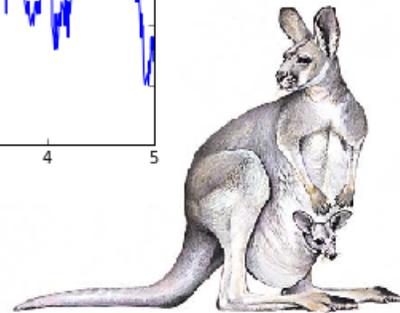
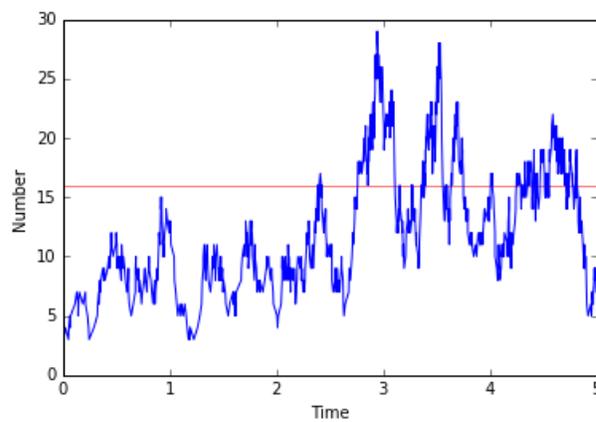




MAP581
APPROFONDISSEMENT EN MATHÉMATIQUES APPLIQUÉES

Distributions quasi-stationnaires de populations



Chenlin GU & Yiyang YU
Promotion X2014

Tuteur : Vincent BANSAYE
Resposable : Thomas WICK

DISTRIBUTIONS QUASI-STATIONNAIRES DE POPULATIONS

CHENLIN GU AND YIYANG YU

RÉSUMÉ. On propose une étude sur les distributions quasi-stationnaires des modèles de populations dans ce rapport. Après avoir introduit les définitions et les notions utiles dans la théorie des distributions quasi-stationnaires, on analyse quelques modèles classiques de populations sous angle de distribution quasi-stationnaire, en temps discret puis en temps continu. On s'intéresse ensuite aux simulations numériques. Ainsi, on établit la relation entre la vitesse de convergence et la loi quasi-stationnaire, et on propose une méthode originale pour calculer numériquement la distribution quasi-stationnaire.

TABLE DES MATIÈRES

1. Introduction	2
2. Définitions, propriétés générales	3
2.1. Notations	3
2.2. Limite de Yaglom, QLD, QSD	3
2.3. Caractérisation spectrale	5
2.4. Convergence exponentielle	5
3. Application dans le processus de Galton-Watson	6
3.1. Définitions, propriétés	6
3.2. Existence et unicité de distribution quasi-stationnaire	8
4. Application dans le processus de naissance et mort	9
4.1. Définitions, condition d'extinction et condition de non-explosion	9
4.2. Condition d'unicité	11
4.3. Résultats numériques	12
5. Simulations numériques	14
5.1. Une méthode naïve : Monte-Carlo sur les chaînes de Markov	14
5.2. Approche par système de particules de type de Fleming-Viot	14
5.3. Calculer la loi quasi-stationnaire	16
Références	18
Annexe : codes Python pour les simulations	19

Date: Mars 2017.

Key words and phrases. Distribution quasi-stationnaire, Limite de Yaglom, Modèle de population, Processus et Galton-Watson, Processus de naissance et mort.

1. INTRODUCTION

Dans le cours de *Modèles aléatoires en Écologie et Évolution*, nous avons vu que dans beaucoup de cas, notre modèle de population modélisé par un processus de Markov s'éteint presque sûrement. Pourtant, le temps nécessaire à l'extinction peut être très long (par exemple dans le cas critique pour un processus de Galton-Watson), et l'échelle de temps n'est pas humaine. Néanmoins, dans certains cas, on peut observer une apparente stationnarité de la population, alors que celle-ci est en voie d'extinction.

L'observation ci-dessus nous incite à donner un sens mathématique à cette stabilité avant l'extinction. Par exemple, on peut mener la recherche par analogie avec la théorie de probabilité stationnaire. Les distributions quasi-stationnaires apparaissent lorsque l'on considère un processus de Markov Z_n atteint presque sûrement un point absorbant et que l'on conditionne ce processus à ne pas être absorbé, c'est-à-dire ne pas atteindre l'état absorbant $\{0\}$ dans les modèles de populations. Les distributions quasi-stationnaires laissent alors invariante la loi conditionnelle de Z_n sachant que la population n'est pas éteinte au temps n . Elles s'interprètent alors comme la distribution de la taille d'une population en temps grand, conditionnée à survivre alors qu'elle devrait s'éteindre. Aussi, on s'intéresse à savoir si, la limite de la loi conditionnelle de Z_n sachant que la population n'est pas éteinte au temps n , existe toujours, et quelle est la relation entre celle-ci et la distribution quasi-stationnaire.

Partant de ces notions théoriques, on souhaite étudier comment elles s'appliquent dans des modèles classiques de population. Et aussi, on souhaite obtenir numériquement les distributions quasi-stationnaires par simulation.

Ce rapport est organisé de la façon suivante : dans section 2, on introduit des définitions et des propriétés générales de la quasi-stationnarité, la caractérisation spectrale et aussi la convergence exponentielle. Dans section 3 et 4, on présente ses applications dans deux modèles classiques de population : le processus de Galton-Watson et le processus de naissance et mort. Dans section 5, on explique comment réaliser efficacement des simulations des distributions quasi-stationnaires.

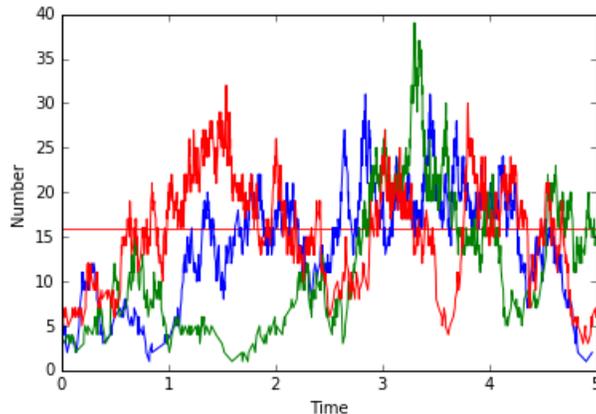


FIGURE 1. Évolution de la population en fonction de temps

2. DÉFINITIONS, PROPRIÉTÉS GÉNÉRALES

Dans cette section, on présente la théorie de base de la quasi-stationnarité. On introduit d'abord quelques notations, qui sont adoptées dans la suite du rapport. Puis, on présente les trois définitions de distributions limites : tous les trois étant de loi quasi-stationnaire, on expliquera les différences subtiles entre elles. La caractérisation spectrale est un outil essentiel dans les études de loi quasi-stationnaires. Enfin on parle de la vitesse de convergence, qui donne une analyse quantitative du comportement à long terme.

2.1. Notations. On se limite, tout au long de ce rapport, dans une modélisation markovienne. On note $\{Z_t\}_{t \geq 0}$ la dynamique d'une population. On suppose que $\{Z_t\}_{t \geq 0}$ prend des valeurs dans l'espace dénombrable E . Par exemple, dans le cas d'un modèle d'une population, $E = \mathbb{N}$; et pour un modèle de multi-type, $E = \mathbb{N}^d$ où d signifie le nombre de types de population.

On note ∂ l'état absorbant dans l'espace, et on suppose $\partial = 0$ dans la suite sauf explication. On note l'espace non-absorbant $E^* = E \setminus 0$. En plus, on note (dans le cas sans immigration)

$$\text{Extinction} = \{\exists t \geq 0, Z_t = 0\} = \{\exists t \geq 0, \forall u \geq t, Z_u = 0\}$$

et le temps d'extinction

$$T_0 = \inf\{t > 0, Z_t = 0\}$$

On définit aussi le demi-groupe P_t associé à ce processus avant l'extinction, ainsi que les intégrations dans E^* . Soit f une fonction mesurable bornée. On note

$$\begin{aligned} P_t f(z) &= \mathbb{E}_z[f(Z_t) \mathbf{1}_{T_0 > t}] \\ \nu(f) &= \int_{E^*} f(z) \nu(dz) \\ \nu P_t(f) &= \nu(P_t f) = \mathbb{E}_\nu[f(Z_t) \mathbf{1}_{T_0 > t}] \end{aligned}$$

2.2. Limite de Yaglom, QLD, QSD. Comme présenté dans l'introduction, on s'intéresse au comportement longtemps conditionné en vie. Dans cette partie, on aborde des différentes définitions.

Définition 2.1 (QLD). Une loi de probabilité α sur E^* est dite **loi quasi-limite**(QLD) s'il existe une loi ν en E^* telle que $\forall A \subset E$ mesurable

$$\lim_{t \rightarrow \infty} \mathbb{P}_\nu[Z_t \in A | T_0 > t] = \alpha(A)$$

Quelques fois, la QLD peut être indépendante de la loi initiale, ce qui suggère la définition de limite de Yaglom comme suivant.

Définition 2.2 (Yaglom). Une loi de probabilité α en E^* est dite **limite de Yaglom** s'il satisfait que $\forall A \subset E^*$ mesurable et $\forall x \in E^*$

$$\lim_{t \rightarrow \infty} \mathbb{P}_x[Z_t \in A | T_0 > t] = \alpha(A)$$

La troisième définition est une probabilité invariante sous la demi-groupe.

Définition 2.3 (QSD). Une loi de probabilité α en E^* est dite **loi quasi-stationnaire** s'il satisfait que $\forall A \subset E^*$ mesurable

$$\mathbb{P}_\alpha[Z_t \in A | T_0 > t] = \alpha(A)$$

Une question naturelle est le rapport entre les trois lois. Dans le cas générale, le théorème suivant nous l'explique.

