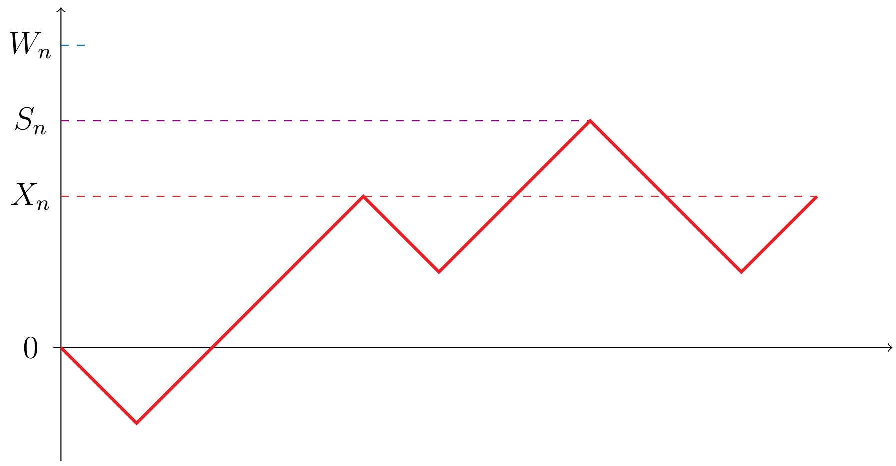
Exercice 1 : Processus de Bessel discret.

Soit $(A_n)_{n\in\mathbb{N}}$ une suite de variables indépendantes de loi $\mathbb{P}[A_n = 1] = \mathbb{P}[A_n = -1] = \frac{1}{2}$; et $X_n = A_1 + A_2 + \cdots + A_n$. On rappelle que la marche aléatoire $(X_n)_{n\in\mathbb{N}}$ est une chaîne de Markov sur $\mathbb{Z}$, qui est irréductible récurrente nulle ; on pourra réutiliser ce fait sans démonstration. On note

$$\begin{aligned} S_n &= \max\{X_k,\ k \in [\![0, n]\!]\}; \\ W_n &= 2S_n - X_n; \\ V_n &= (W_n, X_n). \end{aligned}$$



1. Comparer X_n, S_n et W_n. Montrer que $(S_n)_{n\in\mathbb{N}}$ et $(W_n)_{n\in\mathbb{N}}$ tendent vers l'infini presque sûrement. Si $W_n = k$, montrer que les valeurs possibles pour X_n sont

   $$I_k = \{-k, -k+2, -k+4, \ldots, k-2, k\}.$$

   Pour tout $l \in I_k$, dessiner une trajectoire possible pour $(X_n)_{n\in\mathbb{N}}$ telle que $W_n = k$ et $X_n = l$.

2. On suppose que $W_n = X_n = k$. Montrer que $V_n = (k, k)$, et que la loi de V_{n+1} sachant $V_n = (k, k)$ est

   $$\mathbb{P}[V_{n+1} = (k+1, k+1) \,|\, V_n = (k, k)] = \mathbb{P}[V_{n+1} = (k+1, k-1) \,|\, V_n = (k, k)] = \frac{1}{2}.$$

   Si $k = W_n > X_n = l$, calculer également la loi de V_{n+1} sachant $V_n = (k, l)$. En déduire que $(V_n)_{n\in\mathbb{N}}$ est une chaîne de Markov, dont on précisera l'espace des états et la matrice de transition.

3. Rappeler la valeur de $\mathbb{P}[X_n = l]$ (compter les chemins de taille n qui relient 0 à l). Si $k \geq l$ a la même parité que l, montrer que

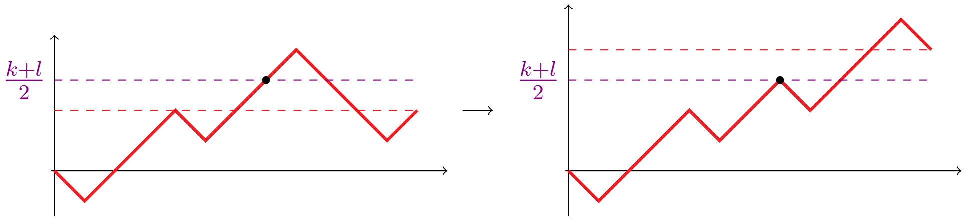
   $$\mathbb{P}\left[S_n \geq \frac{k+l}{2} \text{ et } X_n = l\right] = \mathbb{P}[X_n = k].$$

On pourra utiliser la transformation par réflexion suivante, qui établit une bijection entre

$$\left\{\text{trajectoires de taille } n \text{ dépassant } \frac{k+l}{2} \text{ et finissant en } l\right\}$$

et

$$\{\text{trajectoires de taille } n \text{ finissant en } k\} :$$



Sur le dessin, le point marqué est le premier temps où la trajectoire dépasse la valeur $\frac{k+l}{2}$, et on réfléchit la trajectoire après ce point.

