

Rapport de projet - Méthodes de Monte-Carlo

LEFAIX Marie,
HOSSEINKHAN Rémy

Professeur : Stoehr Julien

December 2017

1 Algorithme de rejet

1. Justifiez que pour obtenir une réalisation suivant la loi de densité f , on peut appliquer l'algorithme du rejet à \tilde{f} .

Soit $(Y_n)_{n \geq 1}$ les variables aléatoires i.i.d. de densité g et $(U_n)_{n \geq 1}$ i.i.d. de loi $U([0, 1])$ nécessaires à l'algorithme.

On remarque que l'on a $T := \inf\{n \in \mathbb{N}^* : cMg(Y_n)U_n \leq f(Y_n)\} = \inf\{n \in \mathbb{N}^* : Mg(Y_n)U_n \leq \tilde{f}(Y_n)\}$. Ainsi, il suffit d'effectuer le test sur la fonction \tilde{f} afin d'obtenir le temps n pour lequel on obtiendra une réalisation de f .

2. Trouvez les constantes M_1 , M_2 et la densité g telles que, pour tout $x \in \mathbb{R}^2$

$$\tilde{f}_1(x) \leq M_1 g(x) \text{ et } \tilde{f}_2(x) \leq M_2 g(x)$$

On reconnaît dans \tilde{f}_1 et \tilde{f}_2 une partie de la densité d'un vecteur gaussien de moyenne nulle et de matrice de dispersion $\Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}$.

On peut majorer les fonctions trigonométriques par 1.

Par conséquent, on obtient $g \sim N(0, \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix})$, $M_1 = 4\pi$ et $M_2 = 6\pi$.

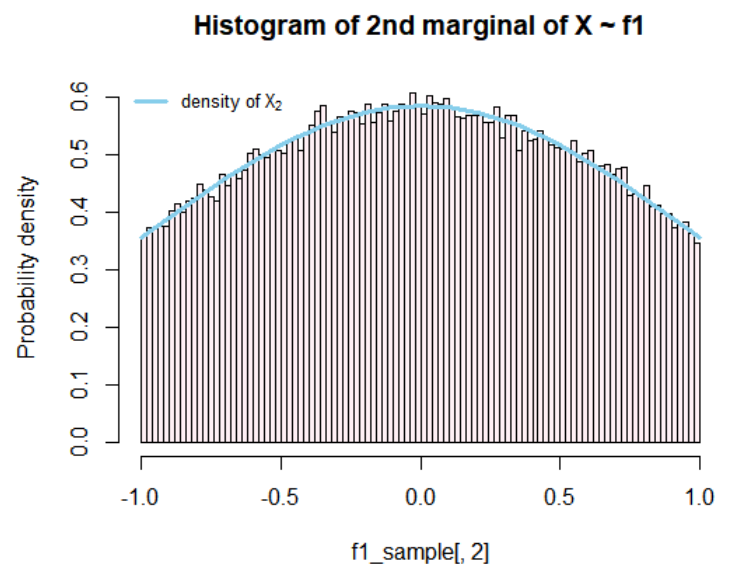
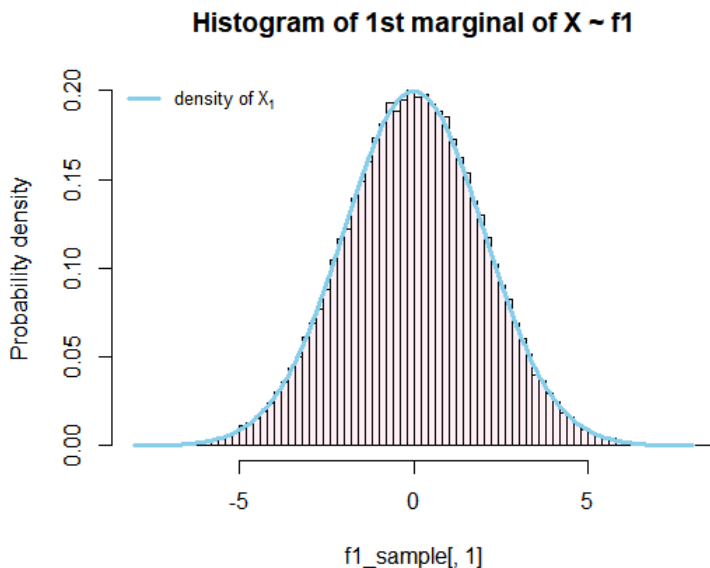
3. En déduire une méthode de simulation suivant les densités f_1 et f_2 .

Pour simuler selon f_1 et f_2 nous allons appliquer la méthode de rejet en utilisant la coïncidence des temps d'arrêts avec ceux des fonctions \tilde{f}_1 et \tilde{f}_2 . Notamment, on va pouvoir simuler selon f_2 alors qu'on ne sait pas calculer explicitement cette densité (constante de normalisation difficile à trouver).

Nos algorithmes utilisent la version vectorielle que vous avez suggérée, en utilisant les constantes M_1 et M_2 afin de prévoir un nombre optimal de variables de densité g pour pouvoir produire des réalisations suivant la densité f . Cette optimisation repose sur le fait que $\mathbb{E}[T] = cM$.

4. Comparer les densités marginales empiriques d'un échantillon suivant la loi de densité f_1 avec les densités marginales théoriques de f_1 .

