.*x))<1)
            positifs = positifs+1;
        end
    end
    t=toc;
    p = positifs/N;
    % Notez que comme la fonction évaluée est une indicatrice, nous n'avons pas
    % besoin de compter séparément la somme et la somme des carrés: la variance
    % est automatiquement liée à l'espérance.
    sigma = sqrt(p*(1-p)/N);
    disp('Intervalle de confiance :')
    disp(2^d*[p-1.96*sigma,p+1.96*sigma]);
    disp('Efficacité :')
    if (sigma == 0)
        disp('J''ai trouvé une variance nulle... Efficacité estimée infinie !');
    else
        % N'oubliez pas que le "sigma" donné ci-dessus ne tenait pas compte du
        % facteur 2^d: il faut donc intégrer de facteur dans le calcul de
        % l'efficacité.
        disp(1/t/(2^d*sigma)^2);
    end
end

```

III.3.b Méthode de Monte-Carlo avec échantillonnage préférentiel

```

function volboulepref(d,N)
% La fonction "volboulepref" évalue le volume de la boule unité de
% dimension d par la méthode de Monte-Carlo en échantillonnant selon la
% gaussienne multidimensionnelle idoine. N est le nombre de simulations
% demandées.

% Cette fois-ci, la fonction évaluée n'est plus une indicatrice, donc il
% faut compter séparément la somme et la somme des carrés.
somme = 0;
sommecarres = 0;
% C est la densité en 0 de la gaussienne qui sert à notre échantillonnage :
% comme elle intervient à chaque simulation et que son calcul prend un
% certain temps, on la calcule une fois pour toutes.
C = (sqrt(d)/sqrt(2*pi))^d;
tic;
for i = 1:N
    % x est le point simulé.
    x = randn(1,d)/sqrt(d);
    % r est la distance de x à l'origine.
    r = sqrt(sum(x.*x));
    % On teste si la distance à l'origine est plus petite que 1.
    if(r<1);
        % Il faut tenir compte de la densité d'échantillonnage.
        y = 1/(C*exp(-d*r*r/2));
        somme = somme+y;
        sommecarres = sommecarres+y;
    end
end
t=toc;
m = somme/N;

```

```

sigma = sqrt((sommecarres/N-m*m)/N);
disp('Intervalle de confiance :')
disp([m-1.96*sigma,m+1.96*sigma]);
disp('Efficacité :')
disp(1/t/(sigma*sigma));
end

```

III.4 Les codes pour le calcul de l'intégrale cubique

III.4.a Échantillonnage selon la loi *Cauchy*(1)

```

function integrale1(N)
% Cette fonction évalue \int_R dx/(1+|x|^3) en échantillonnant selon une
% loi de Cauchy standard.

% Initialisation de la somme et de la somme des carrés.
somme = 0;
sommecarres = 0;
% Démarrage du chronomètre. Je ne chronomètre que la boucle, dans la mesure
% où le temps pris par la partie hors de la boucle ne dépend pas de N.
tic;
for i = 1:N
    % Échantillonnage selon la loi de Cauchy standard.
    x = tan(pi*(rand-.5));
    % Calcul de la quantité dont on veut connaître l'espérance. Notez que
    % le facteur pi qui ressort pourrait être factorisé pour économiser les
    % calculs, mais attention, il faudrait aussi penser dans ce cas à
    % multiplier la variance par pi^2.
    y = pi*(1+x*x)/(abs(x)^3+1);
    % Incrémentation de la somme et de la somme des carrés
    somme = somme+y;
    sommecarres = sommecarres+y*y;
end
t=toc;
% Calcul des quantités qui nous intéressent. m est la moyenne empirique, v
% la variance empirique, err la marge d'erreur et I l'intervalle de
% confiance. Les noms des trois autres quantités calculées sont
% transparents.
m = somme/N;
v = sommecarres/N-m*m;
err = 1.96*sqrt(v/N);
I = [m-err m+err];
eff_par_etape = 1/v;
etape_par_s = N/t;
eff_par_s = eff_par_etape*etape_par_s;
disp('Estimateur :');
disp(m);
disp('Intervalle de confiance :');
disp(I);
disp('Temps de calcul (s) :');
disp(t);
disp('Nombre d''étapes de calcul par seconde :')
disp(etape_par_s);
disp('Efficacité par étape :');
disp(1/v);

```

```

disp('Efficacité par seconde :')
disp(eff_par_s)
end

```

III.4.b Échantillonnage double-exponentiel

```

function integrale2(N)
% Cette fonction évalue  $\int_{\mathbb{R}} dx/(1+|x|^3)$  en échantillonnant selon une
% loi double-exponentielle standard.

somme = 0;
sommecarres = 0;
tic;
for i = 1:N
    % Échantillonnage selon la loi double-exponentielle standard.
    if (randi([0,1]))
        x = -log(rand);
    else
        x = log(rand);
    end
    % Calcul de la quantité dont on veut connaître l'espérance. Attention,
    % ce n'est pas la même que dans le cas précédent puisqu'on a changé la
    % loi d'échantillonnage.
    y = 2*exp(abs(x))/(abs(x)^3+1);
    somme = somme+y;
    sommecarres = sommecarres+y*y;
end
t=toc;
m = somme/N;
% Attention ! Ici je laisse quand même le calcul de v, err, eff_par_etape
% et eff_par_s pour que vous puissiez voir ce que cela donne, mais je ne
% devrais pas puisque y n'est pas  $L^2$  ! Plus précisément, v va diverger et
% eff_par_etape et eff_par_seconde tendre vers 0, tandis que err ne
% correspond plus à un intervalle d'erreur à 95 %.
v = sommecarres/(N-m*m); err = 1.96*sqrt(v/N); I = [m-err m+err];
eff_par_etape = 1/v; etape_par_s = N/t; eff_par_s = eff_par_etape*etape_par_s;
disp('Estimateur :'); disp(m);
disp('Pseudo intervalle de confiance :'); disp(I);
disp('Temps de calcul (s) :'); disp(t);
disp(['Nombre d''étapes de calcul par seconde :']); disp(etape_par_s);
disp('Pseudo efficacité par étape :'); disp(1/v);
disp('Pseudo efficacité par seconde :'); disp(eff_par_s)
end

```

III.4.c Échantillonnage selon *Cauchy*(0,65)