4. Déduire de la question précédente que si $l \in I_k$, alors
$$\mathbb{P}[W_n = k \text{ et } X_n = l] = \mathbb{P}\left[S_n = \frac{k+l}{2} \text{ et } X_n = l\right] = \mathbb{P}[X_n = k] - \mathbb{P}[X_n = k+2].$$
Expliciter le terme de droite de cette formule.

5. Donner une formule pour $\mathbb{P}[W_n = k]$, puis pour $\mathbb{P}[X_n = l \,|\, W_n = k]$, avec $l \in I_k$. Montrer que la loi conditionnelle de X_n sachant $W_n = k$ est la loi uniforme sur I_k.

6. On admet le résultat plus général suivant : si $(w_0, \ldots, w_n = k)$ est une suite d'entiers positifs, avec $\mathbb{P}[W_0 = w_0, \ldots, W_n = w_n] \neq 0$, alors
$$\mathbb{P}[X_n = l \,|\, W_0 = w_0, \ldots, W_n = w_n = k] = \begin{cases} \frac{1}{k+1} & \text{si } l \in I_k, \\ 0 & \text{sinon.} \end{cases}$$
Autrement dit, conditionnellement à W_n, la loi de X_n est indépendante du reste $(W_0, \ldots, W_{n-1})$ de la trajectoire de W, et uniforme sur I_{W_n}. Avec ce résultat, montrer que $(W_n)_{n\in\mathbb{N}}$ est une chaîne de Markov sur $\mathbb{N}$, de matrice de transition
$$P(k, k+1) = \frac{k+2}{2(k+1)} \quad ; \quad P(k, k-1) = \frac{k}{2(k+1)}.$$
On pourra décomposer la probabilité $\mathbb{P}[W_0 = w_0, \ldots, W_n = k, W_{n+1} = k \pm 1]$ en fonction des valeurs de X_n et X_{n+1}.

Exercice 2 : configurations planaires du modèle *hard-core*.

Pour cet exercice, les programmes peuvent être écrits dans n'importe quel langage, ou en pseudo-code.

Soit $N \geq 1$ un entier, et $S_N = [\![1, N]\!] \times [\![1, N]\!]$ la grille planaire de taille $N \times N$. Deux points (i, j) et (i', j') sont voisins dans S_N si $i = i'$ et $|j - j'| = 1$, ou si $j = j'$ et $|i - i'| = 1$. Une configuration admissible sur S_N est une fonction $\sigma : S_N \to \{+1, 0, -1\}$, telle que si $\sigma(i, j) = +1$, alors aucun voisin (i', j') de (i, j) ne vérifie $\sigma(i', j') = -1$. Une configuration admissible représente la répartition de particules de deux types ($+1$ et -1) dans le plan, telle que deux particules $+1$ et -1 ne peuvent pas être voisines. On note X_N l'ensemble des configurations admissibles de taille N.

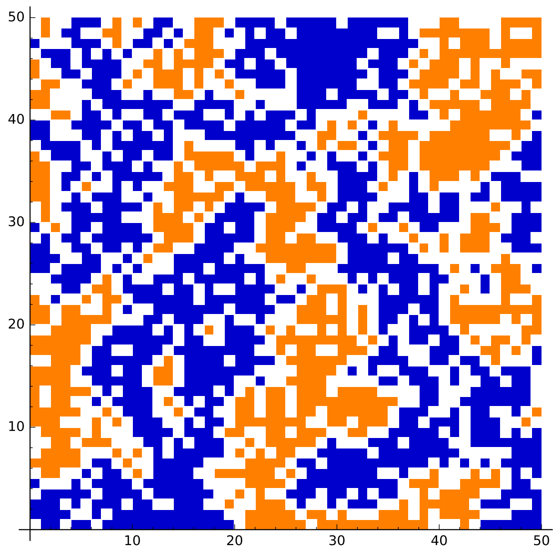


FIGURE 1 – Une configuration admissible de taille 50×50, avec les particules $+1$ en bleu et les particules -1 en orange.

1. Montrer que
$$2^{N^2} \leq \operatorname{card} X_N \leq 3^{N^2}.$$
Écrire un programme `ConfigurationAdmissible` qui prend en argument une matrice de taille $N \times N$ constituée de 0, de $+1$ et de -1, et qui vérifie si c'est une configuration admissible. Écrire un autre programme `DessinerConfiguration` qui dessine une configuration.