En intégrant \tilde{f} nous obtenons $c = \frac{1}{\phi(1) - \phi(-1)}$ comme constante de normalisation. Ci-dessous les deux graphes correspondent à l'histogramme de chaque marginale, obtenu pour 100 000 réalisations :



2 Retour sur les méthodes de réduction variance

Nous avons fixé l'échelle des graphes pour que la comparaison visuelle soit pertinente. Nous donnons nos résultats pour 5000 échantillons afin de pouvoir percevoir convenablement la performance des différentes méthodes.

Premier cas. On souhaite calculer pour $X = (X_1, X_2)$ de densité f_1 ,

$$p = \mathbb{P}[e^{X_1} + e^{X_2} \geq 5]$$

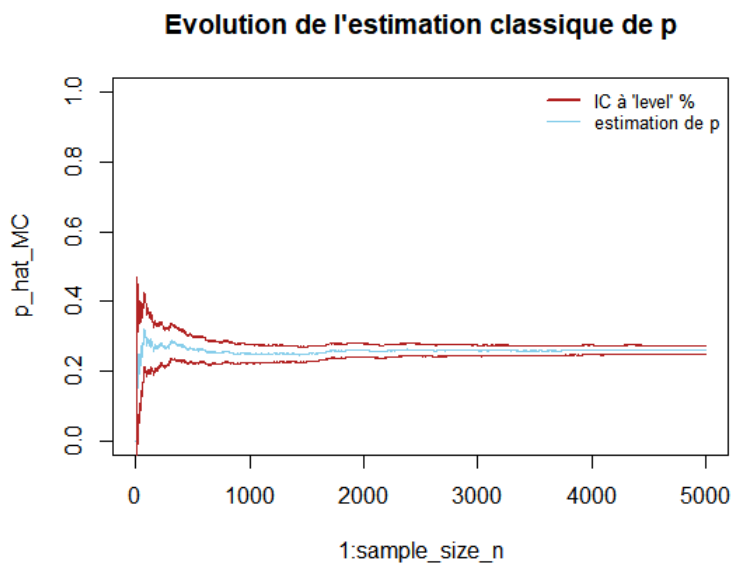
5. Proposer une estimation de p par :

a. la méthode de Monte-Carlo classique

Il s'agit ici d'utiliser la moyenne empirique. La donnée `p_hat_MC[sample_size_n]` est la valeur de notre estimateur à l'état final :

```
> #Estimation Monte-Carlo classique finale de la proba p; variance et IC associés
> p_hat_MC[sample_size_n]
[1] 0.26
> IC_MC[sample_size_n,]
      var      binf      bsup
5000 0.1924385 0.2478407 0.2721593
```

Graphique de l'évolution de l'intervalle de confiance de niveau 5% associé à notre estimation :

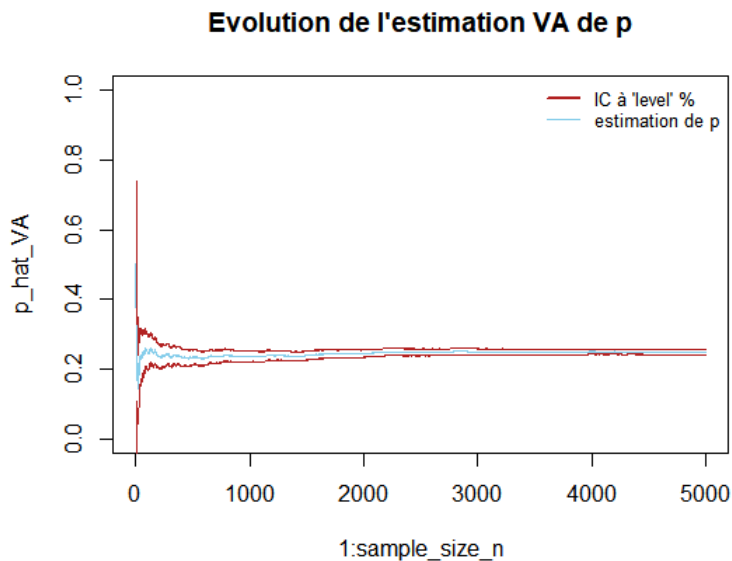


b. la méthode des variables antithétiques

On utilise la symétrie de X . On a utilisé $X' = -X$ qui suit la même loi que X . Comme prévu, on obtient une covariance négative après transformation même si nous sommes déçus par sa valeur :

```
> #Estimation finale, de la proba p; variance et IC associés
> cov(((exp(f1_sample[,1]) + exp(f1_sample[,2])) >= 5), ((exp(-f1_sample[,1]) + exp(-f1_sample[,2])) >= 5))
[1] -0.06131306
> p_hat_VA[sample_size_n]
[1] 0.2478
> IC_VA[sample_size_n,]
      var      binf      bsup
5000 0.06250766 0.2408701 0.2547299
```

Voici le graphe associé :