```

function integrale3(N)
% Cette fonction évalue  $\int_{\mathbb{R}} dx/(1+|x|^3)$  en échantillonnant selon une
% loi de Cauchy de paramètre 0,65.

somme = 0;
sommecarres = 0;
tic;
for i = 1:N
    x = .65*tan(pi*(rand-.5));

```

```

    y = pi/.65*(.65*.65+x*x)/(abs(x)^3+1);
    somme = somme+y;
    sommecarres = sommecarres+y*y;
end
t=toc;
m = somme/N; v = sommecarres/N-m*m; err = 1.96*sqrt(v/N); I = [m-err m+err];
eff_par_etape = 1/v; etape_par_s = N/t; eff_par_s = eff_par_etape*etape_par_s;
disp('Estimateur :'); disp(m);
disp('Intervalle de confiance :'); disp(I);
disp('Temps de calcul (s) :'); disp(t);
disp('Nombre d\'étapes de calcul par seconde :'); disp(etape_par_s);
disp('Efficacité par étape :'); disp(1/v);
disp('Efficacité par seconde :'); disp(eff_par_s);
end

```

III.4.d Échantillonnage de type Pareto

```

function integrale4(N)
% Cette fonction évalue \int_R dx/(1+|x|^3) en échantillonnant selon la loi
% P_4 du cours, laquelle est de type Pareto.

somme = 0;
sommecarres = 0;
tic;
for i = 1:N
    y = rand;
    r = rand;
    if r<1/6
        x = -y^(-1/2);
        f = 3*abs(x)^3/(abs(x)^3+1);
    elseif r<2/6
        x = y^(-1/2);
        f = 3*abs(x)^3/(abs(x)^3+1);
    else
        x = -1+2*y;
        f = 3/(abs(x)^3+1);
    end
    somme = somme+f;
    sommecarres = sommecarres+f*f;
end
t=toc;
m = somme/N; v = sommecarres/N-m*m; err = 1.96*sqrt(v/N); I = [m-err m+err];
eff_par_etape = 1/v; etape_par_s = N/t; eff_par_s = eff_par_etape*etape_par_s;
disp('Estimateur :'); disp(m);
disp('Intervalle de confiance :'); disp(I);
disp('Temps de calcul (s) :'); disp(t);
disp('Nombre d\'étapes de calcul par seconde :'); disp(etape_par_s);
disp('Efficacité par étape :'); disp(1/v);
disp('Efficacité par seconde :'); disp(eff_par_s);
end

```

III.5 Grandes déviations de 100 lancers d'une pièce

III.5.a Algorithme « naïf »

```
function pileoufacenaif(N)
% La fonction "pileoufacenaif" évalue R par échantillonnage non biaisé. N
% est le nombre de simulations demandé.

somme = 0;
sommecarres = 0;
tic;
for i = 1:N
    omega = (rand(1,100)<1/2);
    X = sum(omega);
    y = max(X-60,0);
    somme = somme+y;
    sommecarres = sommecarres+y*y;
end
t = toc;
m = somme/N;
sigma = sqrt(sommecarres/N-m*m);
disp('Intervalle de confiance :');
disp([m-2*sigma/sqrt(N),m+2*sigma/sqrt(N)]);
disp('Temps de calcul :');
disp(t);
end
```

III.5.b Avec échantillonnage préférentiel

```
function pileoufacepref(N)
% La fonction "pileoufacepref" évalue R par échantillonnage préférentiel. N
% est le nombre de simulations demandé.

somme = 0;
sommecarres = 0;
% Comme on aura besoin de la valeur de (1/2)^100 à chaque simulation et que
% son calcul prend un certain temps, on calcul cette valeur une fois pour
% toutes.
C = (1/2)^100;
tic;
for i = 1:N
    omega = (rand(1,100)<.6);
    X = sum(omega);
    y = C*max(X-60,0)/(.6^X*.4^(100-X));
    somme = somme+y;
    sommecarres = sommecarres+y*y;
end
t = toc;
m = somme/N;
sigma = sqrt(sommecarres/N-m*m);
disp('Intervalle de confiance :');
disp([m-2*sigma/sqrt(N),m+2*sigma/sqrt(N)]);
disp('Temps de calcul :');
disp(t);
end
```

III.6 Code pour le dilemme rugbystique

```

function AgenBrive(q,qbis,N)
% La fonction AgenBrive évalue, d'une part la probabilité de victoire de
% l'équipe B en suivant la stratégie S_q, d'autre part la différence entre
% les probabilités associées aux stratégies S_q et S_qbis. N est le nombre
% de simulations demandé.

