

THÉORIE NEUTRE UNIFIÉE DE LA BIODIVERSITÉ – CORRIGÉ

I.1 Commençons par détailler les faiblesses du modèle. Les hypothèses du modèle neutre donnent un même poids à tout individu d'un écosystème, indépendamment de son espèce. Autrement dit, on donne la même probabilité de survie à un chêne millénaire et à un scarabée, ce qui est clairement une limite du modèle. D'autre part, le modèle neutre n'autorise pas de fluctuation du nombre total d'individus au cours du temps. Cette invariance est vérifiée expérimentalement sur de grands domaines, mais elle n'est évidemment plus vraie pour un espace géographique de taille restreinte, où des individus peuvent immigrer ou émigrer. En réalité, la théorie unifiée de la biodiversité permet de prendre en compte ces migrations, en divisant l'écosystème (aussi appelé méta-communauté) en plus petites régions (communautés locales) ; mais le texte ne traite pas de cet aspect et de son influence sur la répartition des individus en différentes espèces. Enfin, le modèle neutre tel qu'il est présenté ici ne s'intéresse pas du tout aux mécanismes précis de reproduction des individus, et il propose une vision extrêmement simplifiée (et clairement fausse) de ces mécanismes (une mort et une naissance à chaque instant observé).

L'intérêt principal du modèle neutre est de proposer une explication simple à la répartition "universelle" des individus d'un écosystème en diverses espèces (la courbe en "S" dessinée au début du texte). Il permet d'extraire pour chaque type d'écosystème un unique paramètre de biodiversité θ , qui régit la répartition par espèces indépendamment de la taille N du système considéré. Il donne par ailleurs une explication dynamique à cette répartition par espèces (les processus aléatoires $(c(t))_{t \in \mathbb{N}}$ et $(p(t))_{t \in \mathbb{N}}$).

I.2 Pour montrer que \mathfrak{C}_N^* a pour cardinal 2^{N-1} , on peut procéder par récurrence sur N , le cas $N = 1$ étant trivial puisque $\mathfrak{C}_1^* = \{(1)\}$. Supposons établi le résultat au rang N , et considérons une composition $c = (N_1, N_2, \dots, N_l)$ de taille

$$N_1 + N_2 + \dots + N_l = N + 1.$$

Si $N_l = 1$, on peut associer à c la composition $c' = (N_1, N_2, \dots, N_{l-1})$ de taille N . D'autre part, si $N_l \geq 2$, on peut associer à c la composition $c' = (N_1, N_2, \dots, N_l - 1)$ de taille N . L'application $c \in \mathfrak{C}_{N+1}^* \mapsto c' \in \mathfrak{C}_N^*$ est surjective, et elle atteint toute composition $c' \in \mathfrak{C}_N^*$ deux fois : on peut retrouver c à partir de c' soit en rajoutant 1 à la dernière part de c' , soit en rajoutant une part de taille 1 à la fin de c' . On conclut que

$$\text{card } \mathfrak{C}_{N+1}^* = 2(\text{card } \mathfrak{C}_N^*) = 2^N,$$

d'où le résultat au rang $N + 1$.

I.3 Le processus $(R(t))_{t \in \mathbb{N}}$ est une marche aléatoire sur \mathbb{N} avec

$$\begin{aligned} \mathbb{P}[R(t+1) - R(t) = 1] &= \nu \\ \mathbb{P}[R(t+1) - R(t) = 0] &= 1 - \nu, \end{aligned}$$

et les variables $B(t) = R(t+1) - R(t)$, $t \geq 0$ qui sont indépendantes (et de même loi). En effet, on rajoute une part à $c(t)$ pour obtenir $c(t+1)$ lorsqu'on applique l'étape 2 de l'algorithme, ce qui arrive avec probabilité ν ; et sinon $c(t)$ et $c(t+1)$ ont le même nombre de parts.

