Sup'Galilée janvier 2022
MACS1

Initiation au calcul numérique – TP Proba. 2

Simulation de variables aléatoires discrètes

Les exercices marqués d'un astérisque sont à réaliser en **priorité**.

Avant de commencer ce TP, créer un sous-répertoire `TP_Proba_2`, et créer un script (vide) `TP.m` dans ce sous-répertoire.

Dans ce TP, on va s'intéresser à la façon de **simuler des lois discrètes**. Rappelons que la fonction `rand` d'Octave fournit des réalisations de variables aléatoires de loi uniforme sur $[0,1]$. C'est à partir de cette fonction que l'on va chercher à obtenir des variables suivant n'importe quelle loi prescrite.

On verra ensuite comment valider les tirages obtenus en estimant leur loi par méthode de Monte-Carlo sous forme d'histogramme.

Enfin, on appliquera tout cela pour étudier numériquement le modèle d'**urne d'Ehrenfest**, qui a joué un rôle important dans l'histoire de la thermodynamique (et qui sera étudié mathématiquement en MACS 2).

1 Lois discrètes quelconques, lois classiques

On rappelle que définir une probabilité sur un ensemble dénombrable D (c'est-à-dire une loi discrète) revient à se donner une famille $(p_i)_{i\in D}$ de réels positifs tels que

$$\sum_{i\in D} p_i = 1.$$

On dit alors que X a pour loi $(p_i)_{i\in D}$ si X est une v.a. à valeurs dans D et $P(X = i) = p_i$ pour tout $i \in D$.

N.B. Rigoureusement, la loi est $\mu = \sum_{i\in D} p_i\delta_i$, mais on l'assimile souvent, pour simplifier, à la famille $(p_i)_{i\in D}$.

Exercice 1* : Simulation d'une loi quelconque sur $\mathbb{N}$

Soit $(p_n)_{n\in\mathbb{N}}$ une probabilité sur $\mathbb{N}$. Le cas d'une probabilité sur $\{0, 1, \ldots, N\}$ correspond au cas où $p_n = 0$ pour tout $n > N$. On va utiliser la propriété suivante pour simuler une variable aléatoire de loi $(p_n)_{n\in\mathbb{N}}$:

Proposition 1.1. *Soit U une variable aléatoire de loi uniforme sur $[0,1]$. On définit*

$$X = \min\{k \geq 0 \,|\, U < p_0 + p_1 + \cdots + p_k\}$$

Alors X a pour loi $(p_n)_{n\in\mathbb{N}}$.

On commence par démontrer cette propriété.

1. À quelle condition sur U a-t-on $X = 0$? En déduire $P(X = 0)$.

2. Plus généralement, pour $k \geq 1$, à quelle condition sur U a-t-on $X = k$? En déduire $P(X = k)$. Conclure.

3. Écrire une fonction `x=simul_discrete(vp)` qui, étant donné un vecteur `vp=[p0 p1` $\cdots$ `pn]` dont la somme vaut 1, renvoie une valeur aléatoire (à valeurs dans $\{0, 1, \ldots, n\}$) qui suit la loi donnée par `vp`.
4. (Écrire dans `TP.m`) Utiliser la fonction précédente pour remplir un vecteur `tirage1` avec 10000 valeurs aléatoires indépendantes, de loi donnée par le vecteur $p = (0.1, 0.5, 0.25, 0.15)$.
5. On calcule un vecteur qui servira ensuite. Définir une fonction `vp=binomiale(n,p)` qui, étant donné `n` entier et `p`$\in [0,1]$, renvoie le vecteur

$$\texttt{[p0 p1} \cdots \texttt{pn]} = \texttt{[}\binom{n}{0}p^0(1-p)^n,\ \binom{n}{1}p^1(1-p)^{n-1},\ \cdots,\ \binom{n}{n}p^n(1-p)^0\texttt{]}$$

On pourra utiliser la fonction Octave `nchoosek(n,k)` qui renvoie $\binom{n}{k}$.
En déduire une fonction `vx=simul_binomiale(n,p,nb)` qui renvoie un vecteur de `nb` tirages indépendants de loi $\mathcal{B}(n,p)$.
6. (Écrire dans `TP.m`) *La loi binomiale $\mathcal{B}(n,p)$ est la loi $(p_k)_{0\leq k\leq n}$ donnée par $p_k = \binom{n}{k}p^k(1-p)^{n-k}$ pour $k = 0, \ldots, n$.* Remplir un tableau `tirage2` avec 10000 valeurs aléatoires indépendantes, de loi binomiale $\mathcal{B}(10, 0.4)$.