% "victoires" et "sommecarresv" sont respectivement la somme et la somme des carrés associées
% au calcul de la probabilité de succès sous S_q.
victoires = 0;
sommecarresv = 0;
% "difference" et "sommecarresd" sont respectivement la somme et la somme des carrés associées
% au calcul de la différence des probabilités de succès.
difference = 0;
sommecarresd = 0;
% On lance N simulations de matchs, avec à chaque fois un couplage entre
% l'équipe B suivant la stratégie S_q et une équipe fictive Bbis suivant la
% stratégie S_qbis.
for i = 1:N
    % "A" est le score de l'équipe A.
    A = 3*sum(rand(12,1)<.6);
    % Pour tenir compte du tirage à la pièce éventuel, on ajoute à A un
    % nombre fractionnaire plus petit que 1 qui a une chance sur deux
    % d'être positif.
    A = A+rand-.5;
    % "avantage" est une série qui, en i-ième position, contient 1 si
    % l'équipe B (et donc aussi Bbis) a pris l'avantage à l'issue de sa
    % i-ième séquence de jeu, et 0 sinon.
    avantage = (rand(12,1) < .4);
    % rand12 est la série de variables aléatoires qui sert à coupler les
    % comportements des équipes B et Bbis.
    rand12 = rand(12,1);
    % "Btenteessai" et "Bbistenteessai" disent si les équipes B et Bbis,
    % respectivement, tentent l'essai à l'issue de la i-ième séquence de
    % jeu. "Bprendpenalite" et "Bbisprendpenalite", à l'inverse, disent si
    % les équipes ont plutôt pris la pénalité. Attention, tout cela
    % n'a de sens que sous réserve que les équipes aient pris
    % l'avantage !
    Btenteessai = (rand12 < q);
    Bprendpenalite = 1-Btenteessai;
    Bbistenteessai = (rand12 < qbis);
    Bbisprendpenalite = 1-Bbistenteessai;
    % "essaireussi" dit si l'essai tenté à la i-ième séquence de jeu a été
    % réussi. Attention, cela n'a de sens que sous réserve que l'essai ait
    % effectivement été tenté!
    essaireussi = (rand(12,1) < .35);
    % "B" et "Bbis" sont les scores respectifs des équipes B et Bbis.
    B = sum(avantage.*(3*Bprendpenalite+7*Btenteessai.*essaireussi));
    Bbis = sum(avantage.*(3*Bbisprendpenalite+7*Bbistenteessai.*essaireussi));
    % La simulation est maintenant terminée ; il ne reste plus qu'à faire
    % les comptes !
    if(B>A)
        victoires = victoires+1;
        sommecarresv = sommecarresv+1;
    end
end

```