Par la loi des grands nombres, $\frac{R(t)-R(0)}{t} = \frac{1}{t} \sum_{i=1}^t B(i) \rightarrow \nu$ presque sûrement, et de même, $\frac{R(t)}{t} \rightarrow \nu$ presque sûrement lorsque t tend vers l'infini.

- I.4 Dans la suite, les programmes écrits fonctionnent dans **Sage**, et à quelques modifications près, ils devraient également fonctionner dans **Python**. On utilisera le programme suivant pour choisir une part d'une composition $c = (N_1, N_2, \dots, N_r)$ de taille N , la part i étant choisie avec probabilité $\frac{N_i}{N}$:

```
def choose_part(c):
    N = sum(c)
    count = 0
    res = -1
    alea = random()
    while count < alea:
        res +=1
        count += c[res]/N
    return res
```

Le programme suivant prend en argument la taille N de la population, le paramètre ν , un temps T , et en argument optionnel une répartition initiale $c0$ (le programme utilise sinon comme répartition initiale une composition aléatoire de taille N). Il calcule toutes les compositions $c(t)$ pour $t \in [0, T]$.

```
def markov_composition(N, nu, T, c0=None):
    res = []
    #composition initiale
    if c0 == None:
        comp = []
        count = 1
        while sum(comp) < N:
            if sum(comp) + count == N:
                comp.append(count)
            elif random() < 0.5:
                comp.append(count)
                count = 1
            else:
                count +=1
        else:
            comp = copy(c0)
    res.append(copy(comp))
    #transitions markoviennes
    for t in range(0,T):
        i = choose_part(comp)
        if random() < nu:
            comp[i] -= 1
            comp.append(1)
        else:
            j = choose_part(comp)
            comp[i] -=1
            comp[j] +=1
        res.append(copy(comp))
    return res
```

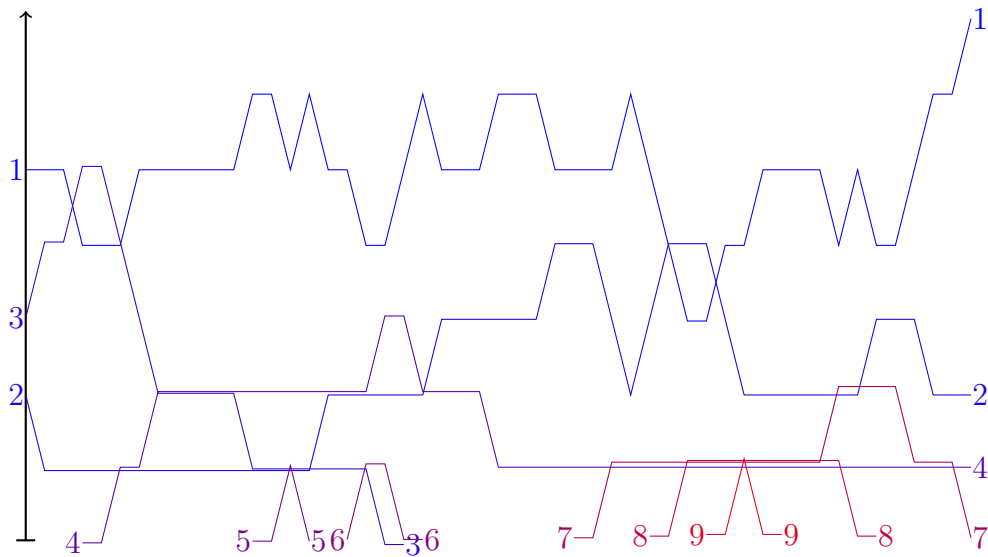
Par exemple, avec $N = 10$, $\nu = 0.1$, $T = 20$ et $c(0) = (5, 2, 3)$, on obtient comme résultat :

```

[[5, 2, 3],
 [4, 3, 3],
 [3, 3, 3, 1],
 [3, 4, 2, 1],
 [2, 4, 2, 1, 1],
 [2, 3, 2, 2, 1],
 [2, 3, 2, 2, 1],
 [1, 4, 2, 2, 1],
 [1, 4, 2, 2, 1],
 [1, 5, 2, 1, 1],
 [1, 4, 2, 2, 1],
 [1, 3, 2, 3, 1],
 [0, 3, 2, 4, 1],
 [0, 3, 2, 3, 2],
 [0, 3, 2, 3, 2],
 [0, 2, 2, 3, 3],
 [0, 2, 1, 3, 4],
 [0, 2, 1, 2, 5],
 [0, 2, 1, 2, 4, 1],
 [0, 3, 1, 2, 4, 0],
 [0, 2, 1, 3, 4, 0]]