Theorème 2.4. *La définition de QSD et celle de QLD sont équivalentes. Si la limite de Yaglom existe, elle est QSD. C'est-à-dire*

$$(2.1) \quad \text{Limite de Yaglom} \Rightarrow QLD \Leftrightarrow QSD$$

Démonstration. Les implications Limite de Yaglom \Rightarrow QLD et QSD \Rightarrow QLD sont évidentes : il s'agit d'une application des définition. On se concentre ensuite sur l'implication QLD \Rightarrow QSD.

On utilise l'identité

$$\alpha(f) = \lim_{t \rightarrow \infty} \mathbb{E}_\nu[f(Z_t) | T_0 > t] = \lim_{t \rightarrow \infty} \frac{\mathbb{E}_\nu[f(Z_t), T_0 > t]}{\mathbb{P}_\nu[T_0 > t]}$$

et la propriété de Markov, et avec $f(z) = \mathbb{E}_z[\mathbf{1}_{T_0 > s}]$.

$$\alpha(f) = \mathbb{P}_\alpha[T_0 > s] = \lim_{t \rightarrow \infty} \frac{\mathbb{E}_\nu[\mathbb{E}_{Z_t}[\mathbf{1}_{T_0 > s}], T_0 > t]}{\mathbb{P}_\nu[T_0 > t]} \stackrel{\text{Markov}}{=} \lim_{t \rightarrow \infty} \frac{\mathbb{P}_\nu[T_0 > t + s]}{\mathbb{P}_\nu[T_0 > t]}$$

On utilise la même stratégie pour $g(z) = \mathbb{E}_z[\mathbf{1}_{T_0 > s}, Z_s \in A]$.

$$\begin{aligned} \alpha(g) = \mathbb{P}_\alpha[T_0 > s, Z_s \in A] &= \lim_{t \rightarrow \infty} \frac{\mathbb{E}_\nu[\mathbb{E}_{Z_t}[\mathbf{1}_{T_0 > s, Z_s \in A}], T_0 > t]}{\mathbb{P}_\nu[T_0 > t]} \\ &\stackrel{\text{Markov}}{=} \lim_{t \rightarrow \infty} \frac{\mathbb{P}_\nu[Z_{t+s} \in A, T_0 > t + s]}{\mathbb{P}_\nu[T_0 > t]} \end{aligned}$$

On combine avec le premier résultat et utilise la définition de QLD

$$\begin{aligned} \mathbb{P}_\alpha[T_0 > s, Z_s \in A] &= \lim_{t \rightarrow \infty} \frac{\mathbb{P}_\nu[Z_{t+s} \in A, T_0 > t + s]}{\mathbb{P}_\nu[T_0 > t]} \\ &= \lim_{t \rightarrow \infty} \frac{\mathbb{P}_\nu[Z_{t+s} \in A, T_0 > t + s]}{\mathbb{P}_\nu[T_0 > t + s]} \frac{\mathbb{P}_\nu[T_0 > t + s]}{\mathbb{P}_\nu[T_0 > t]} \\ &= \alpha(A) \mathbb{P}_\alpha[T_0 > s] \end{aligned}$$

Ce qui correspond bien à la définition de QSD. On a montré donc que les deux définitions QSD et QLD sont équivalentes. \square

Ce théorème nous dit que si la limite de Yaglom existe, elle est unique et les trois définition sont cohérentes. Mais généralement, il peut y avoir plusieurs QLDs ou QSDs.

On finit cette partie par une proposition.

Proposition 2.5 (Extinction avec vitesses exponentielle). *Soit α une QSD, alors il existe une constante $\theta(\alpha) > 0$ telle que*

$$\mathbb{P}_\alpha[T_0 > t] = e^{-\theta(\alpha)t}$$

Démonstration.

$$\begin{aligned} \mathbb{P}_\alpha[T_0 > t + s] &= \mathbb{E}_\alpha[\mathbb{E}_\alpha[\mathbf{1}_{T_0 > t+s} | \mathcal{F}_t]] \\ &= \mathbb{E}_\alpha[\mathbb{E}_\alpha[\tilde{T}_0 > s] | T_0 > t] \\ &= \mathbb{P}_\alpha[T_0 > t] \mathbb{P}_\alpha[T_0 > s] \end{aligned}$$

On résout cette équation fonctionnelle et utilise la fait $\mathbb{P}_\alpha[T_0 > 0] = 1$ pour obtenir le résultat souhaité. \square

2.3. Caractérisation spectrale. Dans cette partie, on introduit la caractérisation spectrale de QSD, un outil puissant qui nous permet de trouver QSD directement par résolution de l'équation linéaire.

Theorème 2.6 (Générateur et QSD). *Soit L l'opérateur associé au demi-groupe P_t . On note par ailleurs D le domaine de l'opérateur. Soit α une QSD, et $\theta(\alpha)$ sa puissance d'extinction. Alors, $\forall f \in D$, on a*

$$\alpha(Lf) = -\theta(\alpha)\alpha(f)$$

Démonstration. On applique le résultat de 2.5. $\forall A \subset E$ mesurable, on a

$$\begin{aligned} \alpha(A) = \mathbb{P}_\alpha[Z_t \in A | T_0 > t] &= \frac{\mathbb{P}_\alpha[Z_t \in A, T_0 > t]}{\mathbb{P}_\alpha[T_0 > t]} = e^{\theta(\alpha)t} \alpha P_t(\mathbf{1}_A) \\ &\Rightarrow \alpha P_t(\mathbf{1}_A) = e^{-\theta(\alpha)t} \alpha(A) \end{aligned}$$

Comme la classe des fonctions indicatrices $\{\mathbf{1}_A\}$ est dense dans l'espace des fonctions continues bornés, on peut généraliser ce résultat pour tout $f \in C_b(E)$

$$\alpha P_t(f) = \sum_{x \in E^*} f(x) e^{-\theta(\alpha)t} \alpha(x)$$

En utilisant l'équation de Kolmogorov

$$\alpha L P_t(f) = \frac{d}{dt} \alpha P_t(f) = \sum_{x \in E^*} (-\theta(\alpha)) f(x) e^{-\theta(\alpha)t} \alpha(x)$$

Quand $t \rightarrow 0$, on a alors $\alpha(Lf) = -\theta(\alpha)\alpha(f)$. \square

Autrement dit, α est le vecteur propre à gauche associé à L , dont la valeur propre est $\theta(\alpha)$, qui est la constante d'extinction exponentielle.

2.4. Convergence exponentielle. Une question clé dans la théorie de loi quasi-stationnaire est de savoir la condition de l'unicité de QSD et la vitesse de convergence. [3] a étudié cette question et a obtenu une réponse complète avec des critères faisables qui s'appliquent dans les exemples.

On rappelle la définition de la distance totale, que l'on utilisera pour mesurer la distance entre la loi en temps t et QSD.

Définition 2.7 (Distance totale). Soit ν, μ deux probabilités à valeurs dans E . la distance totale entre elles est définie comme

$$\|\nu - \mu\|_{TV} = \frac{1}{2} \sum_{x \in E} |\nu(x) - \mu(x)|$$

[3] propose deux conditions suivantes comme critères.

Hypothèse 2.8. *Il existe une probabilité ν_0 à valeur dans E telle que*

(H1) *il existe $t_0, c_1 > 0$ tels que $\forall x \in E$*

$$\mathbb{P}_x[X_{t_0} \in \cdot | T_0 > t_0] \geq c_1 \nu_0(\cdot)$$

(H2) *il existe c_2 tel que $\forall x \in E$ et $t \geq 0$*

$$\mathbb{P}_{\nu_0}[T_0 > t] \geq c_2 \mathbb{P}_x[T_0 > t]$$

Theorème 2.9. *2.8 implique qu'il existe une loi de probabilité α sur E telle que pour toute probabilité initiale ν ,*

$$\|\mathbb{P}_\nu[X_t \in \cdot | T_0 > t] - \alpha(\cdot)\|_{TV} \leq 2(1 - c_1 c_2)^{\lfloor t/t_0 \rfloor}$$

Réciproquement, on a aussi que, s'il existe une convergence exponentielle uniformément pour toute loi de probabilité initiale vers une unique QSD, alors l'hypothèse 2.8 est établie.

On ne reproduit pas ici la démonstration donnée dans l'article. On cite une autre proposition intéressante exposée dans cet article, qui consiste à dire que la vitesse d'extinction est asymptotiquement la même que la QSD. En plus, cette proposition nous inspire une simulation de QSD et le détail sera présenté dans la dernière section.

Proposition 2.10. *Sous l'hypothèse 2.8, il existe une fonction positive η en E telle que pour toute ν probabilité initiale,*

$$\nu(\eta) = \lim_{t \rightarrow \infty} e^{\theta(\alpha)t} \mathbb{P}_\nu[T_0 > t]$$

Démonstration. Soit $x \in E$ et $t > 0$, on note

$$\eta_t(x) = \frac{\mathbb{P}_x[T_0 > t]}{\mathbb{P}_\alpha[T_0 > t]} = e^{\theta(\alpha)t} \mathbb{P}_x[T_0 > t]$$

En utilisant la propriété de Markov

$$\begin{aligned} \eta_{t+s}(x) &= e^{\theta(\alpha)(t+s)} \mathbb{E}_x[\mathbf{1}_{T_0 > t} \mathbb{P}_{Z_t}[T_0 > s]] \\ &= \eta_t(x) \mathbb{E}_x[\eta_s(Z_t) | T_0 > t] \end{aligned}$$

L'hypothèse (H1) entraîne que

$$\eta_t(x) = e^{\theta(\alpha)t} \mathbb{P}_x[T_0 > t] \leq \frac{1}{c_2} e^{\theta(\alpha)t} \mathbb{P}_{\nu_0}[T_0 > t]$$

et on sait $\mathbb{P}_{\nu_0}[T_0 > t]$ converge vers $\mathbb{P}_\alpha[T_0 > t]$, donc $\eta_t(x)$ est uniformément bornée. Puis, la convergence exponentielle implique aussi

$$|\mathbb{E}_x[\eta_s(Z_t) | T_0 > t] - \alpha(\eta_s)| \leq \frac{C}{c_2} e^{-\gamma t}$$

dont $\alpha(\eta_s) = 1$. Donc, on obtient

$$\sup_{x \in E} |\eta_{t+s}(x) - \eta_t(x)| \leq \frac{C}{(c_2)^2} e^{-\gamma t}$$

qui implique que η_t est une suite de Cauchy. Alors sa limite η existe. En plus, car il a une vitesse exponentielle de convergence, le théorème de convergence dominée s'applique et $\alpha(\eta) = 1$ et cette proposition pour n'importe quelle loi initiale. \square

3. APPLICATION DANS LE PROCESSUS DE GALTON-WATSON

3.1. Définitions, propriétés. La chaîne de Galton-Watson (que nous abrègerons BGW, pour Bienaymé-Galton-Watson), qui décrit la dynamique d'une population en temps discret, est une chaîne sans immigration et densité-indépendante. C'est le prototype le plus simple de processus de branchement, défini pour des temps et un espace d'états discrets.