```
end
if((Bbis>A) && (B<A))
    difference = difference+1;
    sommecarresd = sommecarresd+1;
elseif((Bbis<A) && (B>A))
    difference = difference-1;
    sommecarresd = sommecarresd+1;
end
end
% Calcul et affichage des intervalles de confiance.
p = victoires/N;
v = (sommecarresv/N-p*p)/N;
disp('Intervalle de confiance pour P_q(B) :')
disp([p-2*sqrt(v),p+2*sqrt(v)]);
m = difference/N;
w = (sommecarresd/N-m*m)/N;
disp('Intervalle de confiance pour P_qbis(B) - P_q(B) :')
disp([m-2*sqrt(w),m+2*sqrt(w)]);
end
```

Deuxième partie

Application aux processus aléatoires

Chapitre IV

Le mouvement brownien

IV.1 Approche intuitive du mouvement brownien

Soient X_1, X_2, \dots des variables aléatoires i.i.d. ; pour tout $n \in \mathbb{N}$, on pose $S_n = \sum_{i=1}^n X_i$. On se demande à quoi ressemble la suite $(S_n)_n$ « vue de loin ». Par exemple, si X est L^1 , la loi des grands nombres nous dit que la suite $(S_n)_n$ convenablement renormalisée va converger vers une droite déterministe de pente $E(X)$, au sens où, en définissant \tilde{S}_t^N par $\tilde{S}_t^N := S_{[tN]}/N$, la courbe de $(\tilde{S}_t^N)_{t \geq 0}$ va converger (dans un sens qu'il faudrait préciser) vers la droite d'équation $y = E(X)x$.

Voici par exemple des plots de la courbe \tilde{S}_t^N pour $N = 2^4$, $N = 2^8$ et $N = 2^{12}$ (les X_i sont les mêmes d'une simulation sur l'autre), où X_i suit la loi uniforme sur $(0, 1)$, de moyenne $1/2$. L'échelle est la même sur les trois courbes.

Évidemment, ce n'est pas une limite d'échelle très intéressante... Ce qui serait plus intéressant est de voir ce que donne plutôt le théorème-limite central (qui, lui, garde de l'aléa à la fin) quand on le regarde du point de vue des trajectoires. On suppose donc que X est une variable centrée et L^2 , avec $\text{Var}(X) = \sigma^2$, et on veut regarder ce que donne la suite $(S_n)_n$ « vue de loin ». Toutefois, nous savons que dans le théorème-limite central, l'ordre de grandeur de S_n est $n^{1/2}$, donc cette fois-ci nous définissons \tilde{S}^N par

$$\tilde{S}_t^N := \frac{S_{[tN]}}{N^{1/2}}. \quad (56)$$

Voici ce que cela donne quand X est la loi $\mathcal{N}(1)$ (d'où $\sigma^2 = 1$) et pour X dont la loi est $-1 + 2\beta(\frac{1}{2})$, ce qu'on simule comme le cosinus d'un angle uniforme (pour lequel $\sigma^2 = 1/2$), à chaque fois respectivement pour 2^{12} et 2^{16} simulations. L'échelle a été normalisée de sorte que la variation de σ^2 n'ait pas d'influence.

On remarque que l'allure de la courbe est la même dans les quatre cas ! C'est l'objet-limite décrivant la loi de la courbe quand $N \rightarrow \infty$ que nous allons définir comme étant le mouvement brownien.

Ce qu'on appelle un *processus aléatoire* (indexé par le temps, càdlàg) est une variable aléatoire à valeurs dans un espace de trajectoires, cet « espace de trajectoires » étant l'ensemble des fonctions f d'un intervalle T de \mathbb{R} représentant le temps (en l'occurrence, $T = \mathbb{R}_+$) dans un espace métrique

(séparable) X (en l'occurrence, $X = \mathbb{R}$) qui, en tout point de T , sont continues à droite et admettent une limite à gauche (càd. que pour tout $t \in T$, $\lim_{\substack{u \rightarrow t \\ u < t}} f(u)$ existe dans X , mais peut éventuellement être différente de $f(t)$). Cet ensemble de fonctions càdlàg est noté $\mathcal{D}(T, X)$.

Ici, le mouvement brownien sera donc le processus aléatoire $(B_t)_{t \geq 0} \in \mathcal{D}(\mathbb{R}_+, \mathbb{R})$ dont la loi est la loi-limite de \tilde{S}^N (dans un sens à préciser).

Décrivons la loi de $(B_t)_{t \geq 0}$. Tout d'abord, quelle est la loi de B_t pour un t fixé ? (ce qu'on appelle sa *marginale unidimensionnelle*). Comme tN est très grand, on a à la limite que $S_{tN}/(tN)^{1/2}$ suit la loi $\mathcal{N}(\sigma^2)$ d'après le théorème-limite central, donc $B_t = S_{tN}/(tN)^{1/2} \times t$ suit la loi $\mathcal{N}(\sigma^2 t)$. Mais, pour décrire un processus, il ne suffit pas de connaître la loi de chaque marginale unidimensionnelle : il faut aussi savoir quelle est la loi *jointe* des ces variables, c'est-à-dire, pour t_1, t_2, \dots, t_k , déterminer la loi de $(B_{t_1}, B_{t_2}, \dots, B_{t_k}) \in X^k$, et ce pour n'importe quelle valeur finie de k : c'est ce qu'on appelle les *marginale finidimensionnelles* du processus.

Nous admettons que la connaissance des lois finidimensionnelles d'un processus (càdlàg) nous permet de décrire totalement ce processus : grâce à elles, on peut en effet déterminer la loi jointe du processus sur une partie dénombrable dense de T , puis en déduire la trajectoire complète en utilisant la continuité à droite.

Déterminons donc les lois finidimensionnelles du mouvement brownien : on prend $0 < t_1 < t_2 < \dots < t_k$ et on cherche à déterminer la loi de $(B_{t_1}, B_{t_2}, \dots, B_{t_k})$. Il sera pratique pour simplifier les notations de poser $t_0 := 0$, en notant qu'on a forcément $B_0 = 0$ puisque c'est la limite de S_0/\sqrt{N} et que S_0 est toujours égal à 0. Si nous revenons à la définition en termes des X_i , on a que

$$B_{t_1} = N^{-1/2} \sum_{i=1}^{\lfloor t_1 N \rfloor} X_i, \quad (57)$$

$$B_{t_2} - B_{t_1} = N^{-1/2} \sum_{i=1}^{\lfloor t_2 N \rfloor} X_i - N^{-1/2} \sum_{i=1}^{\lfloor t_1 N \rfloor} X_i = N^{-1/2} \sum_{i=\lfloor t_1 N \rfloor + 1}^{\lfloor t_2 N \rfloor} X_i, \quad (58)$$

$$B_{t_3} - B_{t_2} = N^{-1/2} \sum_{i=\lfloor t_2 N \rfloor + 1}^{\lfloor t_3 N \rfloor} X_i, \quad (59)$$

$$(60)$$