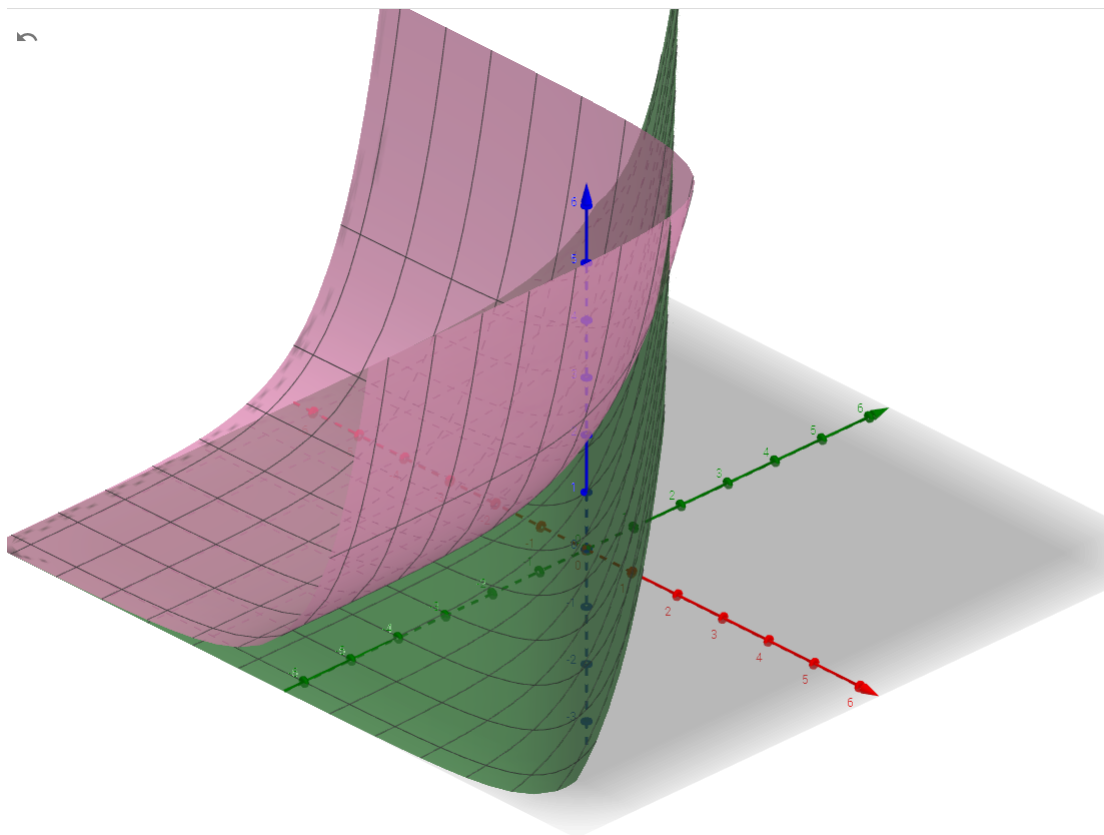


c. la méthode de la variable de contrôle

Il s'agit d'utiliser l'information sur $h := \mathbf{1}_{\{e^{X_1+X_2} \geq 5\}}$ pour construire l'approximation. Pour $Z_1 \sim N(0, 4)$ et $Z_2 \sim N(0, 1)$, on a : $\mathbb{E}[\mathbf{1}_{\{e^{Z_1+Z_2} \geq 5\}} \mathbf{1}_{\{|Z_2| \leq 1\}}] = c \times \mathbb{E}[\mathbf{1}_{\{e^{X_1+X_2} \geq 5\}}]$ (1)

Par conséquent, nous allons pouvoir utiliser la loi normale afin de faire une estimation de p . Nous construisons un estimateur de la forme $\hat{p} = \frac{1}{n} \sum_{i=1}^n h(Z_i) - b(h_0(Z_i) - \mathbb{E}[h_0(Z)])$ avec $h_0 := \mathbf{1}_{\{e^{X_1+X_2} \geq 5\}}$. Nous n'avons malheureusement pas déterminé la valeur exacte de $\mathbb{P}[e^{X_1+X_2} \geq 5]$, ainsi nous avons été contraint d'estimer cette quantité en utilisant (1) et une simulation de 1 000 000 de réalisations des Z_i ce qui enlève de la fiabilité à la méthode. Un développement limité nous a aussi induit la possibilité d'utiliser $h_1 := \mathbf{1}_{\{1+e^{X_1+X_2} \geq 5\}}$ comme fonction de contrôle. Nous donnons sa corrélation avec h un peu plus bas.

Dans le but de se convaincre de la proximité entre h et h_0 , voici une représentation 3D des fonctions $(x, y) \mapsto e^x + e^y$ et $(x, y) \mapsto e^{x+y}$.



Montrons maintenant la corrélation entre la fonction cible et la fonction de contrôle :

```
> #on propose quelques valeurs de corr(h0,g); pour confirmer la proximité de notre fonction h0 avec h !
> cor( x = ((exp(f1_sample[,1]) + exp(f1_sample[,2])) >= 5), y = h_0(f1_sample[,1], f1_sample[,2]))
[1] 0.8555392
>
>
> cor( x = ((exp(f1_sample[,1]) + exp(f1_sample[,2])) >= 5), y = h_1(f1_sample[,1], f1_sample[,2]))
[1] 0.8668369
```

On constate que h_1 pourrait être aussi un bon candidat. La corrélation est bel et bien proche de 1.
Pour un choix initial de la variable de contrôle $b = 1$, on obtient :

```
> #Estimation finale, de la proba p; variance et IC associés
> p_hat_VC[sample_size_n]
[1] 0.2535187
> IC_VC[sample_size_n,]
      var      binf      bsup
5000 0.05042673 0.2472943 0.259743
```