On équipe X_N de la probabilité suivante :
$$\mu_q(\sigma) = \frac{q^{A(\sigma)}}{\sum_{\sigma \in X_N} q^{A(\sigma)}} = \frac{q^{A(\sigma)}}{Z_N(q)},$$
où $q > 0$ est un paramètre fixé, et $A(\sigma)$ est le nombre de particules $+1$ ou -1 de la configuration σ, qui est aussi $A(\sigma) = \sum_{(i,j) \in S_N} |\sigma(s)|$.

2. Soit P la matrice de transition d'une chaîne de Markov d'espace d'états X_N. On suppose que P a la propriété suivante :
$$\forall \sigma, \sigma' \in X_n,\ \ \mu_q(\sigma)\, P(\sigma, \sigma') = \mu_q(\sigma')\, P(\sigma', \sigma).$$
Montrer que μ_q est une mesure de probabilité invariante pour la chaîne de Markov de matrice P.
3. Si $(i,j) \in S_N$, $t \in \{0, +1, -1\}$ et $\sigma \in X_N$, on note $\sigma_{(i,j),t}$ la configuration qui ne diffère de σ qu'en (i,j), avec $\sigma_{(i,j),t}(i,j) = t$. Cette nouvelle configuration n'est plus forcément admissible. On introduit ensuite l'application
$$\begin{aligned} F_{(i,j),t} : X_N &\to X_N \\ \sigma &\mapsto \begin{cases} \sigma_{(i,j),t} & \text{si } \sigma_{(i,j),t} \text{ est admissible,} \\ \sigma & \text{sinon.} \end{cases} \end{aligned}$$

Soit $(I_n, J_n)_{n\in\mathbb{N}}$ une suite de points aléatoires indépendants dans S_N et de loi uniforme : $\mathbb{P}[(I_n, J_n) = (i,j)] = \frac{1}{N^2}$. On considère également une suite $(T_n)_{n\in\mathbb{N}}$ d'éléments indépendants dans $\{0, +1, -1\}$, avec

$$\mathbb{P}[T_n = 0] = \frac{1}{1+2q} \qquad ; \qquad \mathbb{P}[T_n = +1] = \mathbb{P}[T_n = -1] = \frac{q}{1+2q}.$$

On définit une suite aléatoire $(\sigma_n)_{n\in\mathbb{N}}$ dans X_N par

$$\sigma_0 = 0 \text{ (configuration dont tous les sites valent 0)} \qquad ; \qquad \sigma_{n+1} = F_{(I_n,J_n),T_n}(\sigma_n).$$

Montrer que $(\sigma_n)_{n\in\mathbb{N}}$ est une chaîne de Markov sur X_N, et expliciter sa matrice de transition.

4. Montrer que la chaîne de Markov précédemment construite est irréductible récurrente positive sur X_N, et que sa mesure invariante est la loi μ_q.
5. Si n est assez grand, que peut-on dire de la loi de la configuration aléatoire σ_n ? Écrire un programme `MarkovConfiguration(N,n,q)` qui calcule et affiche une configuration aléatoire σ_n obtenue par la chaîne de Markov (indication : lorsqu'on vérifie à chaque étape si $(\sigma_n)_{(I_n,J_n),T_n}$ est admissible, il suffit de vérifier le voisinage du point (I_n, J_n), et on n'a pas besoin de regarder toute la grille). Dessiner pour $N = 30$ et $n = 100000$ des configurations σ_n de paramètre $q = 0.5$, $q = 1$, $q = 2$ et $q = 4$. Commenter.

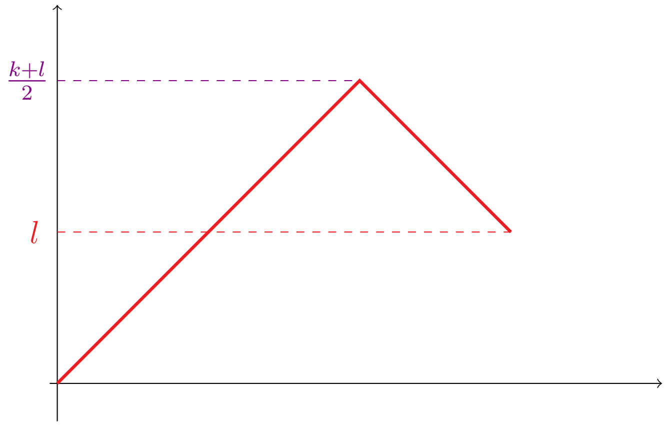
Corrigé de l'exercice 1.

1. Comme $S_n = \max_{k \in [\![0,n]\!]} X_k$, $S_n \geq X_n$. On en déduit que $W_n = S_n + (S_n - X_n) \geq S_n$, et on a aussi $W_n \geq -X_n$, donc :

$$-W_n \leq X_n \leq S_n \leq W_n.$$

Comme $(X_n)_{n \in \mathbb{N}}$ est irréductible récurrente, elle visite tous les états presque sûrement, donc ne reste pas bornée supérieurement ; ceci implique $\lim_{n \to \infty} S_n = +\infty$ presque sûrement. Comme $W_n \geq S_n$, on a également $\lim_{n \to \infty} W_n = +\infty$.