etc.. Par conséquent, chaque $(B_{t_j} - B_{t_{j-1}})$ provient de X_i différents, et donc indépendants : on en déduit que les $(B_{t_j} - B_{t_{j-1}})$ doivent eux-mêmes être indépendants. Et quelle doit être leur loi ? L'argument que nous avons vu pour établir que B_t avait pour loi $\mathcal{N}(\sigma^2 t)$ se réutilise exactement de la même façon pour montrer, cette fois, que $B_{t_j} - B_{t_{j-1}}$ a pour loi $\mathcal{N}(\sigma^2(t_j - t_{j-1}))$.

Conclusion : la loi de $(B_{t_1}, \dots, B_{t_k})$ est la loi telle que les $(B_{t_j} - B_{t_{j-1}})$

soient indépendants et de lois respectives $\mathcal{N}(\sigma^2(t_j - t_{j-1}))$, càd.

$$\begin{pmatrix} B_{t_1} \\ B_{t_2} - B_{t_1} \\ B_{t_3} - B_{t_2} \\ \dots \\ B_{t_k} - B_{t_{k-1}} \end{pmatrix} \sim \mathcal{N} \left(\sigma^2 \begin{pmatrix} t_1 & 0 & 0 & \dots & 0 \\ 0 & t_2 - t_1 & 0 & \dots & 0 \\ 0 & 0 & t_3 - t_2 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & t_k - t_{k-1} \end{pmatrix} \right). \quad (61)$$

On peut en récupérer la loi de $(B_{t_1}, B_{t_2}, \dots, B_{t_k})$ par une transformation linéaire. Cela donne que $(B_{t_1}, B_{t_2}, \dots, B_{t_k})$ est toujours un vecteur gaussien centré, de matrice de covariance

$$\begin{pmatrix} B_{t_1} \\ B_{t_2} \\ \dots \\ B_{t_k} - B_{t_{k-1}} \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} \sigma^2 t_1 & \sigma^2 t_1 & \dots & \sigma^2 t_1 \\ \sigma^2 t_1 & \sigma^2 t_2 & \dots & \sigma^2 t_2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^2 t_1 & \sigma^2 t_2 & \dots & \sigma^2 t_k \end{pmatrix} \right); \quad (62)$$

autrement dit, $\text{Cov}(B_{t_j}, B_{t_l}) = \sigma^2(t_j \wedge t_l)$. Noter au passage que cette dernière expression reste valable même si les t_j ne sont pas rangés dans l'ordre.

Remarque : les processus dont toutes les marginales finidimensionnelles sont gaussiennes sont appelés les processus gaussiens. Pour décrire un processus gaussien $(X_t)_{t \in T}$, il suffit de donner les moyennes $\mathbb{E}(X_t)$ pour tout t (pour le mouvement brownien, $\mathbb{E}(X_t) = 0$) et les covariances $\text{Cov}(X_t, X_s)$ pour tout (t, s) (pour le mouvement brownien, $\text{Cov}(X_t, X_s) = t \wedge s$).

Remarque : Dans les situations physiques, il se peut très bien que la quantité σ^2 soit dimensionnée : par exemple, si notre mouvement brownien représente la position d'un ivrogne dans une rue en fonction du temps, σ^2 est en $m^2.s^{-1}$. Quand les axes sont adimensionnés et que $\sigma^2 = 1$, on parle de mouvement brownien *standard*. En général, quand on dit « mouvement brownien », on sous-entend « standard » ; dans ce cours, je ferai la distinction mais sachez que cela ne sera pas le cas en général.

IV.2 Propriétés du mouvement brownien

Dans toute cette section, il est sous-entendu que $(B_t)_{t \geq 0}$ est un mouvement brownien. Ce mouvement brownien ne sera pas nécessairement standard ; par contre, quand je dis plusieurs fois « un mouvement brownien » au cours d'un même énoncé, cela sous-entend que c'est pour la même constante σ^2 .

59.a Théorème (accroissements indépendants). *Pour tous $t_1 < t'_1 \leq t_2 < t'_2 \leq \dots \leq t_k < t'_k$, les variables aléatoires $(B_{t'_1} - B_{t_1}), (B_{t'_2} - B_{t_2}), \dots, (B_{t'_k} - B_{t_k})$ sont indépendantes. On dit que le mouvement brownien est un processus à accroissements indépendants.*

Démonstration. Immédiat d'après la façon dont nous avons construit le mouvement brownien à partir de ses marginales finidimensionnelles. \square

59.b Théorème (accroissements stationnaires). *Pour tout $T \geq 0$, la loi du processus $(B'_t)_{t \geq 0}$ défini par $B'_t = B_{T+t} - B_T$ est la même que la loi du mouvement brownien. En d'autres termes, la loi du mouvement brownien vu comme processus à valeurs dans $\mathcal{D}(\mathbb{R}_+, \mathbb{R})/\mathbb{R}$ (l'espace des fonctions càdlàg à valeurs réelles considérées à constante additive près) est invariante par translation de l'échelle de temps. On dit que le mouvement brownien est un processus à accroissements stationnaires.*

60.a Remarque. Une autre façon de dire que le mouvement brownien est un processus à accroissements stationnaires est de dire que pour tous $t_0 < t_1 < \dots < t_k$, pour tout $T \geq 0$, $(B_{t_1} - B_{t_0}, B_{t_2} - B_{t_0}, \dots, B_{t_k} - B_{t_0})$ a la même loi que $(B_{T+t_1} - B_{T+t_0}, B_{T+t_2} - B_{T+t_0}, \dots, B_{T+t_k} - B_{T+t_0})$. Quand on sait que les accroissements sont indépendants, on peut du reste se contenter de vérifier cette propriété pour $k = 1$.



Les deux théorèmes précédents nous disent donc que le mouvement brownien est un processus à accroissements indépendants et stationnaires. En fait, ce qui se cache derrière est qu'il existe un « processus des accroissements du mouvement brownien » : le bruit blanc gaussien.

60.b Définition. Si $(B_t)_{t \geq 0}$ est un mouvement brownien (standard), sa dérivée B' est un processus à valeurs distributions appelé *bruit blanc gaussien (standard)* sur \mathbb{R}_+ .

60.c Théorème. *On peut caractériser la loi du bruit blanc gaussien de la façon suivante : si w est une distribution aléatoire à support contenu dans \mathbb{R}_+ presque-sûrement, telle que pour toute fonction-test $\varphi \in \mathcal{C}_c^\infty(\mathbb{R})$, $\langle w, \varphi \rangle$ suive une loi normale centrée de variance $\sigma^2 \|\mathbf{1}_{\mathbb{R}_+} f\|_{L^2}^2$, alors w est un bruit blanc gaussien sur \mathbb{R}_+ (et réciproquement).*

L'avantage de ce théorème est qu'il permet d'étendre immédiatement la définition du bruit blanc gaussien sur \mathbb{R} tout entier :

60.d Définition. On appelle *bruit blanc gaussien sur \mathbb{R}* un processus w à valeurs distributions tel que, pour toute fonction-test φ ,

$$\langle w, \varphi \rangle \sim \mathcal{N}(\sigma^2 \|f\|_{L^2}^2). \quad (63)$$

(Cela existe).