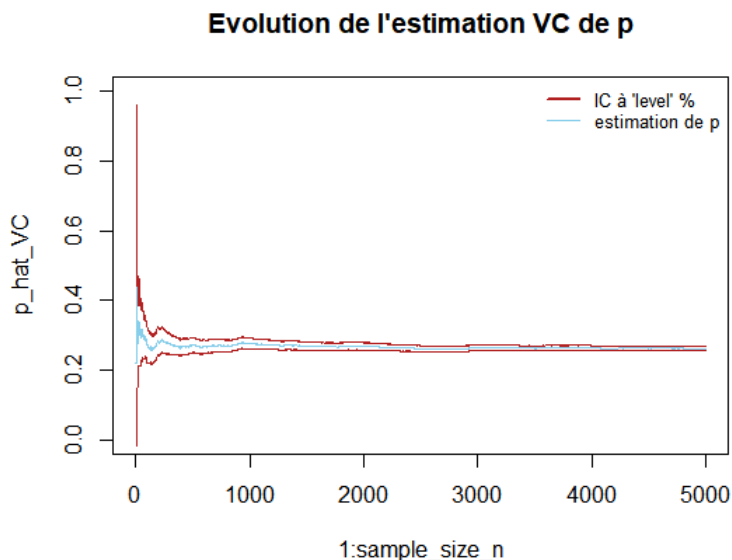
Désormais, il s'agit d'optimiser notre variable de contrôle. On sait que la valeur optimale est atteinte en $b^* = \frac{Cov(h(X), h_0(X))}{Var(h(X))}$ pour l'échantillon X correspondant à l'expérience.

Voilà la nouvelle approximation, sous b^* :

```
> #Nouvelle estimation de p_hat
>
> y_VC <- ((exp(f1_sample[,1]) + exp(f1_sample[,2])) >= 5) - control_term(f1_sample[,1], f1_sample[,2], b = b_opt)
> p_hat_VC <- evol_estim(y_VC, n_expct_h0 = 100000) #g on obtien un sample
> IC_VC <- evol_IC(y_VC, p_hat_VC)
>
>
> #Resultats: (on note une reduction de varianc comme prévu!:)
> p_hat_VC[sample_size_n]
[1] 0.2507559
> IC_VC[sample_size_n,]
      var      binf      bsup
5000 0.04889971 0.2446265 0.2568853
```

On constate effectivement une réduction de la variance de l'estimateur.

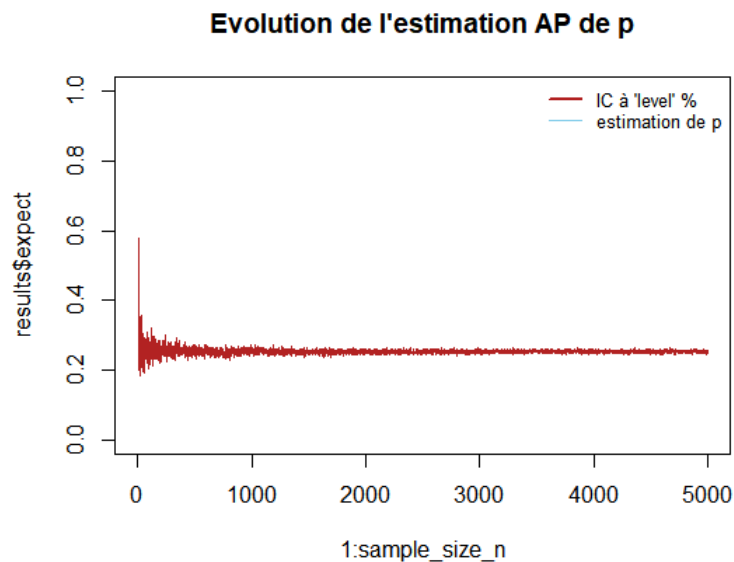
Ci-dessous le graphe de l'évolution :



d. la méthode de stratification avec allocation proportionnelle

L'idée de cette méthode est d'utiliser une partition bien choisie du support. En utilisant le protocole que vous avez suggéré, nous avons obtenu les résultats suivants :

```
> results <- evolution_of_AP_estim(L=5)
> print(results[sample_size_n,])
      expect      var      binf      bsup
5000 0.2526 7.854895e-06 0.2525223 0.2526777
```



La variance a diminué d'une manière impressionnante !

6. Commentez les différents résultats obtenus

On remarque que la variance diminue d'un facteur 3 dès la mise en place d'une méthode par variables antithétiques. La méthode de la variable de contrôle permet de réduire de nouveau cette variance (environ 1 centième). La méthode d'estimation par strates est fabuleuse puisque on atteint une variance de 10^{-6} . Cependant, la vectorisation de cette méthode et de l'algorithme enregistrant l'évolution semble plus complexe.

Second cas. On souhaite calculer

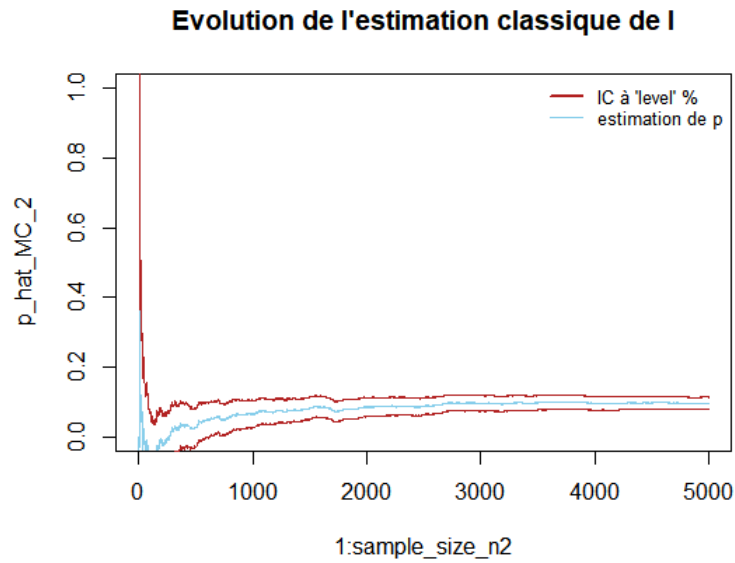
$$I = \int_{\mathbb{R}^2} \cos(y_1 y_2) \sin(y_1) \exp(\sin(y_1 + y_2)) f_2(y_1, y_2) dy_1 dy_2$$