Supposons $W_n = k$. Comme $W_n = 2S_n - X_n$, W_n a la même parité que X_n. De plus, $X_n \in [\![-W_n, W_n]\!]$, donc si $W_n = k$, alors $X_n \in I_k = \{-k, -k+2, \ldots, k-2, k\}$. Toutes ces valeurs peuvent être atteintes, car si $l \in I_k$, alors le chemin



vérifie bien $W_n = k$ et $X_n = l$.

2. Si $W_n = X_n = k$, alors $V_n = (W_n, X_n) = (k, k)$, et $S_n = k$: au temps n, la marche aléatoire atteint son maximum. Fixons des vecteurs $v_0, v_1, \ldots, v_n$ dans

$$\mathfrak{X} = \bigsqcup_{k=0}^{\infty} \{(k,l) \,|\, l \in I_k\},$$

avec $v_n = (k, k)$, et $\mathbb{P}[V_0 = v_0, \ldots, V_n = v_n] \neq 0$. Sachant $V_n = (k, k)$, si $A_{n+1} = 1$, alors $X_{n+1} = k+1$ et la marche aléatoire atteint un nouveau maximum au temps $n+1$, égal à $k+1$. On a donc $S_{n+1} = k+1$, et $W_{n+1} = 2(k+1) - (k+1) = k+1$. Comme A_{n+1} est indépendant de $\mathcal{F}_n = \sigma(A_1, \ldots, A_n) = \sigma(V_1, \ldots, V_n)$,

$$\mathbb{P}[V_{n+1} = (k+1, k+1) \,|\, V_0 = v_0,\, V_1 = v_1, \ldots, V_n = (k,k)] = \mathbb{P}[A_{n+1} = 1] = \frac{1}{2}.$$

De même, si $A_{n+1} = -1$, alors $X_{n+1} = k-1$ et la marche aléatoire garde le même maximum, donc $S_{n+1} = k$ et $W_{n+1} = 2k - (k-1) = k+1$. On obtient donc

$$\mathbb{P}[V_{n+1} = (k+1, k-1) \,|\, V_0 = v_0,\, V_1 = v_1, \ldots, V_n = (k,k)] = \mathbb{P}[A_{n+1} = -1] = \frac{1}{2}.$$

Supposons maintenant $V_n = (k, l)$ avec $k > l$. Alors, $S_n = \frac{k+l}{2} > l$, donc la marche aléatoire n'atteint pas son maximum au temps n. Au temps suivant, elle ne pourra pas dépasser le niveau $\frac{k+l}{2}$ (au mieux, elle pourra de nouveau atteindre ce maximum), donc $S_{n+1} = S_n = \frac{k+l}{2}$. De plus, si $A_{n+1} = 1$, alors $X_{n+1} = l+1$ et $W_{n+1} = k-1$, tandis que si $A_{n+1} = -1$, $X_{n+1} = l-1$ et $W_{n+1} = k+1$. En utilisant de nouveau l'indépendance de A_{n+1} et de la tribu $\mathcal{F}_n$, on conclut que

$$\mathbb{P}[V_{n+1} = (k-1, l+1) \,|\, V_0 = v_0,\, V_1 = v_1, \ldots, V_n = (k,l)] = \mathbb{P}[A_{n+1} = 1] = \frac{1}{2};$$
$$\mathbb{P}[V_{n+1} = (k+1, l-1) \,|\, V_0 = v_0,\, V_1 = v_1, \ldots, V_n = (k,l)] = \mathbb{P}[A_{n+1} = -1] = \frac{1}{2}.$$

On a donc montré que $(V_n)_{n\in\mathbb{N}}$ était une chaîne de Markov d'espace d'états $\mathfrak{X} = \bigsqcup_{k=0}^{\infty} \{(k,l) \,|\, l \in I_k\}$, et de matrice de transition

$$\forall k \in \mathbb{N},\; P((k,k),(k+1,k+1)) = P((k,k),(k+1,k-1)) = \frac{1}{2};$$
$$\forall k \in \mathbb{N},\, \forall l \in I_k \setminus \{k\},\;\; P((k,l),(k-1,l+1)) = P((k,l),(k+1,l-1)) = \frac{1}{2},$$

toutes les autres valeurs de la matrice de transition étant nulles.