```

On a représenté ci-dessous une évolution de la répartition entre espèces avec $N = 10$, $\nu = 0.1$, $c(0) = (5, 2, 3)$ et pour $t \in [0, 50]$ — le dessin a été produit par un programme Python qui a écrit le code LaTeX/TikZ à partir du programme précédent.



On note l'apparition de nouvelles espèces 4, 5, 6, ..., et aussi la disparition de certaines espèces (y compris parmi celles qui sont apparues après le temps $t = 0$).

II.1 Si $N_1 = k$ à un temps t , alors la valeur de N_1 au temps $t + 1$ est :

- soit $k + 1$, ce qui arrive si l'indice 1 n'est pas choisi à l'étape 1, s'il n'y a pas de nouvelle espèce qui apparaît, et si l'indice 1 est choisi dans l'étape 3 de l'algorithme. Ceci arrive avec probabilité

$$\frac{N - k}{N} \frac{k}{N} (1 - \nu)$$

conditionnellement à $N_1(t) = k$.

- soit k , ce qui arrive si l'indice 1 n'est pas choisi à l'étape 1 et si une nouvelle espèce apparaît, ou si l'indice 1 est choisi à l'étape 1 et à l'étape 3, ou si l'indice 1 n'est choisi ni à l'étape 1 ni à l'étape 3. Conditionnellement à $N_1(t) = k$, ceci arrive avec probabilité

$$\frac{N-k}{N} \nu + \left(\frac{k}{N}\right)^2 (1-\nu) + \left(\frac{N-k}{N}\right)^2 (1-\nu).$$

- soit $k-1$, ce qui arrive si l'indice 1 est choisi à l'étape 1 et si une nouvelle espèce apparaît, ou si l'indice 1 est choisi à l'étape 1 et n'est pas choisi à l'étape 3. Conditionnellement à $N_1(t) = k$, ceci se produit avec probabilité

$$\frac{k}{N} \nu + \frac{N-k}{N} \frac{k}{N} (1-\nu).$$

On en déduit la valeur de $\mathbb{E}[N_1(t+1)|N_1(t)]$:

$$\begin{aligned} & \mathbb{E}[N_1(t+1)|N_1(t)] \\ &= N_1(t) + \left(\frac{N-N_1(t)}{N} \frac{N_1(t)}{N} (1-\nu)\right) - \left(\frac{N_1(t)}{N} \nu + \frac{N-N_1(t)}{N} \frac{N_1(t)}{N} (1-\nu)\right) \\ &= N_1(t) \left(1 - \frac{\nu}{N}\right) \leq N_1(t). \end{aligned}$$

Ainsi, $(N_1(t))_{t \geq 0}$ est une surmartingale positive, et d'autre part, on a $\mathbb{E}[N_1(t+1)] = \mathbb{E}[\mathbb{E}[N_1(t+1)|N_1(t)]] = \mathbb{E}[N_1(t)] \left(1 - \frac{\nu}{N}\right)$, donc par récurrence sur t ,

$$\mathbb{E}[N_1(t)] = N_1(0) \left(1 - \frac{\nu}{N}\right)^t.$$

On sait qu'une surmartingale positive $(N_1(t))_{t \in \mathbb{N}}$ converge presque sûrement vers une variable aléatoire positive $N_1(\infty)$, et comme les variables $N_1(t)$ sont bornées dans $[0, N]$, on a aussi convergence dans tous les espaces L^p . En passant à la limite dans la formule précédente, on obtient $\mathbb{E}[N_1(\infty)] = 0$ si $\nu > 0$, et donc $N_1(\infty) = 0$ presque sûrement. Ainsi,