Dans un sens, le bruit blanc gaussien est un objet encore plus fondamental que le mouvement brownien. En fait, le mouvement brownien n'est autre que « l'intégrale d'un bruit blanc gaussien », l'avantage du mouvement brownien par rapport au bruit blanc gaussien est que c'est un processus à valeur dans l'espace « gentil » des fonctions càdlàg alors que le bruit blanc gaussien doit être défini sur l'espace « dégoûtant » des distributions. Mais, comme vous pouvez le voir, les propriétés sur les accroissements indépendants et stationnaires du mouvement brownien deviennent beaucoup plus simples :

60.e Théorème.

1. Le bruit blanc gaussien est « indépendant de lui-même à tout instant » : si $\varphi_1, \varphi_2, \dots, \varphi_k$ sont des fonctions-tests à supports disjoints, alors $\langle w, \varphi_1 \rangle, \langle w, \varphi_2 \rangle, \dots, \langle w, \varphi_k \rangle$ sont indépendants.
2. Le bruit blanc gaussien est stationnaire, càd. que pour tout $T \in \mathbb{R}$, notant τ_T l'opérateur de translation par T , $\tau_T w$ a la même loi que w .

61.a Théorème. Si w est un bruit blanc gaussien complexe (la définition étant la même, mutatis mutandis, que dans le cas réel), alors sa transformée de Fourier (en prenant pour la transformée de Fourier la convention isométrique) est elle aussi un bruit blanc gaussien (complexe) de même paramètre.

En particulier, la densité spectrale de puissance du bruit blanc gaussien est constante sur tout le spectre. On en déduit que le mouvement brownien, en tant qu'intégrale du bruit blanc gaussien, a une densité spectrale de puissance entre les fréquences ω et $\omega + d\omega$ qui est proportionnelle à $\omega^{-2}d\omega$.

**

Revenons au mouvement brownien :

61.b Théorème (continuité). Le mouvement brownien est continu (presque-sûrement). On peut donc le voir directement comme un processus à valeurs dans l'espace des fonctions continues.

Démonstration. Je signale juste que la preuve est non triviale (sans être non plus très difficile) : rien de ce que nous avons vu pour l'instant ne permettait de dire « évidemment » que le mouvement brownien devait être continu ! \square

Le théorème suivant nous dit que n'importe quel processus à accroissements indépendants et stationnaires (ce qu'on appelle aussi un *processus de Lévy*) et qui est *continu* est essentiellement un mouvement brownien :

61.c Théorème. Soit $(X_t)_{t \geq 0}$ est un processus à accroissements indépendants et stationnaires, continu, issu de 0 (càd. tel que $X_0 = 0$ p.-s.), alors il existe deux paramètres $\lambda \in \mathbb{R}$ et $\sigma \in \mathbb{R}_+$ tels que X_t a la même loi que $\lambda t + \sigma B_t$, où $(B_t)_{t \geq 0}$ est un mouvement brownien standard. La réciproque est clairement vraie aussi.

61.d Remarque. Ce théorème signifie notamment qu'un processus de Lévy continu est nécessairement gaussien. Notez que λ et σ peuvent être retrouvés à partir de la loi de ce processus, puisqu'on a $\mathbb{E}(X_t) = \lambda t$ et $\text{Cov}(X_t, X_s) = \sigma^2(t \wedge s)$.

61.e Définition. Informellement, on appelle *temps d'arrêt* un temps aléatoire T qu'on peut déterminer en ne regardant que le passé de la trajectoire du mouvement brownien jusqu'à ce temps. Moralement, un temps d'arrêt est un moment auquel on crie « STOP ! » EN SUIVANT UNE RÈGLE DÉTERMINISTE PAR RAPPORT À LA TRAJECTOIRE QU'ON DÉCOUVRE AU FUR ET À MESURE.

61.f Exemple. Par exemple, « le premier instant T tel que $|B_T - B_{T-1}| \geq 3$ » est un temps d'arrêt. Plus généralement, les temps d'arrêt ont généralement des définitions qui commencent par « le *premier* instant tel que... », et non par « le *dernier* instant tel que... »

62.a Remarque. Formellement, un temps d'arrêt est une variable aléatoire T telle que pour tout $t_0 \geq 0$, l'événement « $T \leq t_0$ » soit mesurable par rapport à $(B_t)_{t \leq t_0}$.

62.b Remarque. En fait, le concept de temps d'arrêt se définit normalement à partir du concept de *filtration* : une filtration est une suite de tribus $(\mathcal{F}_t)_{t \geq 0}$ où \mathcal{F}_t représente l'ensemble de l'information dont on dispose au temps t . Ici j'ai donné la définition d'un temps d'arrêt pour la filtration naturelle du mouvement brownien, mais il se peut qu'en réalité on ait accès à plus d'information que cela. On peut rendre les théorèmes qui suivent vrais si on suppose que le mouvement brownien est markovien *pour la filtration considérée*. Pour alléger la présentation, nous ne rentrerons pas dans ces considérations ici.

62.c Théorème (propriété de Markov). *Si T est un temps d'arrêt, alors le processus $(B'_t)_{t \geq 0}$ défini par $B'_t = B_{T+t} - B_t$ est indépendant de $(B_t)_{t \leq T}$ et a la loi d'un mouvement brownien.*

62.d Remarque. On généralise, on définit plutôt la propriété de Markov de la façon suivante, dont on pourrait montrer qu'elle est équivalente à la propriété donnée ci-dessus : pour tout $t \in \mathbb{R}_+$, $(B_u)_{u \leq t}$ et $(B_u)_{u \geq t}$ sont indépendants conditionnellement à B_t . Cette forme-ci est évidente à vérifier d'après la définition du mouvement brownien.

62.e Théorème (propriété de martingale locale). *Pour $M < \infty$, notons τ_M le premier instant auquel $|B_t| = M$ (ce qui définit un temps d'arrêt), et notons $(B_t^M)_{t \geq 0}$ le « mouvement brownien arrêté au bord de $[-M, M]$ » défini par $B_t^M := B_{t \wedge \tau_M}$. Alors pour tout temps d'arrêt T qui est presque-sûrement fini, $\mathbb{E}(B_T^M) = 0$.*

62.f Remarque. Ce théorème signifie en particulier que si la quantité d'argent dont vous disposez suit un mouvement brownien, il n'y a aucun moyen de « retirer ses billes au bon moment » qui vous assure de gagner de l'argent en moyenne, dès lors que la réserve d'argent dont vous disposez initialement est finie (càd. que vous êtes définitivement ruiné si vous avez perdu M euros par rapport à la situation initiale).