7. Proposer une estimation de I par :

a. la méthode de Monte-Carlo classique

Par estimation classique, nous obtenons :

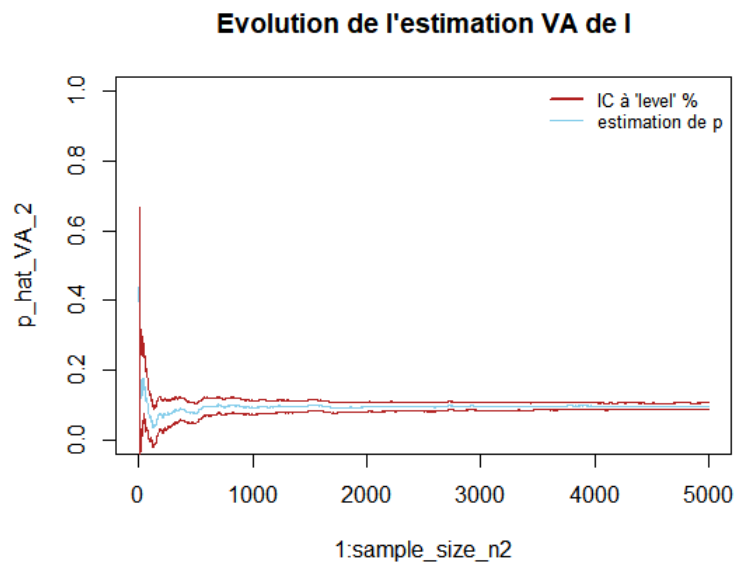
```
> p_hat_MC_2[sample_size_n2]
[1] 0.09589055
> IC_MC_2[sample_size_n2,]
      var      binf      bsup
5000 0.3902056 0.07857605 0.113205
```



b. la méthode des variables antithétiques.

En utilisant la transformation habituelle $A = -1$, nous avons :

```
> p_hat_VA_2[sample_size_n2]
[1] 0.096879
> IC_VA_2[sample_size_n2,]
      var      binf      bsup
5000 0.1010132 0.08806948 0.1056885
```



8. Commenter les différents résultats obtenus.

De nouveau, on voit que la méthode des variables antithétiques réduit significativement la variance (d'un facteur 4 ici).

3 Recyclage dans l'algorithme du rejet

9. Donner la densité des $Z^{(i)}$, $i = 1, \dots, N - T$, conditionnellement à T .

Soit $(Y_n)_{n \geq 1}$ la suite de variables aléatoires de densité g sur laquelle on applique la méthode de rejet. Soit $(U_n)_{n \geq 1}$ la suite de variables aléatoires i.i.d. de loi $U([0, 1])$. On pose $R_1 = \inf\{n \geq 1 : U_n > \alpha(Y_n)\}$ avec $\alpha(Y_n) = \frac{f(Y_n)}{Mg(Y_n)}$ et $R_i = \inf\{n \geq R_{i-1} : U_n > \alpha(Y_n)\}$ pour $i \geq 2$.

On se place sous la mesure $P(\cdot | T = t)$. Ainsi le nombre de variables qui compose X et Z est fixé. On a donc $Z^{(i)} = Y_{R_i}$ pour $i = 1, \dots, N - t$.

On a $\mathbb{E}[h(Y_{R_1})] = \sum_{k=1}^{\infty} \mathbb{E}[h(Y_k) \mathbf{1}_{R_1=k}] = \sum_{k=1}^{\infty} \mathbb{E}[h(Y_k) \mathbf{1}_{U_k > \alpha(Y_k)}] \times \mathbb{P}(U_1 \leq \alpha(Y_1))^{k-1}$ par convergence monotone pour la série et indépendance.

Comme $\mathbb{E}[h(Y_k) \mathbf{1}_{U_k > \alpha(Y_k)}] = \int_{\mathbb{R}^d} h(y) g(y) (1 - \alpha(y)) dy = \int_{\mathbb{R}^d} h(y) \frac{Mg(y) - f(y)}{M} dy$ et $\mathbb{P}(U_1 \leq \alpha(Y_1))^{k-1} = \frac{M}{M-1}$, nous obtenons que la densité de $Z^{(1)}$ est $\frac{Mg-f}{M-1}$. Le raisonnement est similaire pour $i > 1$.

10. Montrer que δ_1 et δ_2 sont des estimateurs sans biais de $\mathbb{E}_f[h(X)]$.

On a $\mathbb{E}[\delta_1(X)] = \frac{1}{t} \sum_{i=1}^t \mathbb{E}[h(X^{(i)})] = \int_{\mathbb{R}^d} h(x) d\mathbb{P}_{X^{(i)}}(x|T=t) = \int_{\mathbb{R}^d} h(x) f(x) dx$ car on sait que $\frac{d\mathbb{P}_{X^{(i)}}(\cdot|T=t)}{dx} = f$.