$$N_1(t) \xrightarrow{\text{p.s.}} 0.$$

II.2 On a $\mathbb{E}[T] = \sum_{k=1}^{\infty} \mathbb{P}[T \geq k] = \sum_{k=0}^{\infty} \mathbb{P}[T > k]$, et par ailleurs, si $T > k$, alors $N_1(k) \geq 1$, donc

$$\mathbb{E}[N_1(k)] = \mathbb{E}[N_1(k) 1_{N_1(k) \geq 1}] \geq \mathbb{P}[N_1(k) \geq 1] = \mathbb{P}[T > k].$$

On en déduit avec la formule de la question précédente :

$$\mathbb{E}[T] \leq \sum_{k=0}^{\infty} \mathbb{E}[N_1(k)] = N_1(0) \sum_{k=0}^{\infty} \left(1 - \frac{\nu}{N}\right)^k = \frac{N_1(0) N}{\nu}.$$

L'algorithme suivant prend en argument une composition initiale $c0$ et un paramètre nu , et simule la variable aléatoire T :

def extinction_time($c0, \text{nu}$) :

$\text{comp} = \text{copy}(c0)$

$\text{res} = 0$

while $\text{comp}[0] > 0$:

$i = \text{choose_part}(\text{comp})$

if $\text{random}() < \text{nu}$:

$\text{comp}[i] -= 1$

$\text{comp.append}(1)$

```

else :
    j = choose_part(comp)
    comp[i] -=1
    comp[j] +=1
    res += 1
return res

```

Une liste ordonnée de 100 valeurs de temps d'extinction avec $c(0) = (5, 2, 3)$ et $\nu = 0.1$ est :

```

[10, 14, 15, 16, 16, 18, 18, 18, 19, 20, 20, 20, 21, 21, 22, 23, 23,
 24, 26, 27, 27, 30, 30, 31, 31, 35, 36, 37, 39, 42, 42, 43, 43, 45,
 46, 46, 46, 47, 49, 49, 50, 52, 52, 54, 55, 56, 58, 58, 61, 64, 67,
 69, 71, 74, 77, 79, 82, 83, 88, 93, 98, 101, 105, 108, 108, 114, 119,
 119, 121, 126, 134, 146, 148, 153, 157, 158, 163, 168, 175, 189, 190,
 194, 195, 199, 205, 216, 219, 219, 229, 238, 246, 254, 297, 306, 320,
 441, 472, 490, 498, 543]

```

On obtient ici une moyenne empirique $\hat{T} = 112.09$, qui est bien inférieure à $\frac{N_1(0)N}{\nu} = 500$.

III.1 Il n'est pas tout à fait évident que $(p(t))_{t \in \mathbb{N}}$ soit une chaîne de Markov; commençons par expliquer ceci. On sait que $(c(t))_{t \in \mathbb{N}}$ est une chaîne de Markov, et l'algorithme écrit à la question I.4 permet de représenter les transitions de cette chaîne de la façon suivante :

$$c(t+1) = f(c(t), B_t, U_t, V_t),$$

où B_t est une variable aléatoire de Bernoulli de paramètre ν , et U_t et V_t sont deux variables uniformes sur $[1, N]$ qui permettent de choisir une part non nulle N_i de $c(t)$, et éventuellement une part non nulle N_j si l'on utilise l'étape 3 de l'algorithme de transition. Dans cette écriture, les variables $B_{t \geq 0}$, $U_{t \geq 0}$ et $V_{t \geq 0}$ sont toutes indépendantes. Soit $\pi : \mathfrak{C}_N \rightarrow \mathfrak{P}_N$ l'application qui fait correspondre à une composition c la partition p formée par les parts non nulles et réordonnées de c ; par définition, $p(t) = \pi(c(t))$. Il n'est alors pas difficile de voir que pour calculer le processus $(p(t))_{t \in \mathbb{N}}$, on peut utiliser la même fonction de transition f :