7. *La loi géométrique $\mathcal{G}(p)$ est la loi $(p_k)_{k\geq 1}$ donnée par $p_k = (1-p)^{k-1}p$ pour $k = 1, 2, \ldots$ (et donc $p_0 = 0$).* Vérifier qu'on a $p_0 + \cdots + p_k = 1 - (1-p)^k$ et en déduire que $X = \lfloor \frac{\ln(1-U)}{\ln(1-p)} \rfloor + 1$ dans la Proposition 1.1. En déduire une fonction `x=simul_geometrique(p)` qui, étant donné `p`$\in (0,1]$, renvoie une réalisation d'une variable de loi géométrique $\mathcal{G}(p)$.

8. (Écrire dans `TP.m`) Remplir un tableau `tirage3` avec 10000 valeurs aléatoires indépendantes, de loi géométrique $\mathcal{G}(0.4)$.

Exercice 2* : Estimer des probabilités

Répondre aux questions suivantes dans un script `estim_proba.m`, qui fera afficher les résultats.

1. À l'aide de la méthode de Monte-Carlo, estimer la probabilité $P(X > 7)$, où X suit la loi $\mathcal{B}(20, 0.4)$.

2. À l'aide de la méthode de Monte-Carlo, estimer la probabilité $P(X > Y^2)$, où X suit la loi $\mathcal{B}(20, 0.4)$, Y suit la loi uniforme sur $\{1, 2, 3, 4, 5\}$, et X et Y sont indépendantes.

Exercice 3* : Estimer des espérances

Répondre aux questions suivantes dans un script `estim_espe.m`, qui fera afficher les résultats.

1. À l'aide de la méthode de Monte-Carlo, estimer l'espérance $E[\sqrt{X}]$, où X suit la loi $\mathcal{G}(0.3)$.

2. À l'aide de la méthode de Monte-Carlo, estimer l'espérance $E\Big[\frac{X}{X+Y}\Big]$ où X et Y suivent la loi $\mathcal{G}(0.5)$ et sont indépendantes.

3. À l'aide de la méthode de Monte-Carlo, estimer l'espérance $E\Big[\min\{k \geq 1 \,|\, X_1 + \cdots + X_k > 20\}\Big]$ où $X_1, X_2, \ldots$ suivent la loi uniforme sur $\{1, 2, 3, 4, 5, 6\}$ et sont indépendantes.

4. À l'aide de la méthode de Monte-Carlo, estimer l'espérance de la plus grande valeur propre de la matrice AA^T où A est une matrice aléatoire 5×5 symétrique dont les coefficients sont indépendants et de loi uniforme sur $[-1, 1]$. (Avec `eig`)

Exercice 4 : Simulation de la loi binomiale (2e méthode)

On propose de simuler (à nouveau) la loi binomiale de paramètres n et p à l'aide de sa propriété principale :

Proposition 1.2. *Soit $n \in \mathbb{N}$, $p \in [0, 1]$. Si $X_1, \ldots, X_n$ sont des variables aléatoires indépendantes, qui suivent toutes la loi de Bernoulli de paramètre p, alors $S = X_1 + \cdots + X_n$ suit la loi binomiale de paramètres n et p.*

On rappelle que la loi de Bernoulli de paramètre p est la loi d'une v.a. X telle que $P(X = 1) = p$ et $P(X = 0) = 1 - p$. On rappelle aussi que, si U suit la loi uniforme sur $[0, 1]$, alors la v.a. $\mathbf{1}_{\{U \leq p\}}$ suit la loi de Bernoulli de paramètre p.

1. Grâce à cette proposition, définir une fonction `x=simul_binomiale2(n,p)` qui renvoie une réalisation d'une variable de loi $\mathcal{B}(n, p)$.

2. (Écrire dans `TP.m`) Remplir un tableau `tirage4` avec 10000 valeurs aléatoires indépendantes, de loi binomiale $\mathcal{B}(10, 0.4)$.