3. Toutes les trajectoires de taille n dont les pas valent ± 1 ont même probabilité $\frac{1}{2^n}$, donc
$$\mathbb{P}[X_n = l] = \frac{1}{2^n} \,\mathrm{card}\,\{\text{trajectoires de taille } n \text{ reliant } 0 \text{ à } l\}.$$
Si un chemin de taille n relie 0 à l, alors il compte $\frac{n+l}{2}$ pas vers le haut et $\frac{n-l}{2}$ pas vers le bas, et il est entièrement déterminé par le choix des $\frac{n+l}{2}$ pas vers le haut parmi n pas. Donc,
$$\mathbb{P}[X_n = l] = \begin{cases} \frac{1}{2^n}\binom{n}{\frac{n+l}{2}} & \text{si } l \in I_n, \\ 0 & \text{sinon.} \end{cases}$$
Calculons maintenant $\mathbb{P}[S_n \geq \frac{k+l}{2}$ et $X_n = l]$, qui est égal à $\frac{1}{2^n}$ fois le nombre de chemins de taille n qui relient 0 à l en dépassant à un moment le palier $\frac{k+l}{2}$. Si l'on réfléchit un chemin valide après le premier passage en $\frac{k+l}{2}$, on obtient un chemin reliant 0 à
$$\frac{k+l}{2} + \left(\frac{k+l}{2} - l\right) = k.$$
Réciproquement, tout chemin reliant 0 à k passe par $\frac{k+l}{2}$, et donne par réflexion après ce premier passage un chemin valide. Donc,
$$\mathbb{P}\left[S_n \geq \frac{k+l}{2} \text{ et } X_n = l\right] = \frac{1}{2^n} \,\mathrm{card}\,\{\text{trajectoires de taille } n \text{ reliant } 0 \text{ à } k\}$$
$$= \mathbb{P}[X_n = k].$$

4. On déduit du calcul précédent la loi de (S_n, X_n) :
$$\mathbb{P}\left[S_n = \frac{k+l}{2} \text{ et } X_n = l\right]$$
$$= \mathbb{P}\left[S_n \geq \frac{k+l}{2} \text{ et } X_n = l\right] - \mathbb{P}\left[S_n \geq \frac{k+l}{2} + 1 \text{ et } X_n = l\right]$$
$$= \mathbb{P}[X_n = k] - \mathbb{P}[X_n = k+2],$$

en supposant k et l de même parité. Cette quantité vaut

$$\frac{1}{2^n}\left(\binom{n}{\frac{n+k}{2}} - \binom{n}{\frac{n+k}{2}+1}\right) = \frac{1}{2^n}\,\frac{k+1}{n+1}\binom{n+1}{\frac{n+k}{2}+1},$$

et c'est aussi $\mathbb{P}[W_n = k \text{ et } X_n = l]$ (si l'on connaît S_n et X_n, on en déduit $V_n = (W_n, X_n)$, et réciproquement). On a donc trouvé la loi de V_n ; notons que le résultat ne dépend pas de l (si $l \in I_k$).

5. En sommant sur les valeurs possibles de X_n sachant $W_n = k$, on obtient

$$\mathbb{P}[W_n = k] = (\operatorname{card} I_k)\,\mathbb{P}[W_n = k \text{ et } X_n = l] = \frac{1}{2^n}\,\frac{(k+1)^2}{n+1}\binom{n+1}{\frac{n+k}{2}+1}.$$

Puis,

$$\mathbb{P}[X_n = l \,|\, W_n = k] = \frac{\mathbb{P}[W_n = k \text{ et } X_n = l]}{\mathbb{P}[W_n = k]} = \begin{cases} \frac{1}{k+1} & \text{si } l \in I_k, \\ 0 & \text{sinon.} \end{cases}$$

La loi de X_n sachant $W_n = k$ est donc uniforme sur l'ensemble I_k.

6. Soit $w_0, w_1, \ldots, w_n$ des éléments de $\mathbb{N}$, tels que $\mathbb{P}[W_0 = w_0, \ldots, W_n = w_n] \neq 0$ et $w_n = k$. On calcule $\mathbb{P}[W_{n+1} = k \pm 1 \,|\, W_0 = w_0, \ldots, W_n = w_n]$ en décomposant les probabilités en fonction des valeurs x_n et $x_{n+1} = x_n \pm 1$ de X_n et X_{n+1} :

$$\begin{aligned}
&\mathbb{P}[W_0 = w_0, \ldots, W_n = k, W_{n+1} = k \pm 1] \\
&= \sum_{x_n \in I_k} \mathbb{P}[W_0 = w_0, \ldots, W_{n-1} = w_{n-1}, V_n = (k, x_n)] \times \\
&\qquad (P((k, x_n), (k \pm 1, x_n + 1)) + P((k, x_n), (k \pm 1, x_n - 1))) \\
&= \frac{1}{k+1} \sum_{x_n \in I_k} \mathbb{P}[W_0 = w_0, \ldots, W_{n-1} = w_{n-1}, W_n = k] \times \\
&\qquad (P((k, x_n), (k \pm 1, x_n + 1)) + P((k, x_n), (k \pm 1, x_n - 1))).
\end{aligned}$$