$$p(t+1) = \pi \circ f(p(t), B_t, U'_t, V'_t),$$

avec des variables $U'_{t \geq 0}$ et $V'_{t \geq 0}$ qui sont encore indépendantes et uniformes sur $[1, N]$. Plus précisément, U'_t et V'_t sont reliées à U_t et V_t par une certaine permutation de $[1, N]$ qui dépend de $c(t)$ (et du réordonnement qui permet de passer de $c(t)$ à $p(t)$), mais comme il s'agit de variables uniformes, on garde l'indépendance de ces variables aléatoires et leur uniformité. La représentation $p(t+1) = g(p(t), \xi_t)$ avec des vecteurs $(\xi_t)_{t \geq 0}$ indépendants et de même loi garantit que $(p(t))_{t \geq 0}$ soit une chaîne de Markov.

Expliquons maintenant l'irréducibilité. On peut passer en un nombre fini d'étapes de n'importe quelle partition $p = (N_1 \geq N_2 \geq \dots \geq N_\ell)$ à la partition $(1, 1, \dots, 1)$ en choisissant à chaque transition de diminuer d'une unité une part de p plus grande que 2, et en rajoutant une nouvelle part égale à 1 (apparition d'une nouvelle espèce). Réciproquement, partant de la partition $(1, 1, \dots, 1)$, pour obtenir une partition arbitraire $p = (N_1 \geq N_2 \geq \dots \geq N_\ell)$, on peut à chaque étape supprimer une part N_i de taille 1 et augmenter une part N_j d'une unité, jusqu'à obtenir la

partition souhaitée. Ainsi, $(p(t))_{t \geq 0}$ est irréductible sur l'espace d'états \mathfrak{P}_N . Elle est en particulier récurrente positive, puisque l'espace d'états est fini. Comme à chaque étape il y a une probabilité non nulle de ne pas changer la partition (ne pas rajouter d'espèces et choisir $i = j$), la période de chaque état est égale à 1, c'est-à-dire qu'on a apériodicité.

Par le théorème ergodique pour les chaînes de Markov récurrentes positives et apériodiques, les lois marginales de $(p(t))_{t \geq 0}$ admettent une limite μ . Ainsi,

$$\lim_{t \rightarrow \infty} \mathbb{P}[p(t) = p] = \mu(p)$$

pour une certaine mesure de probabilité μ sur \mathfrak{P}_N .

III.2 Dans ce qui suit, on ordonne les partitions de \mathfrak{P}_4 comme suit :

$$\mathfrak{P}_4 = \{(4), (3, 1), (2, 2), (2, 1, 1), (1, 1, 1, 1)\}.$$

La matrice de transition de la chaîne de Markov $(p(t))_{t \in \mathbb{N}}$ est :

$$P = \begin{pmatrix} 1 - \nu & \nu & 0 & 0 & 0 \\ \frac{3(1-\nu)}{16} & \frac{5-3\nu}{8} & \frac{3(1-\nu)}{16} & \frac{3\nu}{4} & 0 \\ 0 & \frac{1-\nu}{2} & \frac{1-\nu}{2} & \nu & 0 \\ 0 & \frac{1-\nu}{4} & \frac{1-\nu}{8} & \frac{5-\nu}{8} & \frac{\nu}{2} \\ 0 & 0 & 0 & \frac{3(1-\nu)}{4} & \frac{1+3\nu}{4} \end{pmatrix}$$

Par ailleurs, la mesure $\mu_{4,\theta}$ s'écrit

$$\mu_{4,\theta} = \frac{1}{Z_\theta} \left(6, \frac{32\nu}{1-\nu}, \frac{12\nu}{1-\nu}, \frac{96\nu^2}{(1-\nu)^2}, \frac{64\nu^3}{(1-\nu)^3} \right)$$

avec $Z_\theta = (\theta + 1)(\theta + 2)(\theta + 3)$. On vérifie aisément que $\mu_{4,\theta} P = \mu_{4,\theta}$.