Exercice 5 : Simulation de la loi géométrique (2e méthode)

On propose de simuler (à nouveau) la loi géométrique de paramètre p à l'aide de sa propriété principale :

Proposition 1.3. *Soit $n \in \mathbb{N}$, $p \in]0, 1]$. Si $X_1, X_2, \ldots$ sont des variables aléatoires indépendantes, qui suivent toutes la loi de Bernoulli de paramètre p, alors $N = \min\{n \geq 1 \,|\, X_n = 1\}$ suit la loi géométrique de paramètre p.*

1. Grâce à cette proposition, définir une fonction `x=simul_geometrique2(p)` qui renvoie une réalisation d'une variable de loi $\mathcal{G}(p)$.

2. (Écrire dans `TP.m`) Remplir un tableau `tirage5` avec 10000 valeurs aléatoires indépendantes, de loi géométrique $\mathcal{G}(0.4)$.

2 Estimation de loi discrète par histogramme

On souhaite utiliser la méthode de Monte-Carlo pour s'assurer que les programmes écrits dans la première partie renvoient des résultats compatibles avec les lois de probabilités simulées.

On utilise d'abord une validation "graphique", à l'aide d'histogrammes, pour représenter la proportion de tirages égaux à chacune des valeurs possibles.

Par la loi des grands nombres, si $X_1, X_2, \ldots$ sont indépendantes et suivent la même loi $p = (p_n)_{n \in \mathbb{N}}$ sur $\mathbb{N}$, alors pour tout $k \in \mathbb{N}$,

$$\frac{\text{nombre de variables égales à } k \text{ parmi } X_1, \ldots, X_N}{N} \underset{N \to \infty}{\longrightarrow} p_k \qquad \text{presque sûrement.}$$

Autrement dit, l'histogramme converge, si on représente un **histogramme normalisé** : la hauteur de la barre centrée en k doit être égale à la *proportion* des données égales à k (par défaut, l'histogramme donne le *nombre* de telles données).

Pour obtenir un histogramme normalisé des données contenues dans un vecteur `v`, où les barres sont centrées au vecteur `centres` (par exemple, `centres=0:10`), on pourra utiliser :

```
nelem=hist(v,centres);  % nelem contient le nombre d'éléments dans chaque classe
N=length(v); % N est le nombre total d'éléments
bar(centres,nelem/N,1);
```

NB. Dans Octave, on peut aussi utiliser `hist(v,centres,1)` pour normaliser un histogramme de façon à sommer à 1.

Exercice 6* : Validation des tirages précédents

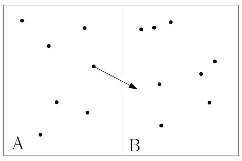
1. (Écrire dans `TP.m`) Représenter dans une même fenêtre graphique, mais sur différents axes (avec `subplot`), des histogrammes des vecteurs `tirage*` obtenus précédemment. On les représentera de façon à ce que, en ordonnée, apparaisse la **proportion** de tirages égaux à chaque valeur (et non le nombre de tirages). Pour `tirage3` et `tirage4`, on représentera en abscisses les valeurs de 1 à 10.

2. (Écrire dans `TP.m`) Modifier le script pour superposer aux graphes précédents les histogrammes des valeurs théoriques (on pourra utiliser `bar(centres,vp,0.5,'r')` : 0.5 représente la largeur relative la barre). Au besoin, corriger les programmes de la partie 1 qui semblent faux !

3. (Écrire dans `TP.m`) Pour chaque jeu de tirages, calculer et faire afficher dans la fenêtre de commande la somme des différences entre les p_k et les proportions de tirages égaux à k (données par `nelem(k)/N`), pour avoir une vision plus quantitative des erreurs.

3 Étude numérique de l'urne d'Ehrenfest

On s'intéresse à un modèle introduit par le couple de physiciens Ehrenfest au tout début du XX^e siècle pour l'expérience suivante : M molécules de gaz (en pratique, $M \simeq 10^{23}$...) sont réparties dans deux récipients A et B séparés par une cloison percée, et on observe l'évolution de la quantité de gaz dans le récipient A. Dans ce modèle, on représente les déplacements de molécules entre récipients de la façon suivante : après un temps fixé, une molécule choisie au hasard, uniformément parmi l'**ensemble** des molécules, et indépendamment de tout ce qui s'est déjà passé, change de récipient ; et ceci se répète indéfiniment.