Par conséquent,

$$\begin{aligned}
&\mathbb{P}[W_{n+1} = k \pm 1 \,|\, W_0 = w_0, \ldots, W_n = w_n] \\
&= \frac{1}{k+1} \sum_{x_n \in I_k} (P((k, x_n), (k \pm 1, x_n + 1)) + P((k, x_n), (k \pm 1, x_n - 1))).
\end{aligned}$$

Pour l'événement $W_{n+1} = k + 1$, il y a $k + 2$ paires $((k, x_n), (k + 1, x_n \pm 1))$ qui contribuent à la somme (chacune pour $\frac{1}{2}$), à savoir $((k, k), (k+1, k+1))$, $((k, k), (k+1, k-1))$, et tous les $((k, l), (k+1, l-1))$ avec $l \in I_k \setminus \{k\}$. Donc,

$$\mathbb{P}[W_{n+1} = k + 1 \,|\, W_0 = w_0, \ldots, W_n = w_n] = \frac{k+2}{2(k+1)}.$$

De même, pour l'événement $W_{n+1} = k - 1$, il y a k paires $((k, x_n), (k - 1, x_n \pm 1))$ qui contribuent à la somme, à savoir tous les $((k, l), (k - 1, l + 1))$ avec $l \in I_k \setminus \{k\}$. Donc,

$$\mathbb{P}[W_{n+1} = k - 1 \,|\, W_0 = w_0, \ldots, W_n = w_n] = \frac{k}{2(k+1)}.$$

On conclut que $(W_n)_{n\in\mathbb{N}}$ est une chaîne de Markov sur $\mathbb{N}$, de matrice de transition

$$P(k, k+1) = \frac{k+2}{2(k+1)} \qquad ; \qquad P(k, k-1) = \frac{k}{2(k+1)}.$$

Corrigé de l'exercice 2.

1. Toutes les configurations avec seulement des $+1$ et des 0 sont admissibles, donc on a au moins $2^{\operatorname{card} S_N} = 2^{N^2}$ configurations admissibles. D'autre part, le nombre total de configurations (admissibles ou non) est 3^{N^2}, puisqu'on a 3 choix possibles pour chaque site. Ainsi,
$$2^{N^2} \leq \operatorname{card} S_N \leq 3^{N^2}.$$
On définit en Sage une classe `Configuration`, à laquelle on ajoutera au fur et à mesure des méthodes :

```
class Configuration:
    def __init__(self,n):
        self.size = n
        self.matrix = matrix(ZZ,n,n)
```

Pour tester si une configuration est admissible, on a besoin de savoir identifier les voisins de chaque site. La méthode suivante, rajoutée à la définition de la classe, renvoie la liste des voisins d'un site (en prenant en compte les effets de bord) :

```
def neighbors(self,site):
    res = []
    n = self.size
    if site[0] < n-1:
        res.append((site[0]+1,site[1]))
    if site[0] > 0:
        res.append((site[0]-1,site[1]))
    if site[1] < n-1:
        res.append((site[0],site[1]+1))
    if site[1] > 0:
        res.append((site[0],site[1]-1))
    return res
```

On teste alors si une configuration est admissible en vérifiant un à un les sites (on effectue plus de vérifications que nécessaire, mais ce n'est pas très grave) :

```
def is_admissible(self):
    res = 0
    for i in range(self.size):
        for j in range(self.size):
            if (-1 in [self.matrix[i,j]*self.matrix[n[0],n[1]]
                               for n in self.neighbors((i,j))]):
                res += 1
    if res == 0:
        return True
    else:
        return False
```

Par exemple, le code suivant

```
C = Configuration(2) ; C.matrix[0,0] = 1 ; C.matrix[0,1] = 1
C.is_admissible()
```

renvoie `True`, tandis que le code

```
C = Configuration(2) ; C.matrix[0,0] = 1 ; C.matrix[0,1] = -1
C.is_admissible()
```

renvoie `False`. Pour dessiner une configuration, on ajoute à la définition de la classe la méthode :

```
def plot(self):
    G = Graphics()
    for i in range(self.size):
        for j in range(self.size):
            if self.matrix[i,j] == 1:
                G += polygon([(i,j),(i,j+1),(i+1,j+1),
                        (i+1,j)],rgbcolor=(0,0,0.8))
            elif self.matrix[i,j] == -1:
                G += polygon([(i,j),(i,j+1),(i+1,j+1),
                        (i+1,j)],rgbcolor=(1,0.5,0))
    G.show(axes=False)
```

2. Montrons que sous la condition de réversibilité, la loi μ_q est invariante pour la matrice P. On calcule

$$(\mu_q P)(\sigma) = \sum_{\sigma' \in X_N} \mu_q(\sigma')\, P(\sigma',\sigma) = \sum_{\sigma' \in X_N} \mu_q(\sigma)\, P(\sigma,\sigma')$$
$$= \mu_q(\sigma) \left(\sum_{\sigma' \in X_N} P(\sigma,\sigma') \right) = \mu_q(\sigma) \times 1 = \mu_q(\sigma).$$

On a donc bien une loi invariante sous P.