Nous étudions ici quelques propriétés révélant de la quasi-stationnarité du processus BGW.

Définition 3.1. (X_n) est une chaîne de Galton-Watson si $X_0 \in \mathbb{N}$ et

$$X_{n+1} = \sum_{k=1}^{X_n} Y_{n,k}$$

pour tout $n \geq 0$, où les $Y_{n,k}$ sont les variables aléatoires indépendantes et identiquement distribuées de même loi μ , qui prend des valeurs dans \mathbb{N} . Cette loi est caractérisée par sa fonction génératrice

$$g(s) = \mathbb{E}(s^Y) = \sum_k q_k s^k$$

pour tout $s \in [0, 1]$

Par construction, X_n est la taille de la n -ième génération de la population, dont chaque individu a un nombre aléatoire d'enfants, suivant la loi μ et indépendant du reste de la population.

Remarque. De cette définition, on voit immédiatement que 0 est un état absorbant.

On note par la suite $m = \mathbb{E}[Y_{0,1}]$ le nombre moyen d'enfants donnés par un individu dans notre processus BGW. Puis on note m_2 son moment au deuxième ordre $m_2 = \sum_{k=0}^{\infty} k^2 q_k$.

Proposition 3.2 (Propriété de branchement). *Soit (X_n) un processus BGW. La loi de (X_n) sous \mathbb{P}_i est la même que la loi de $(Z_n^1 + \dots + Z_n^i)$ où les (Z_n^j) sont des chaînes de Markov indépendantes entre elles et qui ont toutes même loi que la chaîne $(X_n)_n$ sous \mathbb{P}_1*

Démonstration. Si $X_0 = i$, on peut écrire $X_n = \sum_{j=1}^i Z_n^j$, où Z_n^j désigne le nombre d'individus vivants à l'instant n , descendant du j -ième individu initial. \square

Proposition 3.3. *Soit G_n la fonction génératrice de X_n . Alors $\forall n \geq 0$,*

$$G_{n+1}(s) = G_n(g(s))$$

Démonstration.

$$\begin{aligned} G_{n+1}(s) &= \sum_k \mathbb{E}[s^{\sum_{i=1}^k Y_{n,i}} \mathbf{1}_{\{X_n=k\}}] \\ &= \sum_k (g(s))^k \mathbb{P}(X_n = k) \\ &= G_n(g(s)) \end{aligned}$$

\square

Proposition 3.4.

$$\mathbb{P}(\text{Extinction}) = \inf\{s \in [0, 1], g(s) = s\}$$

En particulier, $\mathbb{P}(\text{Extinction}) = 1$ si et seulement si $m \leq 1$

3.2. Existence et unicité de distribution quasi-stationnaire.

Theorème 3.5 (Yaglom, 1947). *Soit (X_n) une chaîne de BGW sous-critique ($m < 1$). Alors il existe une probabilité $\mu = (\mu_j)_{j \geq 1}$ sur \mathbb{N}^* , limite de Yaglom de la chaîne de BGW. En particulier, elle ne dépend pas de la condition initiale, et c'est une loi quasi-stationnaire. De plus, la fonction génératrice \hat{g} de μ est l'unique solution de l'équation suivante : $\forall s \in [0, 1]$,*

$$1 - \hat{g}(g(s)) = m(1 - \hat{g}(s))$$

Démonstration. On note g_n la n -ième itérée de g , c'est-à-dire $g \circ g \circ \dots \circ g$ n fois. C'est la fonction génératrice de $(X_n)_n$. On introduit la fonction génératrice conditionnelle \hat{g}_n définie de la façon suivante : $\forall s \in [0, 1]$

$$\hat{g}_n(s) = \mathbb{E}_1(s^{X_n} | X_n > 0) = \frac{\mathbb{E}_1(s^{X_n} \mathbf{1}_{X_n > 0})}{\mathbb{P}_1(X_n > 0)}$$

Ainsi,

$$\hat{g}_n(s) = \frac{\mathbb{E}_1(s^{X_n}) - \mathbb{P}(X_n = 0)}{1 - \mathbb{P}_1(X_n = 0)} = \frac{g_n(s) - g_n(0)}{1 - g_n(0)} = 1 - \frac{1 - g_n(s)}{1 - g_n(0)}$$

Remarquons que pour s fixé, la suite $n \mapsto \frac{1 - g_n(s)}{1 - g_n(0)}$ est croissante. En effet, comme les fonctions génératrices sont convexes, leur taux d'accroissement est croissant, donc la fonction $u \mapsto \frac{1 - g(u)}{1 - u}$ est croissante. Donc

$$\frac{1 - g_{n+1}(s)}{1 - g_n(s)} \geq \frac{1 - g_{n+1}(0)}{1 - g_n(0)} \quad \text{puis} \quad \frac{1 - g_{n+1}(s)}{1 - g_{n+1}(0)} \geq \frac{1 - g_n(s)}{1 - g_n(0)}$$

Ainsi la suite $(\frac{1 - g_n(s)}{1 - g_n(0)})_n$ est croissante et majorée par 1, la suite $(\hat{g}_n(s))_n$ est donc décroissante et minorée par 0, donc converge vers $\hat{g}(s) \in [0, 1]$.

Ensuite, on montre que \hat{g} est solution d'une équation fonctionnelle.

$$1 - \hat{g}(s) = \lim_n \frac{1 - g_{n+1}(s)}{1 - g_{n+1}(0)}$$

Or, on sait que

$$\lim_{n \rightarrow \infty} \frac{1 - g_n(0)}{1 - g_{n+1}(0)} = \lim_{u \rightarrow 1} \frac{1 - (u)}{1 - g(u)} = \frac{1}{m}$$

Et de plus, par définition de \hat{g}

$$\lim_n \frac{1 - g_n(g(s))}{1 - g_n(0)} = 1 - \hat{g}(g(s))$$

On obtient que \hat{g} est la solution de l'équation fonctionnelle suivante : $\forall s \in [0, 1]$

$$1 - \hat{g}(g(s)) = m(1 - \hat{g}(s))$$

On montre ensuite que \hat{g} ne dépend pas de la distribution initiale. En effet, s'il y a i individus initialement, comme précédemment, on a

$$\hat{g}_{n,i}(s) = \mathbb{E}_i(s^{X_n} | X_n > 0) = \frac{(\mathbb{E}_1(s^{X_n}))^i - (\mathbb{P}(X_n = 0))^i}{1 - (\mathbb{P}_1(X_n = 0))^i}$$

Comme le processus $(X_n)_n$ est sous-critique, $\mathbb{E}_1(s^{X_n})$ et $\mathbb{P}(X_n = 0)$ tendent vers 1 quand n tend vers l'infini.

$$\begin{aligned} \mathbb{E}_1(s^{X_n}) &= 1 + i(\mathbb{E}_1(s^{X_n}) - 1) + o(\mathbb{E}_1(s^{X_n}) - 1) \\ \mathbb{P}_i(X_n = 0) &= 1 - i\mathbb{P}_1(X_n = 0) + o(\mathbb{P}_1(X_n = 0)) \end{aligned}$$

Ainsi

$$\mathbb{E}_i(s^{X_n} | X_n > 0) \sim_{n \rightarrow \infty} \frac{\mathbb{E}_1(s^{X_n}) - 1 + \mathbb{P}(X_n > 0)}{1 - 1 + \mathbb{P}_1(X_n > 0)} = \frac{\mathbb{E}_1(s^{X_n}) - \mathbb{P}(X_n = 0)}{1 - \mathbb{P}_1(X_n = 0)}$$

d'où $\hat{g}_{n,i}$ converge vers $\hat{g}(s)$ quand n tend vers l'infini.

D'ailleurs, comme \hat{g} est continue en 1, \hat{g} est fonction génératrice d'une mesure μ qui ne charge pas 0. μ ainsi définie est la limite de Yaglom. On montre maintenant qu'il s'agit aussi d'une distribution quasi-stationnaire.

$$\mathbb{E}_\mu(s^{X_n} | X_n > 0) = \frac{\mathbb{E}_\mu(s^{X_n}) - \mathbb{P}_\mu(X_n = 0)}{1 - \mathbb{P}_\mu(X_n = 0)}$$

Or

$$\mathbb{P}_\mu(X_n = 0) = \hat{g}(g_n(0)) \quad \text{et} \quad \mathbb{E}_\mu(s^{X_n}) = \hat{g}(g_n(s))$$

Et grâce à la formule précédente, et par récurrence

$$1 - \hat{g}(g_n(s)) = m^n(1 - \hat{g}(s))$$

Finalement, on obtient

$$\mathbb{E}_\mu(s^{X_n} | X_n > 0) = \frac{m^n \hat{g}(s)}{m^n} = \hat{g}(s)$$

La loi conditionnelle est de même fonction génératrice que μ . On a ainsi prouvé que μ est une distribution quasi-stationnaire pour la chaîne X_n . \square

Remarque. Il n'y a pas de distribution quasi-stationnaire dans le cas sur-critique ($m > 1$) ou critique ($m = 1$).

4. APPLICATION DANS LE PROCESSUS DE NAISSANCE ET MORT

Dans cette section, on applique la théorie de la distribution quasi-stationnaire dans le modèle de naissance et mort, qui est une modélisation de population par processus markovien de saut en temps continu.

4.1. Définitions, condition d'extinction et condition de non-explosion.

Définition 4.1 (Processus de naissance et mort). Un processus de naissance et mort est un processus markovien de saut dont les amplitudes des sauts sont égales à 1. Les taux de transition sont donnés par

$$\begin{aligned} i &\rightarrow i + 1 && \text{au taux } \lambda_i \\ i &\rightarrow i - 1 && \text{au taux } \mu_i \end{aligned}$$

avec $(\lambda_i)_i, (\mu_i)_i$ deux suites réelles strictement positives pour tout $i \in \mathbb{N}^*$, et $\lambda_0 = \mu_0 = 0$.

On cite quelques exemples classiques importants.

- (1) **Linéaire** Le processus de naissance et mort linéaire correspond au cas $\lambda_i = i\lambda, \mu_i = i\mu$. C'est aussi un processus de branchement binaire.
- (2) **Avec immigration** Le processus de naissance et mort linéaire avec immigration correspond au cas $\lambda_i = i\lambda + \rho, \mu_i = i\mu$, où ρ s'interprète comme le taux d'immigration.

- (3) **Logistique** Le processus de naissance et mort logistique correspond au cas $\lambda_i = i\lambda, \mu_i = i\mu + ci(i-1)$, dont le terme quadratique dans le taux de mort vient du fait qu'il existe des compétitions, par exemple à cause de quantité limitée de ressources naturelles. C'est aussi de définir le modèle avec compétition faible qui s'appelle **modèle logistique de puissance fractale** dont $\mu_i = i\mu + ci(i-1)^\gamma, \gamma > 0$.

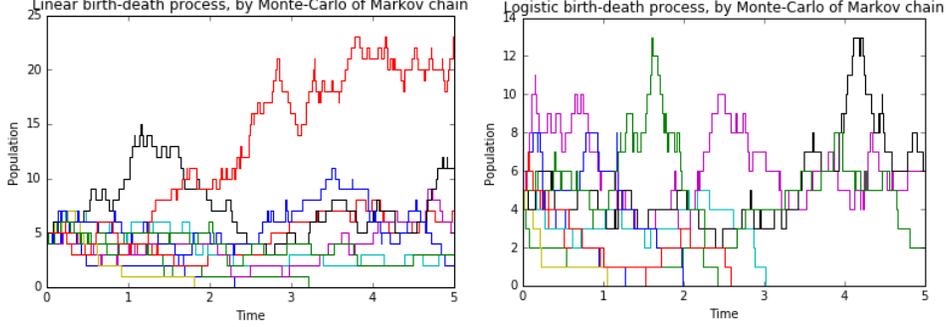


FIGURE 2. Exemples de processus de naissance et mort linéaire à gauche, et logistique à droite, avec $c = 0.1, \mu = 0.99, \lambda = 1.01$, et $Z_0 = 5$, par méthode de Monte-Carle de chaîne de Markov

Comme dans le processus de naissance et mort avec immigration, l'état 0 n'est pas absorbant, les études sur les distributions quasi-stationnaires n'ont pas vocation à s'appliquer dans ce cas. On s'intéresse donc aux applications dans le modèle linéaire et le modèle logistique.

Pour étudier le cas avec extinction presque-sûre, on énonce deux critères sur la propriété de non-explosion et d'extinction.

Theorème 4.2 (Critère de non-explosion). *Supposons que $\lambda_i > 0$ pour tout $i > 1$. Alors le processus de naissance et mort a un temps de vie infini presque-sûrement si et seulement si*

$$R := \sum_{i \geq 1} \left(\frac{1}{\lambda_i} + \frac{\mu_i}{\lambda_i \lambda_{i-1}} + \dots + \frac{\mu_i \cdots \mu_2}{\lambda_i \cdots \lambda_2 \lambda_1} \right) \text{ est infini}$$

Theorème 4.3 (Critère d'extinction). *Nous posons*

$$U_I = \sum_{k=1}^{I-1} \frac{\mu_1 \cdots \mu_k}{\lambda_1 \cdots \lambda_k}$$

Si U_I tend vers l'infini quand $I \rightarrow \infty$, alors toutes les probabilités d'extinction sont égales à 1.

Si U_I converge vers une limite finie U_∞ , alors pour $i \geq 1$,

$$u_i = (1 + U_\infty)^{-1} \sum_{k=i}^{\infty} \frac{\mu_1 \cdots \mu_k}{\lambda_1 \cdots \lambda_k}$$

Il y a une probabilité strictement positive de survie du processus.

C'est facile de vérifier que le modèle logistique satisfait la condition d'extinction presque sûrement et la condition non-explosion grâce au terme quadratique.

4.2. Condition d'unicité. La condition d'unicité est généralement un peu abstrait, mais dans le processus de naissance et mort, on propose une condition équivalente qui est plus facile à vérifier.

Theorème 4.4 (Critère d'unicité de QSD). *Une condition équivalente de convergence exponentielle est*

$$S = \sum_{k \geq 1} \frac{1}{\mu_k \pi_k} \sum_{l \geq k} \pi_k < \infty$$

où $\pi_k = (\prod_{i=1}^{k-1} \lambda_i) / (\prod_{i=1}^k \mu_i)$.

En plus, cette condition assure l'unicité de QSD et elle est la limite de Yaglom.

Démonstration. Étape 1 : $H_1 \Rightarrow S < \infty$ Il suffit de montrer que

$$\sup_{n \geq z} \mathbb{E}_n[T_z] = \sum_{k \geq z+1} \frac{1}{\mu_k \pi_k} \sum_{l \geq k} \pi_k < \infty$$

et puis on ajoute nombre fini de termes. Cette propriété s'appelle la propriété de descendre de l'infini.

On choisit z tel que

$$\nu_0([1, 2 \cdots z]) > 0$$

, alors

$$\begin{aligned} \mathbb{P}_n[X_{t_0} \leq z] &= \mathbb{P}_n[X_{t_0} \leq z | T_0 > t_0] \mathbb{P}_n[T_0 \geq t_0] \\ &\geq c_1 \nu_0([1, z]) \mathbb{P}_n[T_0 \geq t_0] \end{aligned}$$

Car chaque pas, il y a au plus μ_1 de probabilité d'être mort, donc $\mathbb{P}_n[T_0 \geq t_0] \geq e^{-\bar{q}t_0}$ et

$$\sup_{n \geq z} \mathbb{P}_n[X_{t_0} > z] = 1 - \inf_{n \geq z} \mathbb{P}_n[X_{t_0} \leq z] \leq (1 - C e^{-\bar{q}t_0})$$

On utilise la décomposition de l'espérance

$$\begin{aligned} \mathbb{E}_n[T_z] &\leq \sum_{k \geq 0} \mathbb{P}_n[T_z > k] \leq \sum_{k \geq 0} \mathbb{P}_n[X_k > z] \\ &\leq \sum_{i < t_0} \sum_{m=0}^{\infty} \mathbb{P}_n[X_{mt_0+i} > z] \\ &\leq \sum_{i < t_0} \sum_{m=0}^{\infty} (1 - C e^{-\bar{q}t_0})^m \\ &\leq \text{const} \end{aligned}$$

qui entraîne la condition S .

Étape 2 : $S < \infty \Rightarrow H_1$ L'existence de l'espérance implique qu'il existe z assez grand tel que

$$\sup_{n \geq z} \mathbb{E}_n[T_z] < \epsilon$$

en utilisant la méthode de premier moment, on obtient

$$\sup_{n \geq z} \mathbb{P}_n[T_z \geq 1] < \epsilon \Rightarrow \sup_{n \geq z} \mathbb{P}_n[T_z \geq k] < \epsilon^k$$

On veut en plus, le choix de ϵ satisfait que

$$\sup_{n \geq z} \mathbb{E}_n[e^{pT_z}] < \infty$$

où $p = \bar{q} + 1$.

On choisit $x_0 > z$ et $\nu_0 = \delta_{x_0}$ et $K = [1, K] \cup \{x_0\}$. Car $\tau_K \wedge T_0 < T_z$

$$A := \sup_{n \geq z} \mathbb{E}_n[e^{p(\tau \wedge T_0)}] < \infty$$

et on va montrer qu'il existe c_2 tel que

$$\mathbb{P}_{x_0}[T_0 > t] \geq c_2 \mathbb{P}_x[T_0 > t]$$

$$\begin{aligned} \mathbb{P}_x[T_0 > t] &= \mathbb{P}_x[T_0 > T_0 \wedge \tau_K > t] + \mathbb{P}_x[T_0 > t > T_0 \wedge \tau_K] \\ &\leq Ae^{-pt} + \int_0^t \sup_{y \in K \cup \{x_0\}} \mathbb{P}_y[t - s < T_0] \mathbb{P}_x[T_0 \wedge \tau_K \in ds] \\ &\leq Ae^{-pt} + \int_0^t \sup_{y \in K \cup \{x_0\}} \left(\frac{\mathbb{P}_y[t - s < T_0]}{\mathbb{P}_{x_0}[t - s < T_0]} \right) \mathbb{P}_{x_0}[t - s < T_0] \mathbb{P}_x[T_0 \wedge \tau_K \in ds] \\ &\leq Ae^{-pt} + \int_0^t C \mathbb{P}_{x_0}[t < T_0] \frac{\mathbb{P}_{x_0}[t - s < T_0]}{\mathbb{P}_{x_0}[t < T_0]} \mathbb{P}_x[T_0 \wedge \tau_K \in ds] \\ &\leq Ae^{-pt} + \int_0^t C \mathbb{P}_{x_0}[t < T_0] \mathbb{P}_x e^{ps} [T_0 \wedge \tau_K \in ds] \\ &\leq Ae^{-pt} + C \mathbb{P}_{x_0}[t < T_0] \int_0^t e^{ps} \mathbb{P}_x [T_0 \wedge \tau_K \in ds] \\ &\leq A \mathbb{P}_{x_0}[T_0 > t] + AC \mathbb{P}_{x_0}[T_0 > t] \\ &= (1 + C) A \mathbb{P}_{x_0}[T_0 > t] \end{aligned}$$

Étape 3 : $S < \infty \Rightarrow H_2$

$$\begin{aligned} \mathbb{P}_x[\tau < t] &= \mathbb{P}_x[\tau_K < t \wedge T_0] = \mathbb{P}_x[T_0 > t] - \mathbb{P}_x[T_0 > \tau_K > t] \\ &> e^{-\bar{q}t} - Ae^{-pt} \end{aligned}$$

On prend $t = t_0$ assez grand tel que $\inf_{x \in E^*} \mathbb{P}_x[\tau_K < t_0 - 1] > 0$. Car ν est choisi comme δ_{x_0} , il suffit de comparer les deux probabilités sur x_0 . On peut couper la trajectoire en trois parties : elle entre K , puis elle utilise un pas de sauter à x_0 et elle ne quitte plus.

$$\begin{aligned} \inf_{x \in E^*} \mathbb{P}_x[X_{t_0} = x_0] &> \inf_{x \in E^*} \mathbb{P}_x[X_{t_0} = x_0 | X_{\tau_K+1} = x_0] \mathbb{P}_x[X_{\tau_K+1} = x_0 | \tau_K < t_0 - 1] \mathbb{P}_x[\tau_K < t_0 - 1] \\ &> c_1 \delta_{x_0} \end{aligned}$$

Cela finit la démonstration. \square

Le cas logistique vérifie ce critère.

4.3. Résultats numériques. On illustre la convergence par simulation numérique d'un modèle de naissance et mort logistique. Dans ce modèle, les paramètres sont désignés

$$\lambda = 10, \mu = 6, c = 1/4$$

et on fixe $Z_0 = 5$. Avec ce jeu de paramètres, le nombre typique de population est $\frac{\lambda - \mu}{c} = 16$. Les expériences sont réalisées avec 10000 trajectoires.

Ci-dessous une illustration des lois empiriques à chaque instant. Comme les trajectoires évoluent en temps, on voit que la forme converge vers une limite.

Pour comparer la distance totale, on a besoin d'une QSD exacte. Généralement, la QSD n'a pas de formule explicite. Donc, on propose une manière de l'approcher

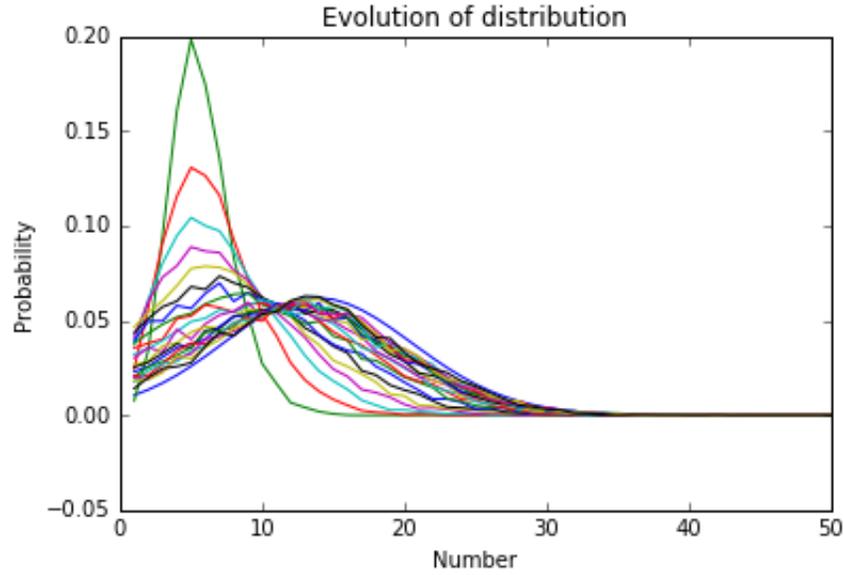


FIGURE 3. Évolution de la fonction de repartition

par la caractérisation spectrale, qui sera précisée dans la sous-section suivante. Les deux images suivantes illustre la différence entre histogramme de QSD et celui de $Z_t|_{T_0>t}$ au temps 0.5 et 1.

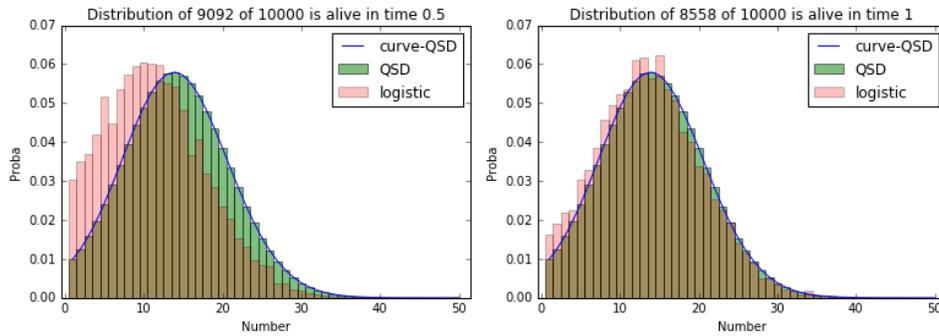


FIGURE 4. Processus de naissance et mort logistique, $c = 1/4$, $\mu = 6$, $\lambda = 10$, histogramme au temps 0.5 à gauche et au temps 1.0 à droite

Finalement, on utilise deux images de dessiner la distance et log-distance entre QSD et $Z_t|_{T_0>t}$ à chaque instant. L'image de log-distance, qui ressemble beaucoup à une droite, suggère que cette convergence est de vitesse exponentielle.

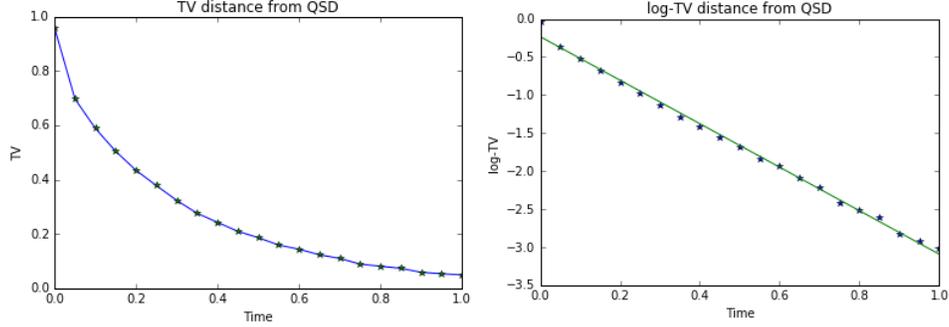


FIGURE 5. Processus de naissance et mort logistique, $c = 1/4$, $\mu = 6$, $\lambda = 10$, distance de QSD et log-distance en fonction de temps

5. SIMULATIONS NUMÉRIQUES

Dans cette section, on discute comment simuler numériquement des distributions quasi-stationnaires, d'un côté dans le cas du temps discret notamment pour le modèle Galton-Watson, et d'autre côté dans le cas du temps continu pour le processus de naissance et mort.

5.1. Une méthode naïve : Monte-Carlo sur les chaînes de Markov. Une idée simple consiste à faire la méthode de Monte-Carlo sur les chaînes de Markov : après avoir lancé un certain nombre de trajectoires qu'on laisse évoluer indépendamment les uns des autres selon leur loi, on ne prend en compte que les trajectoires qui n'atteignent pas 0, puisque les processus sont conditionnés à ne pas s'éteindre.

Comme nous nous plaçons dans le cas où nos processus s'éteignent presque sûrement, La proportion des "trajectoires utiles" pour une simulation des distributions quasi-stationnaires parmi tous les trajectoires, décroît donc de façon exponentielle en temps, d'après les propositions présentées dans les parties précédentes. Ce qui rend cette méthode de Monte-Carlo de chaînes de Markov assez coûteuse, puisque seuls les trajectoires survécus à l'instant final considéré, contribuent à la construction des distributions quasi-stationnaires.

5.2. Approche par système de particules de type de Fleming-Viot. L'approche précédente par Monte-Carlo de chaînes de Markov ayant pour défaut d'être assez coûteuse (en temps de calcul et en espace de mémoire occupé lors de simulation), il est intéressant d'utiliser une approche par système de particules de type Fleming-Viot.

Voici une définition formelle d'un système de particules de type Fleming-Viot.

Définition 5.1. On considère Z_t un processus de Markov fort càdlàg avec extinction, qui évolue sur $E \cup \partial$, où $\partial \notin E$ est un point absorbant. Quand Z arrive à δ , on dit que le processus est mort. Soit $N \geq 2$ et $(x_1, \dots, x_N) \in E^N$. Un système de particule de type Fleming-Viot avec N particules et évoluant comme Z entre leur renaissance, commençant par (x_1, \dots, x_N) , noté par $FV_Z^N(x_1, \dots, x_N)$, est défini de la façon suivante.

- Les N particules du système partent suivant la distribution initiale (x_1, \dots, x_n) en temps $t = 0$

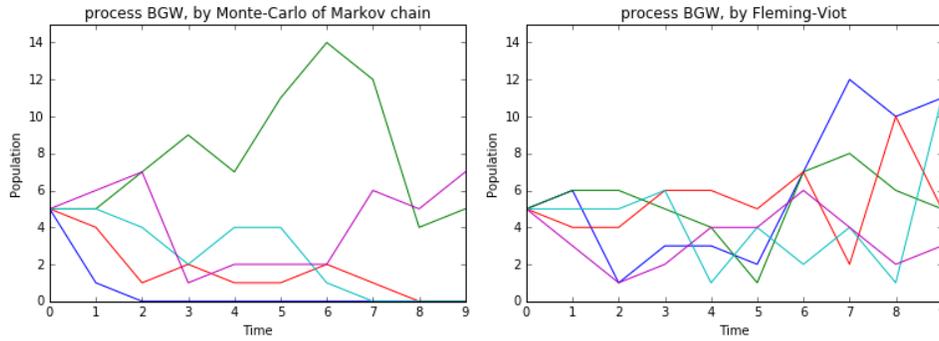


FIGURE 6. Exemple d'une simulation de processus BGW, pour $N = 10$ trajectoires lancées initialement à $Z_0 = 5$, pour $m = 0.9$, par méthode de Monte-Carlo sur chaîne de Markov à gauche, et par Fleming-Viot à droite

- Les N particules évoluent comme N copies indépendantes de Z , jusqu'à ce que l'une d'entre elles meure. Le temps de mort est noté τ_1
- Au temps de mort τ_1 , le système est modifié :
 - si deux ou plusieurs particules s'éteignent simultanément à ce moment, alors on arrête la définition du système de particules de type Fleming-Viot, et on dit que le processus subit un *échec* (En effet, on suppose que ce genre d'évènements ne se produisent presque sûrement pas) ;
 - sinon (c'est-à-dire exactement une particule meurt), la particule morte est déplacée instantanément du point absorbant δ pour arriver à la place d'une particule choisie uniformément parmi les $N - 1$ particules survivées ; dans cette situation on dit que cette particule connaît une *renaissance*.
- Au temps τ_1 et après cette opération de renaissance, chacune des N particules reste dans E . Et elles évoluent comme N copies indépendantes de Z , jusqu'à ce que une prochaine mort tombe. Ce temps de mort est noté τ_2 .
- A ce moment, le système est modifié avec le même mécanisme de renaissance comme précédemment.
- Le système évolue comme N particules indépendantes, et ainsi de suite.

Cette procédure définit un système de particules $FV_Z^N(x_1, \dots, x_N)$ d'une façon progressive. Renaissance après renaissance, jusqu'à ce que le système subisse un échec.

L'article [4] donne un résultat théorique sur la convergence de distribution du système de particules de type Fleming-Viot, vers la loi de distribution conditionnée à ne pas mourir, dans le cas en temps continu, sous certaines hypothèses. On ne précise cette démonstration pas dans ce rapport.

Il est donc légitime d'utiliser un système de particule de type Fleming-Viot pour simuler les distributions quasi-stationnaires dans notre cas : les trajectoires d'évolution de populations sont alors considérées comme particules du système Fleming-Viot.

Ainsi, avec Fleming-Viot, toutes les trajectoires simulées sont "utiles", donc contribuent à la construction des distributions quasi-stationnaires. Par conséquent, il est attendu que les simulations par Fleming-Viot convergent plus rapidement

que par les méthode de Monte-Carle des chaînes de Markov, vers une distribution quasi-stationnaire.

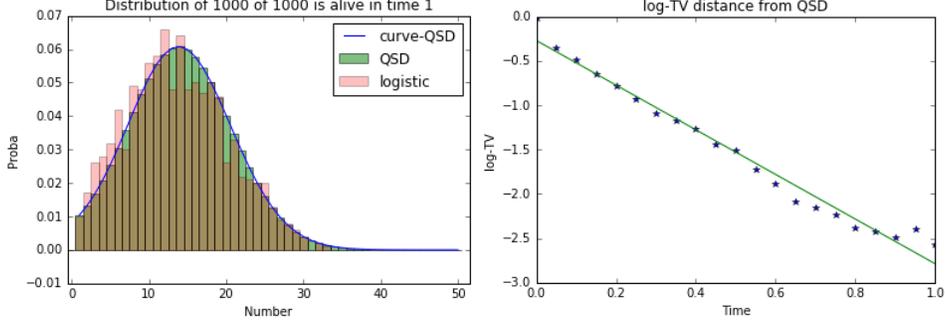


FIGURE 7. Processus de naissance et mort logistique, $c = 1/4$, $\mu = 6$, $\lambda = 10$, simulé par la méthode de Fleming-Viot en utilisant 1000 trajectoires, histogramme au temps 1 à gauche et la distance totale de QSD à droite

5.3. Calculer la loi quasi-stationnaire. Pour illustrer la convergence en distance totale et estimer la vitesse de convergence, la QSD est nécessaire. Mais c'est un vrai défi de calculer la QSD pour un processus en général. Dans cette sous-section, on introduit une méthode heuristique qui s'applique bien dans le cas de processus de naissance et mort. Cette méthode est trouvée par nous-mêmes et elle marche bien en pratique. On donne un peu de justification d'erreurs numériques à la fin.

On rappelle la caractérisation spectrale de QSD 2.6 et l'applique dans le processus de naissance et mort. On obtient la proposition suivante.

Proposition 5.2. *Soit α une distribution de probabilité d'un processus de naissance et mort. α est une QSD si et seulement si*

- $\alpha \geq 0, \forall j \geq 1$ et $\sum_{j \geq 1} \alpha_j = 1$.
- Soit $\theta(\alpha)$ la puissance d'extinction, alors $\forall j \geq 1$,

$$\begin{aligned} \lambda_{j-1}\alpha_{j-1} - (\lambda_j + \mu_j)\alpha_j + \mu_{j+1}\alpha_{j+1} &= -\theta(\alpha)\alpha_j \\ -(\lambda_1 + \mu_1)\alpha_1 + \mu_2\alpha_2 &= -\theta(\alpha)\alpha_1 \\ \mu_1\alpha_1 &= \theta(\alpha) \end{aligned}$$

On voit clairement que l'on est capable de résoudre α lorsque on sait la puissance d'extinction θ . Cependant, 2.10 nous assure que la vitesse de décroissance qui s'approche à θ peut être obtenue à partir de n'importe quelle loi initiale. De cette observation, on propose l'algorithme suivant.

Un algorithme pour calculer QSD

- Simuler N trajectoires de processus i.i.d commençant par une loi initiale ν , que l'on note $X_t^{(1)}, X_t^{(2)}, X_t^{(2)}, \dots, X_t^{(N)}$.
- Discrétiser le temps $[a, b]$ en t_1, t_2, \dots, t_M où $t_i = \frac{i(b-a)+a}{M}$. Calculer la distribution empirique conditionnée à la survie

$$\hat{Y}_{t_n}^N = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{X_{t_n}^{(i)} \neq 0}$$

- Faire une régression sur $\hat{Z}_{t_n}^N = -\log \hat{Y}_{t_n}^N$. On note la puissance d'extinction empirique $\hat{\theta}^N(\alpha)$
- Résoudre le vecteur propre $\hat{\alpha}^N$ de l'équation linéaire

$$\hat{\alpha}^N L = -\hat{\theta}^N(\alpha) \hat{\alpha}^N$$

On justifie la convergence de cette méthode.

Proposition 5.3. *On note $\hat{\theta}$ obtenu dans l'algorithme ci-dessus, puis on a une convergence*

$$\lim_{a \rightarrow \infty, N \rightarrow \infty} \hat{\theta}^N(\alpha) = \theta(\alpha)$$

En plus, on a une convergence de distance totale

$$\|\hat{\alpha}^N - \alpha\|_{TV} \xrightarrow[N \rightarrow \infty, a \rightarrow \infty]{p.s.} 0$$

dans le processus de naissance et mort logistique de puissance fractale dont $\mu_n = n\mu + cn(n-1)^\gamma$

Démonstration. **Étape 1** D'après le théorème des grands nombres, on a

$$\hat{Z}_{t_n}^N \xrightarrow[p.s.]{N \rightarrow \infty} -\log \mathbb{P}_\nu[T_0 > t_n] = -\log[\nu(\eta)] + \theta(\alpha)t_n + \epsilon_{t_n}^1$$

On appelle $\epsilon_{t_n}^1$ erreur d'approximation et il converge vers 0 lorsque t_n tend vers ∞ .

Étape 2 Pour l'intervalle $[a, b]$ et une discrétisation fixée, on note \hat{Z}_M^N, \hat{Y}_M^N le vecteur de variable aléatoire. On utilise le théorème limite centrale et la méthode de substitution

$$\sqrt{N}(\hat{Z}_M^N - (-\log \mathbb{E}(\hat{Y}_M^N))) \xrightarrow[N \rightarrow \infty]{Loi} \mathcal{N}(0, \mathcal{C})$$

Autrement dit,

$$\hat{Z}_M^N \stackrel{Loi}{=} -\log[\nu(\eta)] + \theta(\alpha)t + \epsilon^1 + \epsilon^2$$

où ϵ^2 est erreur statistique, et quand N est très grand, $\epsilon^2 \sim \frac{1}{\sqrt{N}} \mathcal{N}(0, \mathcal{C})$.

Étape 3 On utilise l'équation normale d'implémenter la régression.

$$\hat{\theta}^N(\alpha) = \theta + \frac{1}{\text{var}(t)} \text{cov}(t, \epsilon^1) + \frac{1}{\text{var}(t)} \text{cov}(t, \epsilon^2)$$

Ici, cov signifie la covariance entre des terme d'un vecteur et $\text{var}(t) = \text{cov}(t, t)$. Par exemple,

$$\text{cov}(t, \epsilon^1) = \frac{1}{M} \sum_{n=1}^M t_n \epsilon_{t_n}^1 - \left(\frac{1}{M} \sum_{n=1}^M t_n \right) \left(\frac{1}{M} \sum_{n=1}^M \epsilon_{t_n}^1 \right)$$

Clairement, on voit quand N tend vers ∞ , l'erreur statistique ϵ^2 tend vers 0. Puis, quand $a \rightarrow \infty$, l'erreur d'approximation ϵ^1 tend vers 0.

Étape 4 La dernière étape consiste à analyser la stabilité dans le schéma numérique qui résout α par récurrence. On note les erreurs

$$\begin{aligned} \Delta \hat{\alpha}_j^N &= \hat{\alpha}_j^N - \alpha_j \\ \Delta \hat{\theta}^N(\alpha) &= \hat{\theta}^N(\alpha) - \theta(\alpha) \end{aligned}$$

On somme les identités et obtient

$$\begin{aligned}\mu_{n+1}\hat{\alpha}_{n+1}^N &= \mu_1\hat{\alpha}_1^N + \lambda_n\hat{\alpha}_n^N - \hat{\theta}^N \sum_{k=1}^n \hat{\alpha}_k^N \\ \mu_{n+1}\Delta\hat{\alpha}_{n+1}^N &= \mu_1\Delta\hat{\alpha}_1^N + \lambda_n\Delta\hat{\alpha}_n^N - \hat{\theta}^N \sum_{k=1}^n \hat{\alpha}_k + \theta \sum_{k=1}^n \alpha_k \\ &= \mu_1\Delta\hat{\alpha}_1^N + \lambda_n\Delta\hat{\alpha}_n^N - \hat{\theta}^N \sum_{k=1}^n \Delta\hat{\alpha}_k^N - \Delta\hat{\theta}^N \sum_{k=1}^n \alpha_k\end{aligned}$$

On fixe K assez grand tel que $cK^\gamma > \lambda + 1 + \beta\hat{\theta}^N$ et on prend $\Delta\hat{\theta}^N$ assez petit tel que

$$\begin{aligned}\mu_{K+1}\Delta\hat{\alpha}_{K+1}^N &\leq \mu_1|\Delta\hat{\alpha}_1^N| + \lambda_K|\Delta\hat{\alpha}_K^N| + \hat{\theta}^N \sum_{k=1}^K |\Delta\hat{\alpha}_k^N| + |\Delta\hat{\theta}^N| \sum_{k=1}^K \alpha_k \\ &\leq \mu_{K+1} \frac{C\Delta\hat{\theta}^N}{(K+1)^\beta}\end{aligned}$$

Alors, on fait la récurrence de montrer que $\Delta\hat{\alpha}_n^N \leq \frac{C\Delta\hat{\theta}^N}{n^\beta}$, $1 + \gamma > \beta > 1$ pour $n \geq K + 1$.

$$\begin{aligned}\mu_{n+1}\Delta\hat{\alpha}_{n+1}^N &\leq \mu_1|\Delta\hat{\alpha}_1^N| + \lambda_n|\Delta\hat{\alpha}_n^N| + \hat{\theta}^N \sum_{k=1}^n |\Delta\hat{\alpha}_k^N| + |\Delta\hat{\theta}^N| \sum_{k=1}^n \alpha_k \\ &\leq \left(\frac{1}{(K+1)^\beta} + \frac{\lambda}{n^{\beta-1}} + \frac{\beta\hat{\theta}^N}{(K+1)^{\beta-1}}\right)(C\Delta\hat{\theta}^N) \\ &\leq \left(\frac{1}{(K+1)^\beta} + \frac{\lambda}{(K+1)^{\beta-1}} + \frac{\beta\hat{\theta}^N}{(K+1)^{\beta-1}}\right)(C\Delta\hat{\theta}^N) \\ &\leq \frac{cK^\gamma}{(K+1)^{\beta-1}}(C\Delta\hat{\theta}^N) \leq \frac{cn^\gamma}{(n+1)^{\beta-1}}(C\Delta\hat{\theta}^N) \\ &\leq \mu_{n+1} \frac{C\Delta\hat{\theta}^N}{(n+1)^\beta}\end{aligned}$$

Donc, $\|\hat{\alpha}^N - \alpha\|_{TV} \leq \sum_{n \geq K+1} \frac{C\Delta\hat{\theta}^N}{n^\beta} + \frac{C\Delta\hat{\theta}^N}{K^\beta}$ qui tend vers 0 lorsque $\Delta\hat{\theta}^N$ tend vers 0. \square

Remarque. Pour illustrer la convergence vers $\hat{\alpha}^N$, on doit re-simuler des trajectoires indépendantes de celles qui calculent $\hat{\alpha}^N$.

RÉFÉRENCES

1. Méléard, S., Modèles aléatoires en Écologie et Évolution, Edition École polytechnique
2. Méléard, S., Villemonais, D. (2012). Quasi-stationary distributions and population processes. Probab. Surv, 9(340-410), 112.
3. Champagnat, N., Villemonais, D. (2016). Exponential convergence to quasi-stationary distribution and Q-process. Probability Theory and Related Fields, 164(1-2), 243-283.
4. Villemonais, D. (2011), Interacting particle processes and approximation of Markov processes conditioned to not be killed, <hal-00598085v1>

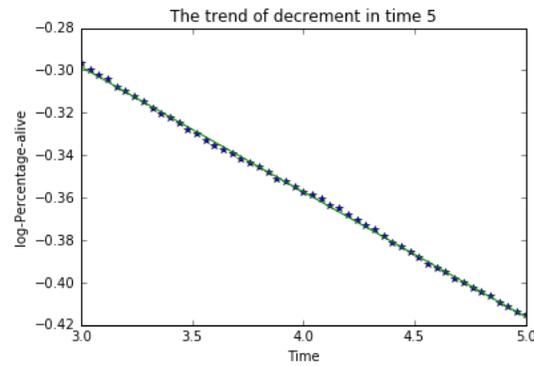


FIGURE 8. Évolution de la fonction de repartition

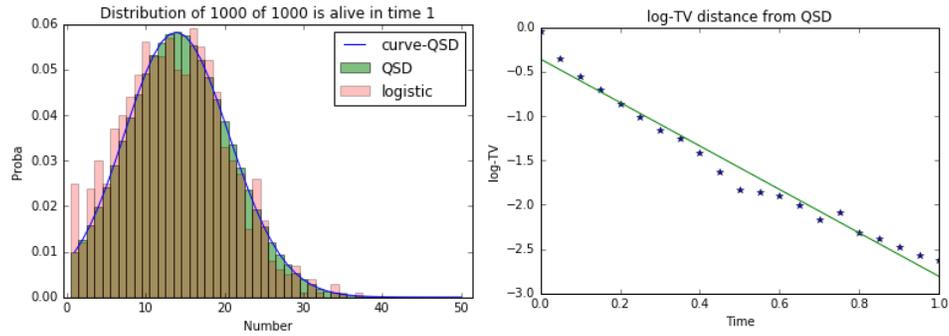


FIGURE 9. Processus de naissance et mort logistique de puissance fractale dont $\mu_n = n\mu + cn(n-1)^\gamma$, $c = 1/4$, $\mu = 6$, $\lambda = 10$, $\gamma = 0.5$, simulé par la méthode de Fleming-Viot en utilisant 1000 trajectoires, histogramme au temps 1 à gauche et la distance totale de QSD à droite

ANNEXE : CODES PYTHON POUR LES SIMULATIONS

Processus Galton-Watson

```
# -*- coding: utf-8 -*-
"""
Created on Mon Jan 16 21:58:34 2017

@author: Yiyang
"""

import numpy as np
import numpy.random as npr

'''
On se place d'abord dans le cas discret,
en faisant appel au modele Bienayme-Galton-Watson
'''
```

```

def BGM(p,M,N,init):
    '''
        entree
    p = loi de reproduction , a valeurs dans un ensemble fini
    M = nombre de simulations
    N = nombre de generations maximal
        sortie
    traj = matrice de taille M*N notant les trajectoires
    tableSurvie = nombre de trajectoires >0 au cours du temps
    liste = numero de trajectoires >0 au temps final
    '''

    traj = np.zeros([M,N])
    nbTrajSurvie = M
    tableSurvie = M * np.ones(N)
    traj[:,0] = init

    liste = list(range(M))

    probaCumul = np.array(np.cumsum(p, axis=1))
    k = np.size(probaCumul)

    for n in range(1,N): # n pour nieme generation
        # reproduction de type Galton-Watson
        for m in range(M):
            X = int(traj[m,n-1])
            if X>0:
                r = npr.rand(1,X)
                R = np.dot(np.ones([k,1]), r)
                P = np.dot(probaCumul.T, np.ones([1,X]))
                z = np.sum(R>P, axis=0)
                Xnew = np.sum(z)
                if (Xnew == 0):
                    nbTrajSurvie = nbTrajSurvie -1
                    liste.remove(m)
            else:
                Xnew = 0
            traj[m,n] = Xnew
            tableSurvie[n] = nbTrajSurvie

    return traj , tableSurvie , liste

def FlemViot(p,M,N,init):
    '''
        entree
    p loi de reproduction , a valeurs dans un ensemble fini
    M nombre de simulations
    N = nombre de generations maximal
        sortie
    traj = matrice de taille M*N notant les trajectoires
    tableSurvie = nombre de trajectoires >0 au cours du temps
    liste = numero de trajectoires >0 au temps final
    '''

```

```

'''
traj = np.zeros([M,N])
traj[:,0] = init

probaCumul = np.array(np.cumsum(p, axis=1))
k = np.size(probaCumul)

for n in range(1,N): # n pour nieme generation
    # reproduction de type Galton-Watson
    for m in range(M):
        X = int(traj[m,n-1])
        # X>0 by construction
        r = npr.rand(1,X)
        R = np.dot(np.ones([k,1]),r)
        P = np.dot(probaCumul.T, np.ones([1,X]))
        z = np.sum(R>P, axis=0)
        Xnew = np.sum(z)
        traj[m,n] = Xnew
    #resuiciter de type Fleming-Viot
    for m in range(M):
        while traj[m,n]==0:
            traj[m,n] = traj[int(npr.rand()*M),n]
return traj
'''

```

Processus naissance et mort

```

# -*- coding: utf-8 -*-
'''
Created on Sun Jan 29 10:28:55 2017

@author: Chenlin
'''

import numpy as np
import numpy.random as npr
import matplotlib.pyplot as plt

'''
#####
#####
This function simulates the linear BD process
t_jump = jump time
t_dead = the time of death. If it is alive, t_dead = T
n_jump = the size at jump time
#####
#####
'''

def BD_linear(lam, mu, T, M, init):
    plus = lam/(lam + mu)
    n_jump = []
    t_jump = []
    t_dead = []
    for i in range(M):
        n_temp = [init[i]]

```

```

t_temp = [0]
t_wait = npr.exponential() / (n_temp[-1] * (lam + mu))
while t_temp[-1] + t_wait < T:
    t_temp.append(t_temp[-1] + t_wait)
    jump = npr.rand()
    if jump < plus:
        n_temp.append(n_temp[-1] + 1)
    else:
        n_temp.append(n_temp[-1] - 1)
    if n_temp[-1] == 0:
        break
    t_wait = t_wait = npr.exponential() / (n_temp[-1] * (lam +
        mu))

if n_temp[-1] == 0:
    t_dead.append(t_temp[-1])
else:
    t_dead.append(T)
t_jump.append(t_temp)
n_jump.append(n_temp)
return t_jump, n_jump, t_dead

'''
#####
#####
This function simulates the logistic BD process
t_jump = jump time
t_dead = the time of death. If it is alive, t_dead = T
n_jump = the size at jump time
#####
#####
'''

def BD_logistic(lam, mu, c, T, M, init):
    plus = lam/(lam + mu)
    n_jump = []
    t_jump = []
    t_dead = []
    for i in range(M):
        n_temp = [init[i]]
        t_temp = [0]
        plus_taux = n_temp[-1] * lam
        minus_taux = n_temp[-1] * mu + c * n_temp[-1] * (n_temp[-1] -
            1)
        t_wait = npr.exponential() / (plus_taux + minus_taux)
        plus = plus_taux / (plus_taux + minus_taux)
        while t_temp[-1] + t_wait < T:
            t_temp.append(t_temp[-1] + t_wait)
            jump = npr.rand()
            if jump < plus:
                n_temp.append(n_temp[-1] + 1)
            else:
                n_temp.append(n_temp[-1] - 1)

```

```

    if n_temp[-1] == 0:
        break
    plus_taux = n_temp[-1] * lam
    minus_taux = n_temp[-1] * mu + c * n_temp[-1] * (n_temp
        [-1] - 1)
    t_wait = npr.exponential() / (plus_taux + minus_taux)
    plus = plus_taux / (plus_taux + minus_taux)

    if n_temp[-1] == 0:
        t_dead.append(t_temp[-1])
    else:
        t_dead.append(T)
    t_jump.append(t_temp)
    n_jump.append(n_temp)
return t_jump, n_jump, t_dead

'''
#####
#####
This function simulates the logistic BD process
t_jump = jump time
t_dead = the time of death. If it is alive, t_dead = T
n_jump = the size at jump time
birth = lam * i
death = mu * i + c*i*(i-1)
#####
#####
'''

def BD_logistic_FV(lam, mu, c, T, M, init):
    n_jump = []
    t_jump = []
    t_dead = []
    for i in range(M):
        n_temp = [init[i]]
        t_temp = [0]
        n_jump.append(n_temp)
        t_jump.append(t_temp)
        t_dead.append(T)
    t = 0
    X = init
    b = lam*X
    d = mu*X + c*X*(X-1)
    taux = b+d
    dt = npr.exponential() / (np.sum(b) + np.sum(d))
    while dt + t < T:
        t += dt
        i = np.sum(npr.rand() * np.sum(taux) > np.cumsum(taux))
        t_jump[i].append(t)

        if npr.rand() * taux[i] < b[i] :
            X[i] += 1
        else:

```

```

        if X[i] > 1:
            X[i] -= 1
        else:
            X[i] = X[int(npr.rand()*M)]
    n_jump[i].append(X[i])

    b = lam*X
    d = mu*X + c*X*(X-1)
    tau_x = b+d
    dt = npr.exponential() / (np.sum(b) + np.sum(d))

    return n_jump, t_jump, t_dead

'''
This function counts the proba of each number
N = the up bound
'''
def stat_alive(n, N):
    u = np.array(range(N))
    M = np.sum(n > 0) + 0.0
    for i in range(1,N):
        u[i] = np.sum(n == i)
    u = u/M
    return u

'''
This function calculates the TV distance
'''
def TV(u1, u2):
    return np.sum(np.abs(u1 - u2))*0.5

'''
#####
This calculates the conditional distribution at each moment.
Then it draws the evolution of distance total variation.
n_t = the number alive at each time
u_t = the probability
disTV = the distance from te QSD
#####
'''
def TV_conv(alpha, t_jump, n_jump, T, M, N):
    Num = int(20*T+1)
    t = np.linspace(0, T, Num)
    n_t = np.zeros([M,Num])
    for j in range(M):
        t_temp = np.array(t_jump[j])
        for i in range(Num):
            index = np.sum(t_temp <= t[i]) - 1
            n_t[j, i] = n_jump[j][index]

    u_t = np.zeros([N,Num])
    disTV = np.zeros(Num)
    for i in range(Num):

```

```

    u_t[:, i] = stat_alive(n_t[:, i], N)
    disTV[i] = TV(alpha, u_t[:, i])

log_disTV = np.log(disTV)
t_poly = np.transpose(np.array([np.ones(Num), t]))
coeff = np.dot(np.dot(np.linalg.inv(np.dot(t_poly.T, t_poly)),
    t_poly.T), log_disTV)
cc = -coeff[1]

plt.figure()
plt.plot(range(1,N), alpha[1:])
for i in range(1, Num):
    plt.plot(range(1,N), u_t[1:, i])
plt.title('Evolution_of_distribution')
plt.xlabel('Time')
plt.ylabel('Probability')

plt.figure()
plt.plot(t, disTV)
plt.plot(t, disTV, '*')
plt.title('TV_distance_from_QSD')
plt.xlabel('Time')
plt.ylabel('TV')

log_disTV_reg = np.dot(t_poly, coeff)
plt.figure()
plt.plot(t, log_disTV, '*')
plt.plot(t, log_disTV_reg)
plt.title('log-TV_distance_from_QSD')
plt.xlabel('Time')
plt.ylabel('log-TV')
return n_t, u_t, disTV, cc

'''
#####
This function draws the distribution of time t
#####
'''
def DrawDistribution(n, T, M, alpha):
    N = np.shape(alpha)[0]
    alive = np.sum(n>0)
    alive_proba = stat_alive(n,N)
    plt.figure()
    plt.plot(np.array(range(1,N)), alpha[1:], label = 'curve-QSD')
    plt.bar(np.arange(N) - 0.5, alpha, width=1, alpha = 0.5, facecolor
        = 'g', label = 'QSD')
    plt.bar(np.arange(N) - 0.5, alive_proba, width=1, alpha = 0.25,
        facecolor='r', label = 'logistic')
    plt.xlim(-0.5, N + 0.5)
    #plt.hist(alive_size, bins, alpha=0.5, normed = 'true', label = '
        logistic', color = 'r')

```

```

#plt.hist(alive_proba, bins, alpha=0.5, normed = 'False', label =
'logistic', color = 'r')
plt.legend()
plt.title('Distribution_of_' + str(alive) + "_of_" + str(M) + '_is
alive_in_time_' + str(T))
plt.ylabel('Proba')
plt.xlabel('Number')
return alive, alive_proba

'''
#####
This function calculates the quasi-stationary distribution
#####
'''
def QuasiApproximation(theta, b, d):
N = np.shape(b)[0]
alpha = np.zeros(N)
alpha[1] = theta/d[1]
alpha[2] = 1.0/d[2] * ((b[1] + d[1])*alpha[1] - theta * alpha[1])
for j in range(2, N-1):
alpha[j+1] = 1.0/d[j+1] * ((b[j] + d[j] - theta) * alpha[j] -
b[j-1] * alpha[j-1])
return alpha

'''
#####
This function draws decrease and also does regressions
#####
'''
def DrawDecrease(t_jump, n_jump, t_dead, T, M):
Num = 10 * int(np.floor(T)) + 1
t = np.linspace(3, T, Num)
n_alive = np.zeros(Num)
for i in range(Num):
alive_index = np.array(t_dead) >= t[i]
alive = np.sum(alive_index)
n_alive[i] = alive

p_log = np.log(n_alive/float(M))
t_poly = np.transpose(np.array([np.ones(Num), t]))
coeff = np.dot(np.dot(np.linalg.inv(np.dot(t_poly.T, t_poly)),
t_poly.T), p_log)
theta = -coeff[1]

'''
plt.figure()
plt.plot(t, n_alive)
plt.title('The trend of decrement in time ' + str(T))

plt.figure()
plt.yscale('log')
plt.plot(t, n_alive)
plt.title('The trend of decrement in time ' + str(T))

```

```
'''  
p_log_reg = np.dot(t_poly, coeff)  
plt.figure()  
plt.plot(t, p_log, '*')  
plt.plot(t, p_log_reg)  
plt.title('The_trend_of_decrement_in_time_' + str(T))  
plt.xlabel('Time')  
plt.ylabel('log-Percentage-alive')  
return n_alive, p_log, theta
```

CMAP UMR 7641 ÉCOLE POLYTECHNIQUE CNRS, ROUTE DE SACLAY, 91128 PALAISEAU
CEDEX FRANCE
E-mail address: chenlin.gu@polytechnique.edu

CMAP UMR 7641 ÉCOLE POLYTECHNIQUE CNRS, ROUTE DE SACLAY, 91128 PALAISEAU
CEDEX FRANCE
E-mail address: yiyang.yu@polytechnique.edu