Notons X_n le nombre de molécules dans A après n échanges. Alors $(X_n)_{n\geq 0}$ se définit par récurrence : X_0 est fixé (par exemple on peut choisir $X_0 = M$, c'est-à-dire que le récipient B est vide) et, pour tout $n \in \mathbb{N}$,

$$X_{n+1} = \begin{cases} X_n - 1 & \text{avec probabilité } \frac{X_n}{M} \\ X_n + 1 & \text{avec probabilité } 1 - \frac{X_n}{M}. \end{cases}$$

En effet, la molécule qui change de récipient se trouve d'abord dans A avec probabilité $\frac{X_n}{M}$ (il y a X_n molécules dans A, parmi les M molécules).

On souhaite étudier la composition de chaque récipient, au fil du temps : on peut s'attendre à un "équilibrage", mais au bout de combien de temps, et vers quel "équilibre" ? Comme l'expérience dépend de M et X_0, on notera $P_{M,X_0}(\cdot)$ et $E_{M,X_0}[\cdot]$ pour la probabilité et l'espérance relatives à cette expérience.

Exercice 7 : Convergence vers l'« équilibre »

Écrire les réponses (autres que les fonctions) dans un script `ex7.m`

1. Écrire une fonction `vx=trajectoire(x0,M,n)` qui simule une réalisation de la suite des positions (ou "trajectoire") $[X_1, \dots, X_n]$ partant de $X_0 =$`x0` et pour M molécules.

2. Représenter graphiquement le résultat pour $X_0 = M = 100$. Qu'observe-t-on ? (faire varier n). Superposer sur le même graphe 5 réalisation de trajectoires. Imposer `ylim([0 M])` pour voir les valeurs de 0 à M en ordonnée.

3. Écrire une fonction `t=temps_atteinte(x0,M,y)` qui simule $X_0, X_1, \dots$, partant de $X_0 =$`x0`, pour `M` molécules, jusqu'au temps d'atteinte de `y` :

$$T_{\mathrm{y}} = \min\{n > 0 \,|\, X_n = \mathrm{y}\},$$

et renvoie la valeur de T_{y}.

4. Par la méthode de Monte-Carlo (on prendra $N = 1000$ seulement), estimer $E_{M,M}[T_{M/2}]$ lorsque $M = 100$.

Exercice 8 : Distribution à l'« équilibre »

Écrire les réponses (autres que les fonctions) dans un script `ex8.m`

1. Dans un script séparé, `ex8A.m`, produire un vecteur `vx` contenant $N = 1000$ réalisations de X_{500} pour $X_0 = M = 100$.
NB. On utilise ici un script séparé pour éviter de recalculer `vx` à chaque question suivante.

2. Dans `ex8.m`, faire représenter un histogramme normalisé des valeurs obtenues. A priori, on prendra `centres=0:M`, mais on pourra adapter la plage aux valeurs significatives. Comment expliquer qu'une classe sur deux soit vide ?

3. Pour faire disparaître le phénomène observé, modifier `ex8A.m` pour que X_0 vale M avec probabilité $1/2$ et $M-1$ avec probabilité $1/2$ à chaque tirage, et refaire la question précédente.

Exercice 9 : Retour au point de départ

A priori, ce modèle est **réversible**, il est en particulier possible d'avoir $X_n = M$ (récipient B vide) pour $n > 0$, et on peut en fait facilement démontrer que $X_n = M$ pour une **infinité** de valeurs de n (cf. cours de MACS 2), ce qui pourtant n'est (heureusement) jamais observé en pratique : on n'observe **jamais** de "trou d'air" spontané dans une pièce ! Autrement dit, l'évolution observée est **irréversible**. Comment expliquer cette apparente contradiction (en supposant le modèle réaliste) ?

1. Par la méthode de Monte-Carlo, estimer la valeur de $E_{M,M}[T_M]$ lorsque $M = 10$ (NB. T_M est le temps de **retour** en M).

2. Plus généralement, représenter graphiquement les estimations (avec $N = 100$ seulement) de $E_{M,M}[T_M]$ lorsque $M = 2, \ldots, 10$. Quelle est votre conclusion ?