On a montré que $\frac{d\mathbb{P}_{Z^{(i)}}(\cdot|T=t)}{dx} = \frac{Mg-f}{M-1}$ donc on a :

$$\mathbb{E}[\delta_2(Z)] = \int_{\mathbb{R}^d} \frac{1}{N-t} \sum_{i=1}^{N-t} h(z) \frac{(M-1)f(z)}{Mg(z)-f(z)} d\mathbb{P}_{Z^{(i)}}(z|T=t) = \int_{\mathbb{R}^d} h(z) \frac{(M-1)f(z)}{Mg(z)-f(z)} \frac{Mg(z)-f(z)}{(M-1)} dz = \int_{\mathbb{R}^d} h(z) f(z) dz.$$

11. Montrer que les variables aléatoires δ_1 et δ_2 sont indépendantes et déterminer la variance de $\alpha\delta_1 + (1 - \alpha)\delta_2$.

Soient $X = (X^{(1)}, \dots, X^{(T)})$ et $Z = (Z^{(1)}, \dots, Z^{(N-T)})$. Par construction, toutes ces variables sont des réalisations indépendantes de densité g . L'évènement $\{T = t\}$ fixe parmi les N réalisations, les indices pour lesquels la variable a réussi le test (cf. cours : les variables qui composent X et Z sont exactement les variables indépendantes de densité g , seuls leurs indices sont définis aléatoirement. Par définition, δ_1 et δ_2 sont respectivement des fonctions mesurables de X et de Z , les estimateurs sont donc indépendants.

Par indépendance on obtient que $\text{Var}[\alpha\delta_1(X) + (1 - \alpha)\delta_2(Z)] = \alpha^2 \text{Var}[\delta_1(X)] + (1 - \alpha)^2 \text{Var}[\delta_2(Z)]$.

12. En déduire un choix optimal de α , α^* , et expliquer comment s'opère l'approximation de α^* en pratique.

L'expression de la variance nous induit à minimiser un polynôme de second ordre en α . Le minimum est atteint en $\alpha^* = \frac{\text{Var}(\delta_2(Z))}{\text{Var}(\delta_1(X)) + \text{Var}(\delta_2(Z))}$. On constate que plus α est proche de 1, plus l'on va donner d'importance à δ_1 . Empiriquement nous avons trouvé une valeur de α^* proche de 0,6. Dans la suite, on poursuit l'expérience avec $\alpha^* = 0.55$. En pratique, pour trouver α^* , on peut, comme nous l'avons fait, approcher la valeur optimale par un estimateur (il suffit d'utiliser les fonctions `cov` et `var` de R). Peut-être qu'une descente de gradient permettrait aussi de trouver plus rapidement cette valeur, car la fonction étudiée est convexe et possède alors quelques propriétés agréables.

13. Proposer une expérience de Monte-Carlo comparant, en fonction de N , la convergence de δ_1 et δ_2 et $\alpha^*\delta_1 + (1 - \alpha^*)\delta_2$ vers $p = \mathbb{P}[e^{X_1} + e^{X_1} \geq 5]$:

Nous avons décidé de faire l'expérience pour $N = 1000$ afin que nos sorties graphiques soient analysables un minimum. Attention cette partie du programme R, n'est pas optimisée : pour $N = 5000$ il faut attendre environ 1 min pour obtenir les résultats. Dans notre programme, `delta_3` représente l'estimateur pondéré $\alpha\delta_1 + (1 - \alpha)\delta_2$.

a. pour $X = (X_1, X_2)$ de densité f_1 ;

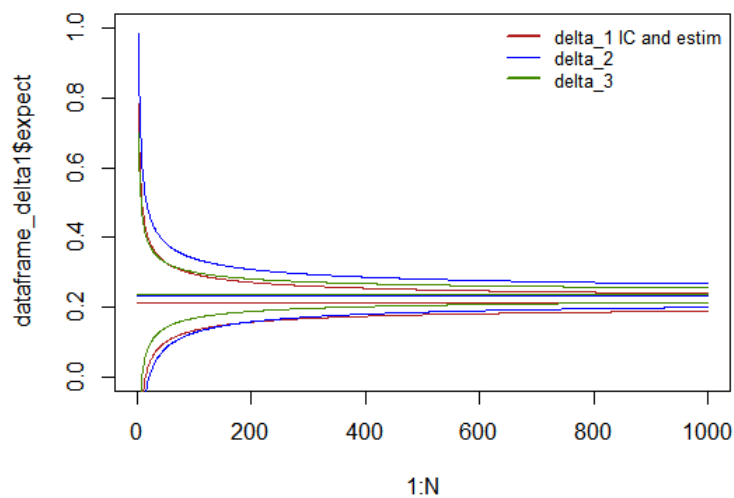
Dans ce cas, voici ce que nous retourne le programme :


```

[1] "Resultat final delta1"
> dataframe_delta1[N,] # afin de voir le resultat
      expect      var      binf      bsup
1000 0.2136364 0.168763 0.1881747 0.239098
>
> plot(1:N, dataframe_delta1$expect, ylim = c(0,1), type = "l", lwd = 1, col = 'firebrick', main = "Evolution
teurs selon N (# of simulations)")
> lines(2:N, dataframe_delta1$binf[2:N], col = "firebrick", lwd = 1)
> lines(2:N, dataframe_delta1$bsup[2:N], col = "firebrick", lwd = 1)
>
>
>
>
> dataframe_delta2 <- evolution_of_deltas(2, f3_sample = 0)
>
> print("Resultat final delta2")
[1] "Resultat final delta2"
> dataframe_delta2[N,]
      expect      var      binf      bsup
1000 0.2327132 0.293791 0.1991187 0.2663076
>
> lines(1:N, dataframe_delta2$expect, ylim = c(0,1), type = "l", lwd = 1, col = 'blue')
> lines(2:N, dataframe_delta2$binf[2:N], col = "blue", lwd = 1)
> lines(2:N, dataframe_delta2$bsup[2:N], col = "blue", lwd = 1)
>
>
>
> dataframe_delta3 <- evolution_of_deltas(3, alpha = 0.55, f3_sample = 0)
>
> lines(1:N, dataframe_delta3$expect, ylim = c(0,1), type = "l", lwd = 1, col = 'chartreuse4')
> lines(2:N, dataframe_delta3$binf[2:N], col = "chartreuse4", lwd = 1)
> lines(2:N, dataframe_delta3$bsup[2:N], col = "chartreuse4", lwd = 1)
>
>
> print("Resultat final delta3")
[1] "Resultat final delta3"
> dataframe_delta3[N,]
      expect      var      binf      bsup
1000 0.2340516 0.1127725 0.2132379 0.2548653