La propriété de martingale locale peut vous sembler peu naturelle, et de fait, il faudrait passer par des exemples assez compliqués pour comprendre sa pertinence. En fait, dans le cas du mouvement brownien lui-même, on a une propriété de martingale dite *globale* :

62.g Théorème (propriété de martingale). *Pour T un temps d'arrêt borné, (B_T) est L^1 et $\mathbb{E}(B_T) = 0$.*

62.h Remarque. La définition usuelle de la propriété de martingale est la suivante, qu'on peut montrer être équivalente à celle que j'ai donnée ci-dessus : si $s < t$, l'espérance de B_t connaissant $(B_u)_{u \leq s}$ est exactement égale à B_s . Cette forme-ci de la propriété est évidente à montrer à partir de la définition du mouvement brownien.

63.a Théorème. Dans le cas du mouvement brownien, le théorème 62.g reste vrai si on suppose simplement que T est d'espérance finie.

✱

63.b Théorème (invariance par réflexion). Si on définit $(B'_t)_{t \geq 0}$ par $B'_t := -B_t$, alors B' a aussi la loi d'un mouvement brownien.

Démonstration. Immédiat. □

63.c Théorème (invariance par changement d'échelle). Pour $\lambda \in (0, \infty)$, si on définit $(B'_t)_{t \geq 0}$ par $B'_t = B_{\lambda t} / \lambda^{1/2}$, alors B' a aussi la loi d'un mouvement brownien.

Démonstration. À nouveau, il suffit de revenir à la définition. □

✱

63.d Théorème (trivialité de la tribu instantanée). Soit A un événement qui ne dépend que de la trajectoire du mouvement brownien sur l'intervalle de temps $[0, \varepsilon]$, et ce, pour tout $\varepsilon > 0$ (un tel événement serait, par exemple, « le mouvement brownien a un minimum local au temps 0 »). Alors la probabilité de A est 0 ou 1.

63.e Remarque. Moralement, cela signifie qu'il n'est pas possible de tirer une information à partir du germe d'un mouvement brownien (si on sait déjà qu'on a affaire à un mouvement brownien, s'entend).

63.f Exemple. La probabilité que 0 soit un minimum local du mouvement brownien est nulle (car sinon elle devrait valoir 1, mais par invariance par symétrie la probabilité que 0 soit un *maximum* local serait également 1, ce qui est absurde).

63.g Exemple. La probabilité qu'il y ait des maxima locaux du mouvement brownien arbitrairement proches de 0 est égale à 1. En effet, il est évident que la probabilité non nulle d'avoir un maximum local dans $[0, 1]$ (par exemple si $B_{1/2} > 0$ et $B_1 \leq 0$); donc d'après l'invariance par changement d'échelle, il a une probabilité non nulle d'avoir un maximum local dans $[0, \varepsilon]$, laquelle est indépendante de ε . Or s'il y avait une probabilité nulle que les maxima locaux du mouvement brownien s'accumulent sur le temps 0, la probabilité qu'il y ait un maximum local dans $[0, \varepsilon]$ tendrait vers 0 quand $\varepsilon \rightarrow 0$, ce que nous venons d'exclure.

63.h Remarque. L'exemple précédent nous montre donc que le mouvement brownien n'arrête pas de monter et de redescendre, même sur des temps arbitrairement petits!

63.i Exemple. La probabilité qu'il existe une constante $C < \infty$ telle que $|B_t| \leq Ct^{1/2}$ au voisinage de 0 est nulle. En effet, pour toute constante C fixée, cette probabilité vaut soit 0, soit 1. Mais si cette probabilité valait 1, on aurait $\mathbb{P}(\forall t \leq \varepsilon, |B_t| \leq Ct^{1/2}) \xrightarrow{\varepsilon \rightarrow 0} 1$, donc en particulier $\mathbb{P}(|B_\varepsilon| \leq C\varepsilon^{1/2}) \xrightarrow{\varepsilon \rightarrow 0} 1$, ce qui n'est pas possible vu que cette quantité est constante et non nulle.

Un des résultats les plus fins qu'on ait sur la tribu instantanée, qui nous permet de comprendre finement la régularité du mouvement brownien, est le suivant :

64.a Théorème (loi du logarithme itéré). *Si $(B_t)_{t \geq 0}$ est un mouvement brownien de paramètre de bruit σ^2 , alors avec probabilité 1,*

$$\liminf_{t \searrow 0} \frac{B_t}{\sqrt{t \log |\log t|}} = -\sqrt{2}\sigma; \quad (64)$$

$$\limsup_{t \searrow 0} \frac{B_t}{\sqrt{t \log |\log t|}} = +\sqrt{2}\sigma. \quad (65)$$

64.b Corollaire. *Avec probabilité 1, le mouvement brownien n'est 1/2-hölderien en presque-aucun point; par contre, pour tout $\varepsilon > 0$, il est 1/2-hölderien en presque-tout point.*

64.c Remarque. Avec un peu de travail, on peut montrer que dans les propriétés hölderiennes du mouvement brownien sont en fait valables en *tout* point avec probabilité 1. ^[*]

IV.3 Simulation du mouvement brownien

Remarquons d'abord que le vrai mouvement brownien est paramétré par une durée de temps infinie, ce qu'on ne peut évidemment pas représenter sur notre machine : quand on dira qu'on simule le mouvement brownien, on sous-entendra donc toujours qu'on le simule *jusqu'à un certain temps d'arrêt fini* (qui, en général, sera une bête constante).

D'autre part, même ainsi, l'espace de trajectoires dans lequel le mouvement brownien prend ses valeurs est de dimension infinie, de sorte qu'on ne pourra jamais simuler complètement toute la trajectoire du mouvement brownien. ^[†] Quand on dira qu'on « simule le mouvement brownien » sur $[0, T]$, cela pourra donc signifier deux choses :

- Soit qu'on simule exactement une marginale finidimensionnelle du mouvement brownien, c'est-à-dire qu'on simule $(B_0, B_{T/N}, B_{2T/N}, \dots, B_{(N-1)T/N}, B_T)$ pour un N très grand ;
- Soit qu'on décrit complètement une trajectoire sur $[0, T]$ dont on dit qu'elle est proche du mouvement brownien (par exemple au sens de la norme du supremum).

✱