Le programme suivant calcule $p(T)$, sachant $p(0) = (1, 1, \dots, 1)$:

def markov_partition(N, nu, T):

```

p = [1]*N
for t in range(0, T):
    i = choose_part(p)
    if random() < nu:
        p[i] -= 1
        p.append(1)
    else:
        j = choose_part(p)
        p[i] -= 1
        p[j] += 1
    p.sort(reverse=True)
    if p[len(p)-1] == 0:
        p = p[0:len(p)-1]
return p

```

Si $N = 4$ et $\nu = 0.5$, $\theta = 4$ et la mesure d'Ewens s'écrit

$$\mu_{4,4} = \left(\frac{1}{35}, \frac{16}{105}, \frac{2}{35}, \frac{16}{35}, \frac{32}{105} \right) = (0.0286, 0.1524, 0.0571, 0.4571, 0.3048).$$

Avec 10000 essais de la fonction **markov_partition** au temps $T = 100$, on obtient la mesure empirique suivante :

$$\hat{\mu}_{4,4} = (0.0305, 0.1465, 0.0628, 0.4537, 0.3065),$$

ce qui est raisonnablement proche de la valeur théorique.

IV.1 L'application s_p est une surjection sur l'ensemble des permutations de type cyclique p . De plus, chaque permutation est atteinte $\prod_{i=1}^N (m_i)! i^{m_i}$ fois, puisque dans une écriture

$$(x_1, x_2, \dots, x_{p_1})(x_{p_1+1}, \dots, x_{p_1+p_2}) \cdots (x_{N-p_l+1}, \dots, x_N),$$

d'une permutation τ de type p comme produit de cycles disjoints :

- on peut permuter les cycles de même taille sans changer la permutation, ce qui représente $\prod_{i=1}^N (m_i)!$ permutations possibles des cycles ;
- on peut dans chaque cycle (a_1, a_2, \dots, a_i) de longueur i effectuer une permutation cyclique des entrées a_j sans changer le cycle, ce qui représente $\prod_{i=1}^N i^{m_i}$ possibilités.

On conclut que l'ensemble C_p des permutations de type cyclique p vérifie :

$$N! = \text{card } \mathfrak{S}_N = \left(\prod_{i=1}^N (m_i)! i^{m_i} \right) \text{card } C_p,$$

d'où la formule pour $\text{card } C_p$.

IV.2 On veut montrer que toute permutation $\sigma \in \mathfrak{S}_N$ s'écrit de manière unique sous la forme $\sigma = (1, n_1) \circ (2, n_2) \circ \cdots \circ (N, n_N)$ avec $n_i \leq i$ pour tout i . Pour des raisons de cardinalité, il suffit de montrer l'existence d'une telle écriture, ce que l'on peut faire par récurrence sur N . Si le résultat est vrai au rang $N - 1$, étant donnée une permutation $\sigma \in \mathfrak{S}_N$, la permutation $\sigma' = \sigma \circ (N, \sigma^{-1}(N))$ envoie N sur N , donc elle appartient à \mathfrak{S}_{N-1} . Il existe donc des entiers n_1, \dots, n_{N-1} tels que $\sigma' = (1, n_1) \circ (2, n_2) \circ \cdots \circ (N-1, n_{N-1})$, et en posant $n_N = \sigma^{-1}(N)$, on conclut que $\sigma = (1, n_1) \circ (2, n_2) \circ \cdots \circ (N, n_N)$.