```

Evolution des estimateurs selon N (# of simulations)

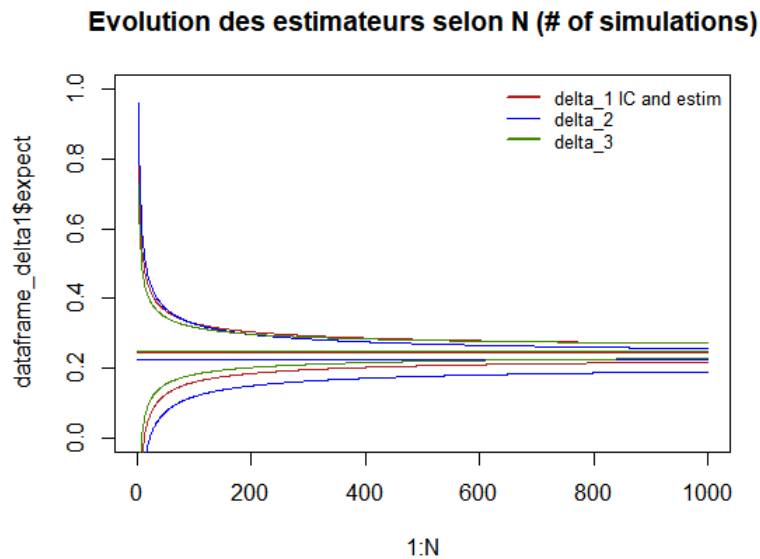


Note : nous avons augmenté la constante de majoration de 30% (choix arbitraire empirique) pour obtenir des variables non nulles qui soient rejetées.

b. pour $X = (X_1, X_2)$ de densité proportionnelle à $\tilde{f}_3(x_1, x_2) = \exp(-\frac{1}{2}(\frac{x_1^2}{4} + x_2^2))(\mathbb{1}_{\{|x_2| \leq 1\}} + 0.5)$.

On obtient :

```
[1] "Resultat final delta1"
> dataframe_delta1[N,] # afin de voir le resultat
  expect      var      binf      bsup
1000 0.2439024 0.1850588 0.2172398 0.2705651
>
> plot(1:N, dataframe_delta1$expect, ylim = c(0,1), type = "l", lwd = 1, col = 'firebrick', main = "Evolution
teurs selon N (# of simulations)")
> lines(2:N, dataframe_delta1$binf[2:N], col = "firebrick", lwd = 1)
> lines(2:N, dataframe_delta1$bsup[2:N], col = "firebrick", lwd = 1)
>
>
>
> dataframe_delta2 <- evolution_of_deltas(2, f3_sample = 1)
>
> print("Resultat final delta2")
[1] "Resultat final delta2"
> dataframe_delta2[N,]
  expect      var      binf      bsup
1000 0.2226174 0.283318 0.1896272 0.2556077
>
> lines(1:N, dataframe_delta2$expect, ylim = c(0,1), type = "l", lwd = 1, col = 'blue')
> lines(2:N, dataframe_delta2$binf[2:N], col = "blue", lwd = 1)
> lines(2:N, dataframe_delta2$bsup[2:N], col = "blue", lwd = 1)
>
>
>
> dataframe_delta3 <- evolution_of_deltas(3, alpha = 0.55, f3_sample = 1)
>
> lines(1:N, dataframe_delta3$expect, ylim = c(0,1), type = "l", lwd = 1, col = 'chartreuse4')
> lines(2:N, dataframe_delta3$binf[2:N], col = "chartreuse4", lwd = 1)
> lines(2:N, dataframe_delta3$bsup[2:N], col = "chartreuse4", lwd = 1)
>
>
> print("Resultat final delta3")
[1] "Resultat final delta3"
> dataframe_delta3[N,]
  expect      var      binf      bsup
1000 0.2493035 0.1192848 0.2278972 0.2707098
```



14. Commenter les différents résultats obtenus. Commenter notamment les performances des ces estimateurs à nombre de simulation fixé avec les estimateurs à taille d'échantillon fixée.

Les estimateurs semblent converger, δ_2 semble sous évaluer la valeur de p . On remarque aussi que l'estimateur obtenu par pondération permet une réduction de la variance d'un facteur d'environ 2. Face aux méthodes à taille d'échantillon fixé, on a l'impression que δ_3 possède une vitesse de convergence plus élevée. Le recyclage permet surtout de réduire la variance face à l'estimation classique : derrière cela il se cache probablement une stratégie d'importance sampling avec un échantillonnage suivant une autre loi. Par ailleurs, on ne comprenait pas pourquoi le ratio de la densité cible par la fonction d'importance était quasi constant. Par définition des fonctions cela s'est compris mais en regardant dans le cours on s'est aperçu que c'était aussi pertinent pour que la méthode se comporte bien. Nous pensions dans un premier temps avoir commis une erreur d'implémentation.