3. Le processus aléatoire $(\sigma_n)_{n\in\mathbb{N}}$ vérifie une équation de récurrence du type $\sigma_{n+1} = F(\sigma_n, \xi_n)$, où les aléas $\xi_n = (I_n, J_n, T_n)$ sont indépendants et identiquement distribués. Par le théorème de représentation des chaînes de Markov, on en déduit que $(\sigma_n)_{n\in\mathbb{N}}$ est une chaîne de Markov sur X_N. De plus, sa matrice de transition est

$$P(\sigma,\sigma') = \mathbb{P}[\{\xi \,|\, F(\sigma,\xi) = \sigma'\}].$$

Par définition, pour tout $\xi \in S_N \times \{0,+1,-1\}$, $F(\sigma,\xi) = F_\xi(\sigma)$ ne peut être égal qu'à σ ou à une configuration admissible σ' qui diffère de σ en un seul site. Dans ce dernier cas, il y a un seul $\xi = (i,j,t)$ tel que $F_\xi(\sigma) = \sigma'$, et $\sigma'(i,j) = t$. La probabilité pour tirer au hasard cet élément ξ est $\frac{1}{N^2}\,\frac{1}{1+2q}$ si $t = 0$, et $\frac{1}{N^2}\,\frac{q}{1+2q}$ si $t = 1$ ou $t = -1$. Ainsi, pour toutes configurations admissibles σ et σ',

$$P(\sigma,\sigma') = \begin{cases} \frac{1}{N^2(1+2q)} & \text{si } \sigma' \text{ diffère de } \sigma \text{ en un seul site } (i,j), \text{ avec } \sigma'(i,j) = 0, \\ \frac{q}{N^2(1+2q)} & \text{si } \sigma' \text{ diffère de } \sigma \text{ en un seul site } (i,j), \text{ avec } \sigma'(i,j) = \pm 1, \\ C_\sigma & \text{si } \sigma' = \sigma, \end{cases}$$

avec

$$C_\sigma = 1 - \sum_{\sigma' \neq \sigma \text{ en un seul site}} P(\sigma, \sigma').$$

4. À partir de n'importe quelle configuration σ, on peut atteindre la configuration 0 en changeant tous les sites un à un en la valeur 0 (ceci ne crée pas de configuration non-admissible). Par conséquent, pour toutes configurations σ et σ', on peut trouver des transitions reliant σ à 0, puis 0 à σ' ; la chaîne est donc irréductible, et comme elle est finie, elle est récurrente positive. Pour montrer que sa mesure invariante est μ_q, il suffit de vérifier la condition de réversibilité. Soit σ et σ' deux configurations admissibles dans X_N ; pour montrer que $\mu_q(\sigma)\, P(\sigma, \sigma') = \mu_q(\sigma')\, P(\sigma', \sigma)$, on peut supposer $\sigma \neq \sigma'$ car ce cas est trivial, et aussi $P(\sigma, \sigma') \neq 0$, c'est-à-dire que σ et σ' ne diffèrent qu'en un seul site (i, j). Notons alors que $P(\sigma, \sigma') = \frac{1}{N^2} \frac{q^{|\sigma'(i,j)|}}{1+2q}$, de sorte que
$$\mu_q(\sigma)\, P(\sigma, \sigma') = \frac{1}{Z_N(q)\, N^2\, (1 + 2q)}\, q^{\sum_{s \in S_N} |\sigma(s)| + |\sigma'(i,j)|}.$$
De même,
$$\mu_q(\sigma')\, P(\sigma', \sigma) = \frac{1}{Z_N(q)\, N^2\, (1 + 2q)}\, q^{\sum_{s \in S_N} |\sigma'(s)| + |\sigma(i,j)|},$$
et l'égalité des deux quantités vient alors de
$$\begin{aligned}\sum_{s \in S_N} |\sigma(s)| + |\sigma'(i,j)| &= \sum_{s \neq (i,j)} |\sigma(s)| + |\sigma(i,j)| + |\sigma'(i,j)| \\ &= \sum_{s \neq (i,j)} |\sigma'(s)| + |\sigma(i,j)| + |\sigma'(i,j)| = \sum_{s \in S_N} |\sigma'(s)| + |\sigma(i,j)|.\end{aligned}$$