Montrons maintenant que $\ell(\sigma) = \text{card } \{i \in [1, N] \mid n_i = i\}$. On procède de nouveau par récurrence sur N , le cas $N = 1$ étant trivial. Si le résultat est vrai au rang $N - 1$, considérons $\sigma = (1, n_1) \circ \cdots \circ (N, n_N)$ dans \mathfrak{S}_N . On a par hypothèse de récurrence

$$\begin{aligned} \sigma' &= (1, n_1) \circ (2, n_2) \circ \cdots \circ (N-1, n_{N-1}) \\ &= (a_{1,1}, \dots, a_{1,p_1}) \circ (a_{2,1}, \dots, a_{2,p_2}) \circ \cdots \circ (a_{l,1}, \dots, a_{l,p_l}) \end{aligned}$$

où $l = \text{card } \{i \in [1, N-1] \mid n_i = i\}$, et où les cycles du produit de la seconde ligne sont disjoints et recouvrent $[1, N-1]$. Supposons d'abord $n_N = N$. Alors,

$$\sigma = (a_{1,1}, \dots, a_{1,p_1}) \circ (a_{2,1}, \dots, a_{2,p_2}) \circ \cdots \circ (a_{l,1}, \dots, a_{l,p_l}) \circ (N),$$

donc $\ell(\sigma) = l + 1 = \text{card } \{i \in [1, N-1] \mid n_i = i\} + 1 = \text{card } \{i \in [1, N] \mid n_i = i\}$. Inversement, si $n_N < N$, on peut supposer à une permutation des cycles près que n_N appartient au premier cycle $(a_{1,1}, \dots, a_{1,p_1})$ de σ' , et à une permutation cyclique près des entrées de ce premier cycle, on peut même supposer $n_N = a_{1,p_1}$. Alors,

$$\sigma = \sigma' \circ (N, n_N) = (a_{1,1}, \dots, a_{1,p_1}, N) \circ (a_{2,1}, \dots, a_{2,p_2}) \circ \cdots \circ (a_{l,1}, \dots, a_{l,p_l}),$$

donc $\ell(\sigma) = l = \text{card } \{i \in [1, N-1] \mid n_i = i\} = \text{card } \{i \in [1, N] \mid n_i = i\}$. On a donc aussi le résultat vrai au rang N .

Dans l'algorithme décrit pour la construction de σ_N , on a :

$$\mathbb{P}[n_i = i] = \mathbb{P}[B_i = 1] = \frac{\theta}{\theta + i - 1}$$

et

$$\mathbb{P}[n_i = j] = \mathbb{P}[B_i = 0 \text{ et } m_i = j] = \frac{i - 1}{\theta + i - 1} = \frac{1}{\theta + i - 1}$$

si $j < i$. On en déduit que la loi de σ_N s'écrit :

$$\mathbb{P}[\sigma_N = \sigma] = \frac{\theta^{\text{card}\{i \mid i=n_i\}}}{\prod_{i=1}^N \theta + i - 1} = \frac{\theta^{\ell(\sigma)}}{\prod_{i=1}^N \theta + i - 1} = \rho_{N,\theta}(\sigma).$$

Finalement, $\ell(\sigma_N) = \sum_{i=1}^N 1_{n_i=i} = \sum_{i=1}^N B_i$, ce qui achève la preuve du théorème.

IV.3 Le programme suivant simule une partition aléatoire sous la mesure d'Ewens $\mu_{N,\theta}$ (on utilise pour ce programme la classe **Permutation** de Sage ; dans Python, il faudrait reprogrammer quelques méthodes pour les permutations, par exemple le produit et le type cyclique).

```
def ewens(N, theta):
    perm = Permutation(range(1, N+1))
    for i in range(2, N+1):
        if random() > theta / (theta + i - 1):
            j = floor(1 + (i - 1) * random())
            perm = perm * Permutation((i, j))
    return perm.cycle_type()
```

On a représenté ci-dessous les log-proportions correspondant à ces partitions aléatoires, pour $N = 10000$ et $\theta \in \{1, 2, 5, 10\}$. On observe des courbes de forme similaire, et étendues horizontalement en fonction du paramètre θ .