4 Algorithme de Metropolis-Hastings

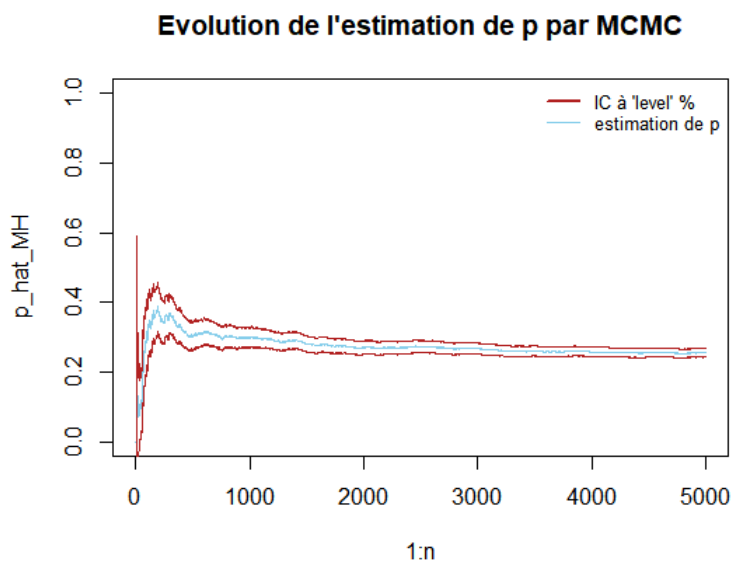
15. A l'aide de l'algorithme de Metropolis-Hastings et en utilisant la fonction g obtenue à la question **2**, produire une chaîne de Markov $(x_t)_{t \in \mathbb{N}} = (x_1^{(t)}, x_2^{(t)})$ de loi stationnaire f_1 et une chaîne de Markov $(y_t)_{t \in \mathbb{N}} = (y_1^{(t)}, y_2^{(t)})$ de loi stationnaire f_2 .

Nous avons simplement implémenté l'algorithme suggéré. Le test du noyau de transition repose sur une loi uniforme générée à chaque tour de boucle. Voir `metropolis_hastings <- function(n, f_tilda = f_tilde1)` pour plus de détails.

16. Étudier, par une expérience de Monte-Carlo, la convergence de l'approximation par le théorème ergodique vers les quantités p et I .

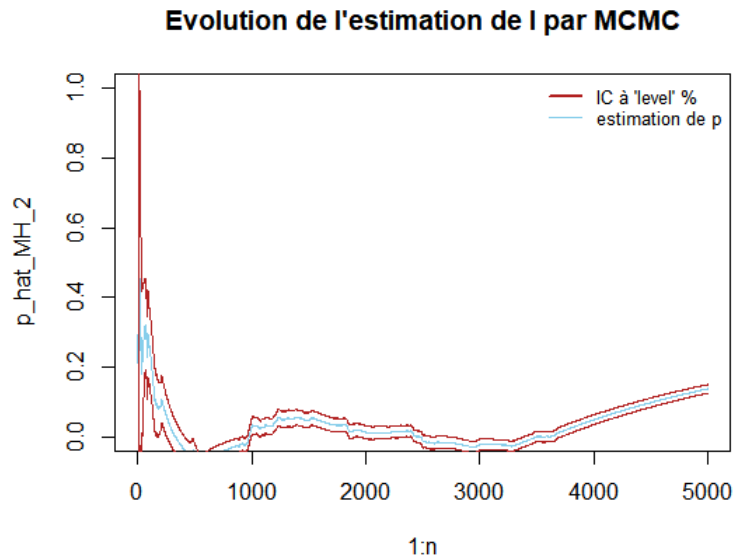
Dans le cadre de l'étude de p nous obtenons :

```
> #Estimation Monte-Carlo classique finale de la proba p; variance et IC associés
> p_hat_MH[n]
[1] 0.2548
> IC_MH[n,]
      var      binf      bsup
5000 0.1899149 0.2427207 0.2668793
```



Pour la valeur de I :

```
> #Estimation Monte-Carlo classique finale de la proba p; variance et IC associés
> p_hat_MH_2[n]
[1] 0.1374179
> IC_MH_2[n,]
      var      binf      bsup
5000 0.197744 0.1250921 0.1497437
```



17. Commenter les différents résultats obtenus.

La méthode se comporte bien pour l'estimation de p , on obtient une variance proche de l'estimateur classique pour l'approximation par théorème ergodique. En revanche, nous n'avons pas réussi à faire converger convenablement la méthode pour l'estimation de I .

5 Conclusion

Ce projet nous aura d'abord permis de mieux comprendre les méthodes de Monte-Carlo que l'on a étudiées, mais surtout de pouvoir faire le lien avec les résultats théoriques que l'on a démontrés ce semestre.

La construction d'expériences a mis en lumière les caractéristiques des différentes méthodes.

Il est très agréable de pouvoir appliquer les connaissances que nous avons acquises jusqu'alors, bien que ce projet ait demandé beaucoup de travail tant au niveau de la réflexion que de la conception. On aura par ailleurs développé notre maîtrise de langage R et de LaTeX.

Nous avons été inspirés par votre perception des statistiques et par votre manière de programmer.

Nous vous remercions pour votre assistance et votre bienveillance durant ce semestre.