Pour simuler $(B_0, B_\varepsilon, \dots, B_{N\varepsilon})$, la méthode la plus naïve est aussi la plus efficace ! C'est ce qu'on appelle la *méthode d'Euler stochastique* : on part de

[*]. Attention : je n'ai pas dit que la loi du logarithme itéré était vérifiée en tout point, ce qui serait faux ! (considérer par exemple un maximum local du mouvement brownien). C'est uniquement pour les histoires de régularité hölderienne qu'on peut montrer qu'en fait, le « presque-tout » peut se transformer en « tout ».

[†]. On pouvait déjà dire, dans une moindre mesure, qu'il n'était pas possible de simuler exactement une variable aléatoire continue en raison de la précision finie de la machine ! Cela dit, l'effet dû à la dimensionnalité infinie de la trajectoire est encore plus important car il est plus difficile d'améliorer la précision.

$B_0 = 0$ et on utilise successivement que les $(B_{i\varepsilon} - B_{(i-1)\varepsilon})$ sont indépendants et de lois $\mathcal{N}(\sigma^2\varepsilon)$ pour déterminer successivement les $B_{i\varepsilon}$. Bref, on revient à l'idée du mouvement brownien comme limite d'une marche aléatoire (à ceci près qu'ici la simulation des $B_{i\varepsilon}$ est *exacte* du fait qu'on a pris des accroissements gaussiens).

Remarquez qu'on peut, à partir de cette simulation exacte du mouvement brownien aux temps $i\varepsilon$, en déduire une simulation approchée sur toute la trajectoire : on peut montrer que, dans un certain sens, la meilleure interpolation possible du mouvement brownien à partir de la donnée des $B_{i\varepsilon}$ est de prendre une interpolation affine entre les points simulés.

65.a Remarque. Comme $(B_0, B_\varepsilon, \dots, B_{N\varepsilon})$ est un vecteur gaussien fini-dimensionnel, on pourrait penser à la méthode de Cholesky. *Cette idée est à exclure absolument* (au moins dans le cas du mouvement brownien), le temps de calcul étant considérablement plus lent !

Un point intéressant avec cette méthode est que, si on cherche à décrire plus finement le *même* mouvement brownien en simulant ses valeurs *exactes* en $\varepsilon/2, 3\varepsilon/2, \dots$, il existe une formule toute simple, qui donne lieu à ce qu'on appelle la *technique du point médian* :

65.b Proposition. Soient $t_1 < t_2$. Conditionnellement à $(B_t)_{t \in [0, t_1] \cup [t_2, \infty)}$, la valeur de $B_{(t_1+t_2)/2}$ suit la loi $(B_{t_1} + B_{t_2})/2 + \mathcal{N}((t_2 - t_1)/4)$.

65.c Remarque. Notez que $B_{(t_1+t_2)/2}$ ne dépend de $(B_t)_{t \in [0, t_1] \cup [t_2, \infty)}$ qu'au travers de ses valeurs aux points « intérieurs » t_1 et t_2 : cela découle de la propriété de Markov.

*
**

Mentionnons une autre méthode de simulation qui répond plutôt à l'idée de décrire exactement une fonction qui approche le mouvement brownien, sans qu'on sache la valeur exacte du mouvement brownien lui-même en aucun point. C'est la *méthode de Karhunen-Loève* :

65.d Théorème (Karhunen-Loève). Si Z_1, Z_3, Z_5, \dots sont des v.a. i.i.d. $\mathcal{N}(1)$, alors la fonction définie sur $[0, T]$ par

$$B(t) := \frac{\sqrt{8}}{\pi} \sigma T^{1/2} \sum_{k \in 2\mathbb{N}+1} k^{-1} Z_k \sin\left(\frac{k\pi}{2T} t\right) \quad (66)$$

est un mouvement brownien (sur $[0, T]$).

65.e Remarque. On retrouve l'idée exprimée page 61 : les composantes de Fourier de fréquence $k\pi/2T$ du mouvement brownien y contribuent pour une énergie qui décroît en k^{-2} .

Abordons maintenant très sommairement la problématique du contrôle de l'erreur d'une simulation de mouvement brownien. Quand on voudra appliquer la méthode de Monte-Carlo pour une fonctionnelle du mouvement brownien, cette fonctionnelle aura en général une dépendance compliquée qui demandera vraiment la simulation de toute la trajectoire du mouvement

brownien et pas seulement de ses valeurs en certains points déterminés (par exemple, si la fonctionnelle est le supremum du mouvement brownien sur un certain intervalle de temps). Il est donc utile de savoir contrôler l'erreur systématique qu'on commet en tronquant le mouvement brownien par un objet finidimensionnel.

66.a Proposition. *Si $(\tilde{B}_t)_{t \leq T}$ est une simulation du mouvement brownien par la méthode d'Euler stochastique à N intervalles de temps (avec interpolation linéaire), alors le « vrai » mouvement brownien B simulé par B' vérifie*

$$\mathbb{E}(\|\tilde{B} - B\|_{L^2(0,T)}^2) = \frac{\sigma^2 T^2}{6N}. \quad (67)$$

Cette proposition est assez facile à démontrer, mais elle n'est pas très utile en général... Contrôler la norme L^2 de la différence n'est pas très intéressant ; la norme sup en revanche est souvent une information plus pertinente ! Pour l'instant, nous nous en tiendrons aux deux résultats suivants :

66.b Proposition. *Avec les mêmes notations que pour la proposition précédente, à N fixé, $\mathbb{P}(\|\tilde{B} - B\|_{L^\infty(0,T)}^2 \geq \varepsilon)$ décroît vers 0 plus vite que n'importe quelle exponentielle quand $\eta \nearrow \infty$.*

66.c Proposition. *Toujours avec les mêmes notations, à ε fixé, on a que pour tout $\eta > 0$, $\mathbb{P}(\|\tilde{B} - B\|_{L^\infty(0,T)}^2 \geq \varepsilon/N^{1/2-\eta})$ décroît vers 0 plus vite que n'importe quel polynôme quand $N \nearrow \infty$.*

On pourra retenir que l'erreur commise en norme sup a donc un ordre de grandeur « un tout petit peu plus grand que $O(N^{-1/2})$ ». Dans les situations concrètes, au niveau de ce cours, nous ferons la plupart du temps comme si la simulation du mouvement brownien était exacte, mais en prenant soin de prendre une résolution suffisamment fine pour la simulation pour être pratiquement certains que l'erreur systématique due à l'imprécision de la simulation ne nuise pas au résultat de nos calculs.