5. Par le théorème ergodique, si π_n désigne la loi de σ_n, alors $\pi_n \rightharpoonup_{n \to \infty} \mu_q$, donc σ_n a presque pour loi μ_q lorsque n est grand. On peut donc utiliser la chaîne de Markov pour (presque) simuler la loi μ_q. Concrètement, les deux méthodes
```
def random_site(self):
    n = self.size
    return (floor(n*random()),floor(n*random()))
```
et
```
def random_sign(self,q):
        alea = random()
        if alea < q/(1+2*q):
                return 1
        elif alea < (2*q)/(1+2*q):
                return -1
        else:
                return 0
```
tirent au hasard un site d'une configuration, et un signe dans $\{0, +1, -1\}$. La méthode

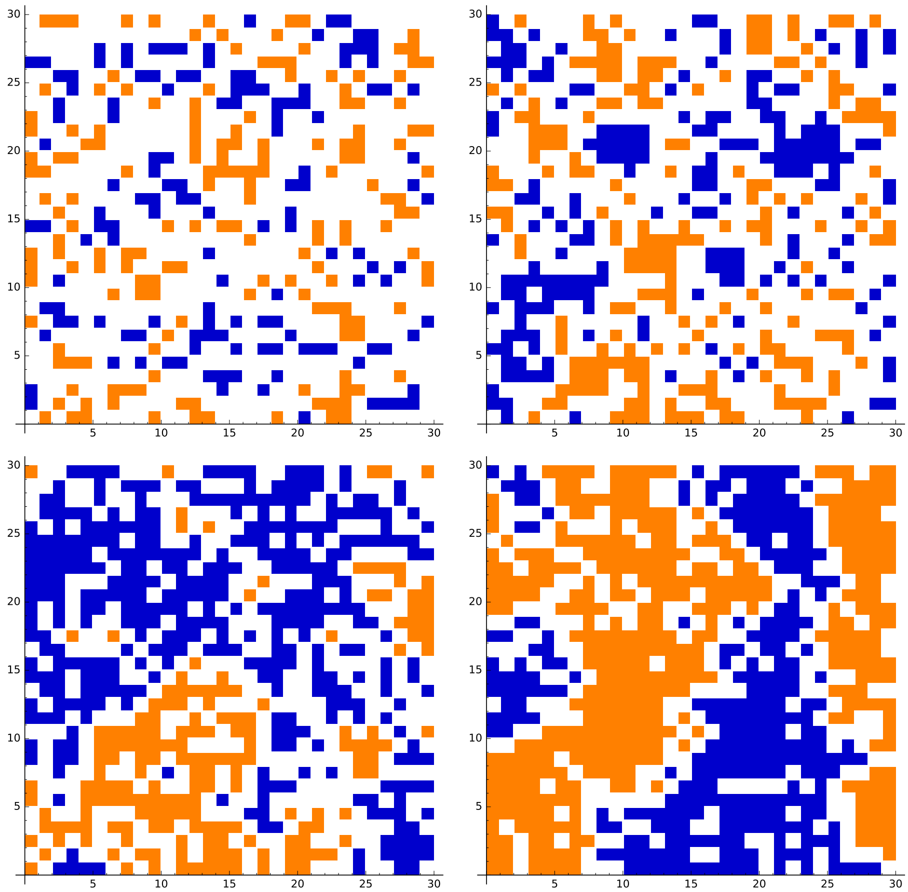
```
def transform(self,site,sign):
    if sign == 1:
        if not (-1 in [self.matrix[n[0],n[1]] for
                        n in self.neighbors(site)]):
            self.matrix[site[0],site[1]] = 1
    elif sign == -1:
        if not (1 in [self.matrix[n[0],n[1]] for
                        n in self.neighbors(site)]):
            self.matrix[site[0],site[1]] = -1
    else:
        self.matrix[site[0],site[1]] = 0
```

correspond à la transformation $F_{(i,j),t}$, et finalement,

```
def markov(self,n,q):
    for i in range(n):
        self.transform(self.random_site(),self.random_sign(q))
```

applique n transitions de la chaîne de Markov de paramètre q à une configuration. On peut donc simuler σ_n avec :

```
def MarkovConfiguration(N,n,q):
    C = Configuration(N)
    C.markov(n,q)
    return C
```



Lorsque q augmente, la mesure μ_q charge de plus en plus les configurations qui ont beaucoup de particules et peu de sites laissés vacants (les sites 0). Comme les sites $+1$ et -1 ne peuvent pas être contigus, pour q assez grand, ceci restreint le nombre de zones $+1$ et -1 disjointes, car elles doivent être séparées par des sites vacants. Ainsi, pour q suffisamment grand, une configuration typique sous μ_q contient un petit nombre de larges domaines $+1$ ou -1, et très peu de sites vacants entre ces domaines. On a dessiné ci-dessous un exemple avec $N = 50$ et $q = 100$: