

Introduction aux Modèles Stochastiques de Dynamique de Population

Fabien Campillo

<http://www-sop.inria.fr/mere/personnel/campillo/module-ed-2010>



Introduction

Premiers pas en R

Processus de naissance et mort

Processus de naissance/mort linéaire

Processus de naissance/mort logistique

Processus de Markov de saut pur $\rightarrow \mathbb{N}$

Chemostat

IBM

Perspectives

Bibliographie

plan général

Introduction

Premiers pas en R

Processus de naissance et mort

Processus de naissance/mort linéaire

Processus de naissance/mort logistique

Processus de Markov de saut pur $\rightarrow \mathbb{N}$

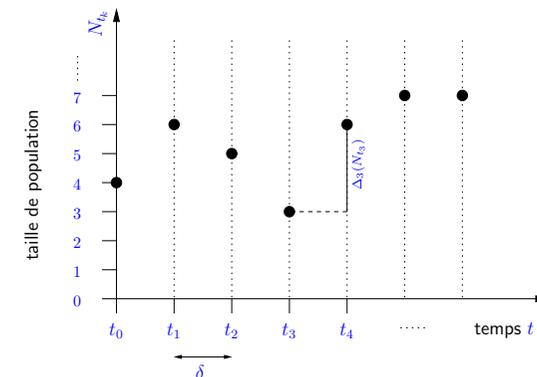
Chemostat

IBM

Perspectives

Bibliographie

dynamique de population



- ▶ N_{t_k} taille de la population à l'instant $t_k = k h$
- ▶ modéliser évolution taille \rightarrow modéliser incrément $\Delta_k(N_{t_k})$

$$N_{t_{k+1}} = N_{t_k} + \Delta_k(N_{t_k}) = N_{t_k} + B - D$$

B nombre de naissances / D nombre de morts

vers l'ODE

- ▶ incrément par unité de temps

$$\frac{N_{t_{k+1}} - N_{t_k}}{h} = \frac{\Delta_k(N_{t_k})}{h}$$

- ▶ lorsque $h \rightarrow 0$

- $t_k \rightarrow t, N_{t_k} \rightarrow N_t$
- $\Delta_k(N_{t_k}) \rightarrow 0$ t.q. $\frac{1}{h} \Delta_k(N_{t_k}) \rightarrow F(N_t)$
- $\frac{1}{h} (N_{t_{k+1}} - N_{t_k}) \rightarrow \dot{N}_t$

- ▶ modèle d'EDO (équation différentielle ordinaire)

$$\dot{N}_t = F(N_t)$$

changement d'échelle (rescaling)

- ▶ lorsque N_t est grand \rightarrow changement d'échelle

$$x_t \stackrel{\text{déf}}{=} \frac{N_t}{V}$$

V (grand) peut représenter

- 1/masse d'une bactérie $\rightarrow x_t$ biomasse
- volume (ou surface) dans lequel évolue la population $\rightarrow x_t$ densité de population
- changement d'échelle artificiel

- ▶ ainsi $\frac{\dot{N}_t}{V} = \frac{1}{V} F(V \frac{N_t}{V})$ posons $b(x) \stackrel{\text{déf}}{=} \frac{1}{V} F(V x)$

$$\dot{x}_t = b(x_t)$$

x_t est la **taille** de la population ou sa **biomasse** ou sa **densité**

- ▶ souvent $b(x) = r(x) x$ où $r(x)$ est un taux de croissance per capita

où se niche l'aléa ?

- ▶ dans la **nature**
 - physique quantique : une information (observable) sur un système physique prend aléatoirement une valeur dans un ensemble donné
 - naissance/mort, génétique
 - comportement animal, humain...
- ▶ dans les **mesures**
- ▶ dans les **modèles** : pour simplifier un modèle on "cache" certaines complexités dans le modèle, ex. le lancé de dé
- ▶ dans les **algorithmes** : méthodes de Monte Carlo

but du cours

- ▶ comment introduire l'aléa ?
 - au niveau microscopique
 - chaque individu peut mourir ou donner naissance à un autre individu
 - on se place à l'échelle de ces événements de naissance/mort
 - ces événements sont aléatoires
- ▶ idée :
 - grandes tailles de population : modèles déterministes
 - petites tailles de population : modèles stochastiques
- ▶ programme :
 - processus de naissance et mort
 - modèle stochastique de Markov
 - comment faire lien déterministe/stochastique ?
 - utiliser **R**
- ▶ travail en commun avec Marc Joannides (UM2)

Introduction

Premiers pas en R

Processus de naissance et mort

Processus de naissance/mort linéaire

Processus de naissance/mort logistique

Processus de Markov de saut pur $\rightarrow \mathbb{N}$

Chemostat

IBM

Perspectives

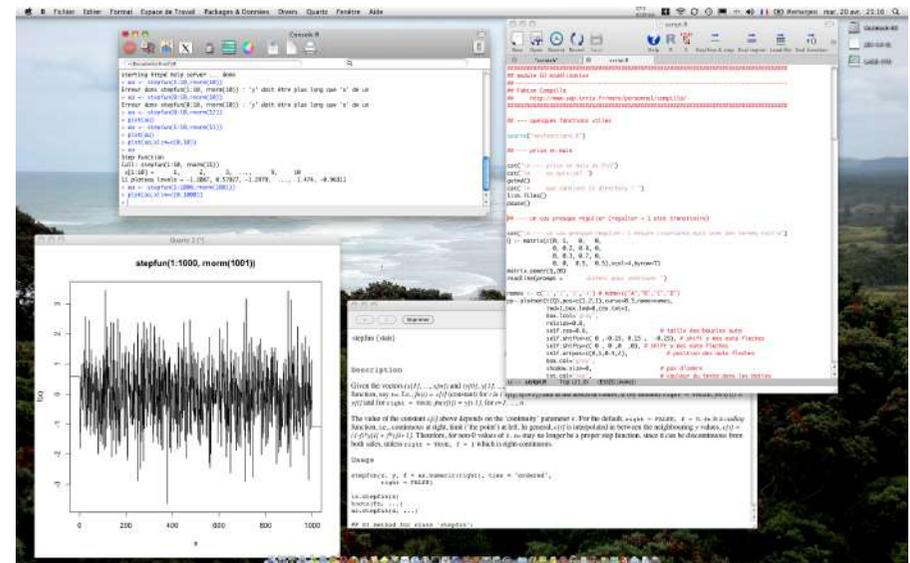
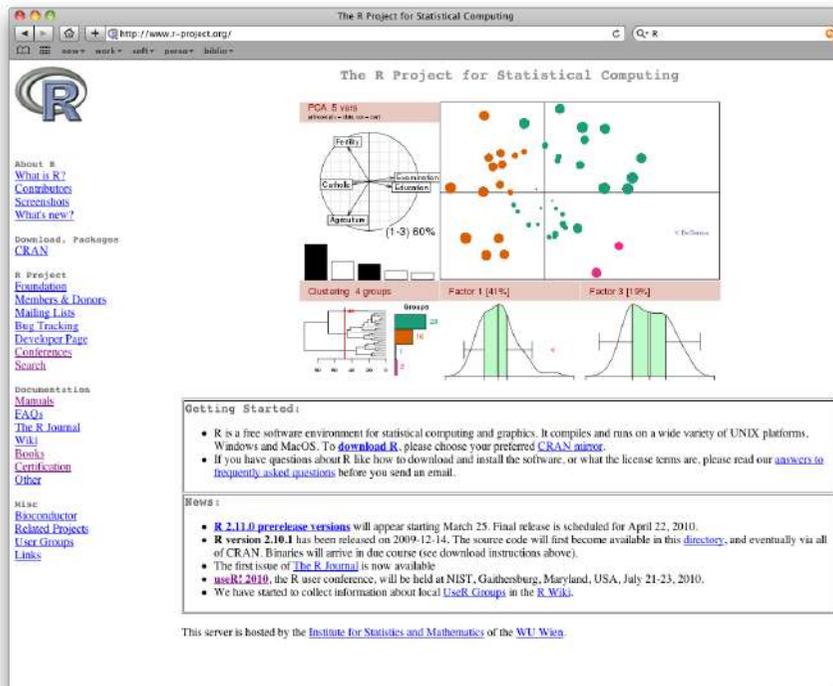
Bibliographie

► calcul numérique

- **matlab** : payant
- **octave** : gratuit
- **scilab** : gratuit
- **R** : gratuit, orienté statistique

► calcul symbolique/numérique

- **maple** : payant
- **mathematica** : payant
- **maxima** : gratuit



- ▶ R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.

☺ : c'est riche
☹ : c'est riche

- ▶ des sites

<http://www.r-project.org/>
<http://www.r-bloggers.com/>
<http://www.rseek.org/>

premier script

- ▶ trouver l'éditeur
- ▶ y inclure la ligne
`cat("Hello world!\n")`
- ▶ sauver le buffer sous le nom `script.R`
- ▶ dans la fenêtre R faire
`source("script.R")`
- ▶ idée : éditer le fichier `script.R` et le "sourcer"

- ▶ pour commencer

```
quit()           # sortir du R
getwd()          # où suis-je?
setwd("toto")    # changer de working directory
dir()            # contenu du working directory
source("fichier.R") # charger un fichier dans la session
source("fichier.R",echo=T) # <<< idem avec echo
```

- ▶ un peu d'aide

```
help(toto)       # aide sur une toto
help.search("toto") # aide sur le mot clé toto
help.start()     # <<< aide dans un browser
example(toto)    # exemples sur certaines fonctions
```

- ▶ opérations de base

```
x <- 2           # affectation
y <- c(x,3)      # concaténation
```

graphiques (script1.R)

```
pause <- function(){
  ## fonction de pause
  readline(prompt = "Pause. Press <Enter> to continue...")
}
plot.new()           # creer un nouveau trace
curve(sin,xlim=c(0,2*pi),col="blue") # xlim les limites du graphiques
pause()
curve(cos,xlim=c(0,2*pi),col="red",add=T) # superposer un graphique
pause()
curve(10*exp(-0.3*x),xlim=c(0,20),col=4) # x est le nom de variable
pause()
xx <- 4*(0:5)        # ou bien seq(0,20,by=4)
yy <- 10*exp(-0.3*xx)
points(xx,yy)
pause()
lines(xx,yy,col="red")
pause()
plot(xx,yy,type="l",col="red") # creer le graphique
matplot(xx,yy,type="s",col="green",add=T) # stairs
points(xx,yy,pch="P",col="blue") # pch plotting character
```

fonction/graphiques (script2.R) I

```
logistique<- fonction(temps_t,x0,taux,K) { # fonction logistique
  K / (1 + (K/x0-1) * exp(-taux * temps_t))
}

tmin <- 0
tmax <- 3
temps <- seq(tmin,tmax,length=200) # suite des instants
x0 <- 20
ttaux <- c(1,2,3,4,5) # différentes valeurs du taux
K <- 400

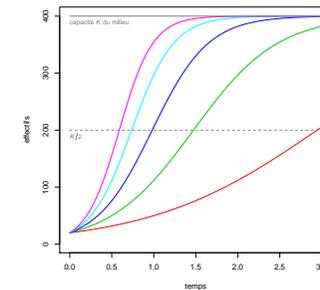
plot(logistique(temps,ttaux[1],x0,K),
     xlim=c(tmin,tmax), # limites en x du trace
     xlab="temps", # titre axe x
     ylab="effectifs", # titre axe y
     type="n") # pas de trace type="n"

for (i in 1:5){
  lines(temps,logistique(temps,x0,ttaux[i],K),xlim=c(tmin,tmax),col=i+1) # courbe de
```

fonction/graphiques (script2.R) II

```
lines(c(tmin,tmax),c(K,K),type="l",lty=1,col="gray50") # asymptote
text(tmin,K,expression(paste("capacité ",italic(K)," du milieu")),
     adj = c(0,1.5),col="gray50", cex=0.8)
lines(c(tmin,tmax),c(K/2,K/2),type="l",lty=2,col="gray50") # K/2
text(tmin,K/2,expression(italic(K/2)),adj = c(0,1.5),col="gray50", cex=0.8)

dev.copy2pdf(file="script2.pdf")
```



ODE (script3.R) I

- ▶ d'abord installer le package `deSolve`
> `install.packages("deSolve")`
puis sourcer `script3.R`

```
library(deSolve) # solveur d'EDO (fonction ode)
```

```
LVmod <- function(Time, State, Pars){
  with(as.list(c(State, Pars)), {
    dProie.dt <- ra*Proie - rb*Proie*Predateur
    ## CroissanceProie - Ingestion
    dPredateur.dt <- rc*Proie*Predateur - rd*Predateur
    ## Prédation - MortPredateur
    return(list(c(dProie.dt, dPredateur.dt)))
  })
}
```

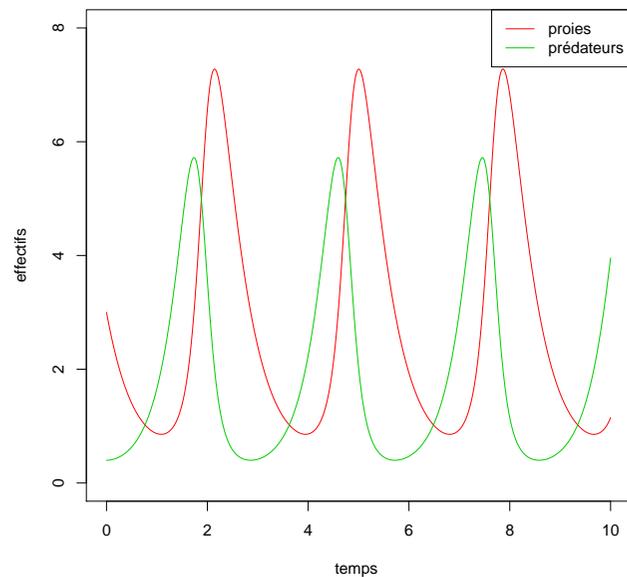
ODE (script3.R) II

```
tmin <- 0 # temps initial de simulation
tmax <- 10 # temps final de simulation
nmax <- 1000 # nombre de pas de discrétisation en temps
temps <- seq(tmin,tmax, length=nmax) # suites des instants
```

```
## taux: croissance des proies, ingestion, efficacité de prédation, mort des prédateurs
parametres <- c(ra=3,rb=1,rc=1,rd=2)
yini <- c(Proie = 0.4, Predateur = 3) # valeurs initiales
out <- as.data.frame(ode(func = LVmod, y = yini,
  parms = parametres, times = temps))
```

```
yrange <- round(range(0,8,out$Proie,out$Predateur))
plot(temps,out$Proie,type = "n",
     ylim=yrange,
     xlab = "temps ",
     ylab = "effectifs")
lines(temps,out$Predateur,type = "l",lwd = 1,col=2)
lines(temps,out$Proie,type = "l",lwd = 1,col=3)
legend("topright",c("proies","prédateurs"),lty=1,col=2:3)
dev.copy2pdf(file="script3.pdf")
```

ODE (script3.R) III



aléatoire (script4.R) I

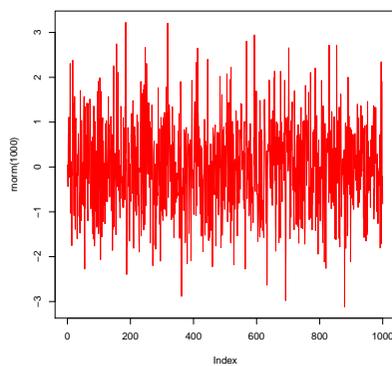
```
vect <- seq(3, 12)
sample(vect, 5)           # échantillon de 5 valeurs dans vectt sans remise
sample(vect, 5, replace=T) # idem avec remise
sample(vect)             # permutation
sample(10, 5)            # idem sample(seq(1,10),5)
## selon une loi discrete donnee
sample(seq(1, 4), 3, prob=c(0.1,0.2,0.3,0.4), replace=T)

rnorm(10, 0, 1)          # 10 loi gaussienne N(0,1) attention:ecart-type
rnorm(10, mean=0, sd=1)  # idem
rnorm(10, mean=c(0,3), sd=c(0.5,1)) # bimodale gaussienne

runif(10, 0, 5)          # 10 loi uniforme U[0,5]
runif(10, min=0, max=5)  # idem
runif(10, c(0, 1), c(5, 7)) # superposition de 2 uniformes

plot(rnorm(1000),type="l",col="red")
dev.copy2pdf(file="script4.pdf")
```

aléatoire (script4.R) II



► autres

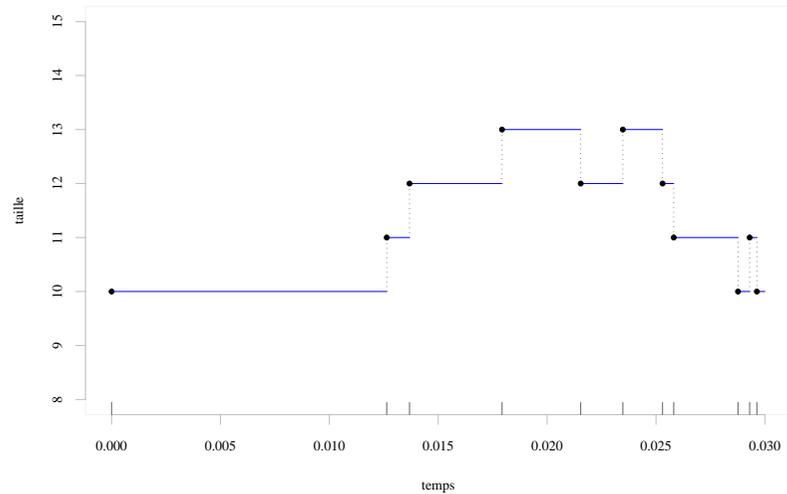
```
[dpqr]exp(x, rate=1)      : loi exponentielle.
[dpqr]binom(x, size, prob) : loi binomiale
[dpqr]dpois(x, lambda)    : loi de poisson
```

d densité | p fonction de répartition | q quantile | r tirage

plan général

- Introduction
- Premiers pas en R
- Processus de naissance et mort
- Processus de naissance/mort linéaire
- Processus de naissance/mort logistique
- Processus de Markov de saut pur $\rightarrow \mathbb{N}$
- Chemostat
- IBM
- Perspectives
- Bibliographie

une trajectoire



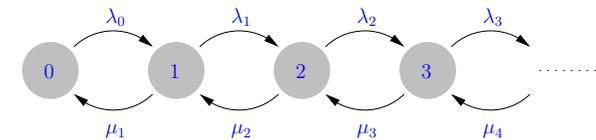
définition

► avec h petit et conditionnellement à $N_t = i$

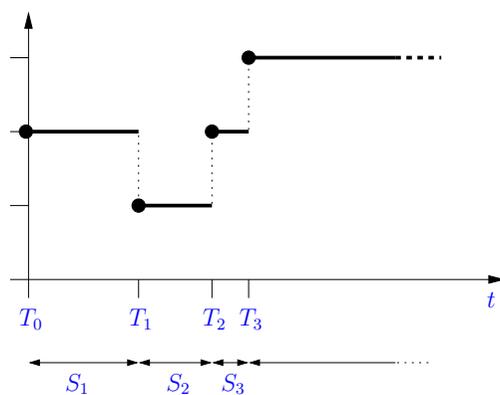
$$N_{t+h} \underset{h \rightarrow 0}{\simeq} \begin{cases} i+1 & \text{avec proba } h \lambda_i \\ i-1 & \text{avec proba } h \mu_i \\ i & \text{avec proba } 1 - h(\lambda_i + \mu_i) \end{cases}$$

avec

- $\lambda_i > 0$ taux de **naissance**
- $\mu_i > 0$ taux de **mort**



temps de sauts/d'attente



- T_k temps de saut
- S_k temps d'attente

temps d'attente

► conditionnellement à $N_{T_k} = i$, pour h petit

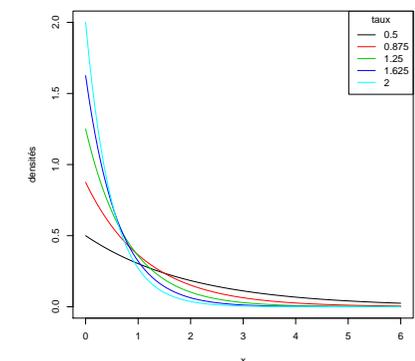
$$N_{T_k+h} = N_{T_k}$$

avec probabilité $1 - h(\lambda_i + \mu_i)$
ainsi

$$S_k \sim \text{Exp}(\lambda_i + \mu_i)$$

et le temps moyen d'attente est $1/(\lambda_i + \mu_i)$

► loi exponentielle (cf. [script5.R](#))



- ▶ à l'instant $T_{k+1} = T_k + S_k$ il y a un saut vers $+1$ (naissance) ou vers -1 (mort)
- ▶ conditionnellement à $N_{T_k} = i$,

$$N_{T_{k+1}} = \begin{cases} i + 1 & \text{avec proba } \lambda_i / (\lambda_i + \mu_i) \\ i - 1 & \text{avec proba } \mu_i / (\lambda_i + \mu_i) \end{cases}$$

simulation exacte : algorithme de Gillespie II

```

T0 ← 0
N0 ← ℓ
for k = 1, 2, ... do
  i ← NTk-1
  r(i) ← λi + μi
  S ~ Exp(r(i))
  Tk ← Tk-1 + S
  if rand() < λi/r(i) then
    NTk ← i + 1 {naissance}
  else
    NTk ← i - 1 {mort}
  end if
end for
    
```

simulation exacte : algorithme de Gillespie I

- ▶ partant de $N_{T_{k-1}} = i$
 - on y reste un temps exponentiel de paramètre $r(i) = \lambda_i + \mu_i$ (temps moyen $\frac{1}{r(i)}$), i.e.

$$S_k \simeq \text{Exp}(r(i)) \quad T_{k+1} = T_k + S_k$$

- puis on saute selon

$$N_{T_{k+1}} = \begin{cases} i + 1 & \text{avec proba } \lambda_i / r(i) \\ i - 1 & \text{avec proba } \mu_i / r(i) \end{cases}$$

plan général

- Introduction
- Premiers pas en R
- Processus de naissance et mort
- Processus de naissance/mort linéaire
- Processus de naissance/mort logistique
- Processus de Markov de saut pur $\rightarrow \mathbb{N}$
- Chemostat
- IBM
- Perspectives
- Bibliographie

définition

- ▶ le taux de naissance et le taux de mort sont linéaire par rapport à la taille de la population :

$$\lambda_i = i \lambda \quad \mu_i = i \mu$$

- ▶ modèle de croissance malthusienne

moyenne/variance des incréments

$$N_{t_{k+1}} - N_{t_k} | N_{t_k} = n = \begin{cases} 1 & \text{avec proba } n h \lambda + o(h) \\ 0 & \text{avec proba } 1 - n h (\lambda + \mu) + o(h) \\ -1 & \text{avec proba } n h \mu + o(h) \end{cases}$$

- ▶ moyenne et variance

$$\mathbb{E}[N_{t_{k+1}} - N_{t_k} | N_{t_k} = n] = 1 [n h \lambda + o(h)] + 0 [1 - n h (\lambda + \mu) + o(h)] - 1 [n h \mu + o(h)] = n h (\lambda - \mu) + o(h)$$

$$\begin{aligned} \text{var}[N_{t_{k+1}} - N_{t_k} | N_{t_k} = n] &= 1 [n h \lambda + o(h)] + 0 [1 - n h (\lambda + \mu) + o(h)] \\ &\quad + 1 [n h \mu + o(h)] + o(h) \\ &= n h (\lambda + \mu) + o(h) \end{aligned}$$

- ▶ donc en moyenne : $\mathbb{E}N_t = \ell e^{(\lambda - \mu)t}$

cript6.R I

```
nm_lineaire <- function(Tmax,X0,lambda,mu) {
  T <- 0; X <- X0; TT <- T; XX <- X # initialisation
  while (T < Tmax) {
    llambda <- lambda*X
    mmu <- mu *X
    r <- llambda+mmu
    S <- rexp(1,rate=r)
    T <- T+S
    X <- X+2*(as.integer(runif(1)<llambda/r)-0.5)
    TT <- c(TT,T)
    XX <- c(XX,X)
    if (X==0) break
  }
  return(list(Tsauts=TT,Xsauts=XX))
}
```

cript6.R II

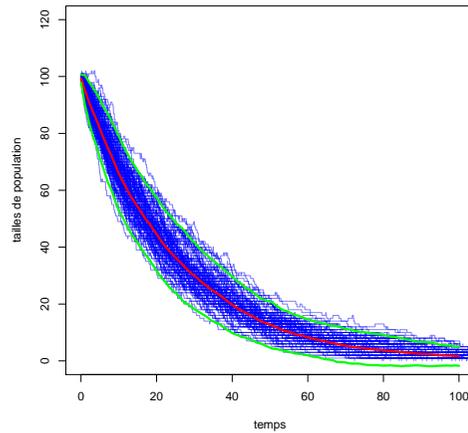
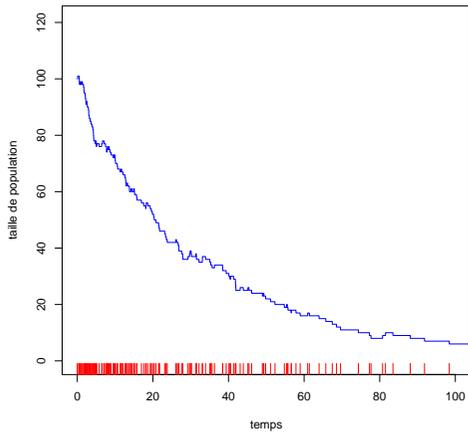
```
Tmax <- 100 # temps final de simulatipon
X0 <- 100 # taille initial de population
lambda <- 0.01 # taux de naissance per capita
mu <- 0.05 # taux de mort per capita

proc <- nm_lineaire(Tmax,X0,lambda,mu)

plot(proc$Tsauts,proc$Xsauts,type="s",col="blue",
      xlim=c(0,Tmax),ylim=c(0,round(1.2*max(proc$Xsauts))),
      xlab="temps",ylab="taille de population")
rug(proc$Tsauts,col="red")
dev.copy2pdf(file="script6.pdf")
```

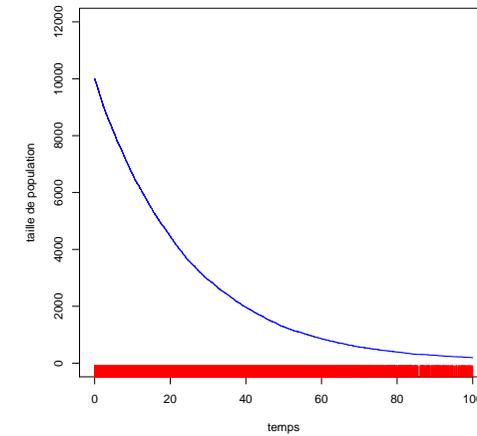
simulation

► $\lambda = 0.01, \mu = 0.05$



(script6.R) moyenne $\ell e^{(\lambda-\mu)t}$ (script7.R)

problème : temps de simulation I



$X_0 < 10000$

problème : temps de simulation II

```
> system.time(proc <- nm_lineaire(Tmax,100,lambda,mu))
utilisateur      système      écoulé
      0.008         0.001         0.009
> length(proc$Tsauts)
[1] 140
> system.time(proc <- nm_lineaire(Tmax,10000,lambda,mu))
utilisateur      système      écoulé
      1.760         0.632         2.359
> length(proc$Tsauts)
[1] 14466
```

► c'est très long pour obtenir la fonction exponentielle!

grande population (N_t grand)

► alors le taux global est grand

$$r(N_t) = (\lambda + \mu) N_t$$

- proportionnellement à la taille N_t de la population les incréments ± 1 sont petit
- de très nombreuses fluctuations de petites amplitudes
- comment accélérer ?
- approximation de N_t par un processus sur une grille $t_k = k h$ avec un pas h convenablement choisi

approximation de Poisson I

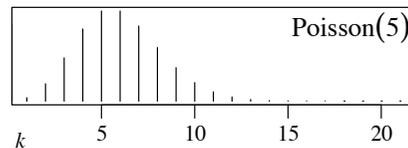
$$N_{t_{k+1}} = N_{t_k} + B - D \quad \begin{cases} B & \text{nb naissances} \\ D & \text{nb morts} \end{cases} \text{ sur } [t_k, t_{k+1}]$$

- ▶ Hyp : h suffisamment petit afin que les taux ne changent pas significativement dans $[t_k, t_{k+1}]$, alors

$$B \simeq \text{Binom}(h \lambda, N_{t_k}) \simeq \text{Poisson}(h \lambda N_{t_k})$$

$$D \simeq \text{Binom}(h \mu, N_{t_k}) \simeq \text{Poisson}(h \mu N_{t_k})$$

- ▶ Rappel : $N \sim \text{Poisson}(r)$ si $\mathbb{P}[N = k] = e^{-r} r^k / k!$ alors $\mathbb{E}[N] = \text{var}[N] = r$



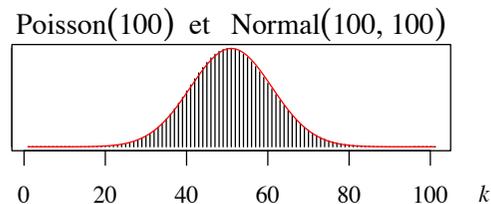
approximation normale I

- ▶ on souhaite quand même que h ne soit pas trop petit (trop lent)
- ▶ Hyp : h suffisamment petit afin que les taux ne changent pas significativement dans $[t_k, t_{k+1}]$ mais suffisamment grand pour qu'il y ait de nombreux événements, alors

$$B \simeq \text{Poisson}(h \lambda N_{t_k}) \simeq \mathcal{N}(h \lambda N_{t_k}, h \lambda N_{t_k})$$

$$D \simeq \text{Poisson}(h \mu N_{t_k}) \simeq \mathcal{N}(h \mu N_{t_k}, h \mu N_{t_k})$$

- ▶ rappel : si r est grand alors $\mathcal{P}(r) \simeq \mathcal{N}(r, r)$:



approximation de Poisson II

$k \leftarrow 0$

$t_k \leftarrow 0$

$\tilde{N}_k \leftarrow \ell$

while $t_k + h \leq T_{\max}$ **do**

$k \leftarrow k + 1$

$t_k \leftarrow t_{k-1} + h$

$B \sim \text{Poisson}(h \lambda \tilde{N}_{t_k})$

$D \sim \text{Poisson}(h \mu \tilde{N}_{t_k})$

$\tilde{N}_{t_k} \leftarrow \tilde{N}_{t_{k-1}} + B - D$

end while

- moyenne :

$$\mathbb{E}(B - D) = \mathbb{E}B - \mathbb{E}D$$

$$= \tilde{N}_{t_k} h (\lambda - \mu)$$

- variance :

$$\text{var}(B - D) = \text{var}B + \text{var}D$$

$$= \tilde{N}_{t_k} h (\lambda + \mu)$$

approximation normale II

- ▶ $B \simeq \mathcal{N}(h \lambda N_{t_k}, h \lambda N_{t_k})$ et $D \simeq \mathcal{N}(h \mu N_{t_k}, h \mu N_{t_k})$ donc

$$B - D \simeq \mathcal{N}(h (\lambda - \mu) N_{t_k}, h (\lambda + \mu) N_{t_k})$$

$$= h (\lambda - \mu) N_{t_k} + \sqrt{h (\lambda + \mu) N_{t_k}} \mathcal{N}(0, 1)$$

- ▶ approximation :

$$\bar{N}_{t_{k+1}} = \bar{N}_{t_k} + h (\lambda - \mu) \bar{N}_{t_k} + \sqrt{h (\lambda + \mu) \bar{N}_{t_k}} \times \mathcal{N}(0, 1)$$

extinction : si $\bar{N}_{t_{k+1}} < 0$ alors $\bar{N}_{t_{k+1}} = 0$

- ▶ **approximation diffusion** : N_{t_k} est une approximation en temps discret de l'équation différentielle stochastique suivante

$$dX_t = (\lambda - \mu) X_t dt + \sqrt{(\lambda + \mu) X_t} dB_t$$

où B_t est un **mouvement brownien**

algorithmme

```
k ← 0
tk ← 0
N̄k ← ℓ
while tk + h ≤ tmax do
  k ← k + 1
  tk ← tk-1 + h
  ξ ∼ N(0,1)
  N̄tk ← N̄tk-1 + h(λ - μ)N̄tk + √h(λ + μ)N̄tk × ξ
  if N̄tk < 0 then
    N̄tk ← 0 {extinction}
  end if
end while
```

un peu de R I

► script8.R

```
tmax <- 30
x0 <- 1000
lambda <- 10
mu <- 10-0.1
N <- 400 # nb intervalles discrétisa
h <- tmax/N # pas de temps
temps <- seq(0,tmax,length=N+1) # liste des instants

malthus <- fonction(temps_t,x0,lambda,mu) {
  x0 * exp( (lambda - mu) * temps_t )
}

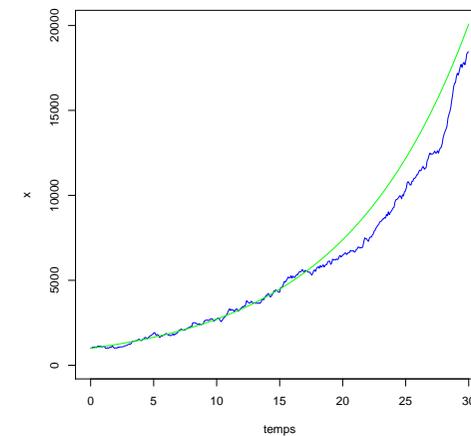
mmalthus <- malthus(temps,x0,lambda,mu)
```

un peu de R II

```
set.seed(222)
x <- array(dim=N+1,x0)
ggauss <- array(dim=N+1,rnorm(N+1))
for (n in 2:(N+1)) {
  xx <- x[n-1]
  x[n] <- pmax(xx + xx*h*(lambda-mu)
              + sqrt(xx*h*(lambda+mu)) * ggauss[n],0)
}

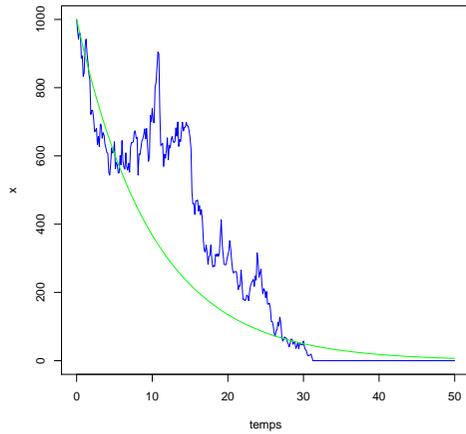
plot(temps,x,type="l",col="blue",ylim=c(0,max(mmalthus)))
lines(temps,mmalthus,type="l",col="green")
dev.copy2pdf(file="script8.pdf")
```

un peu de R III

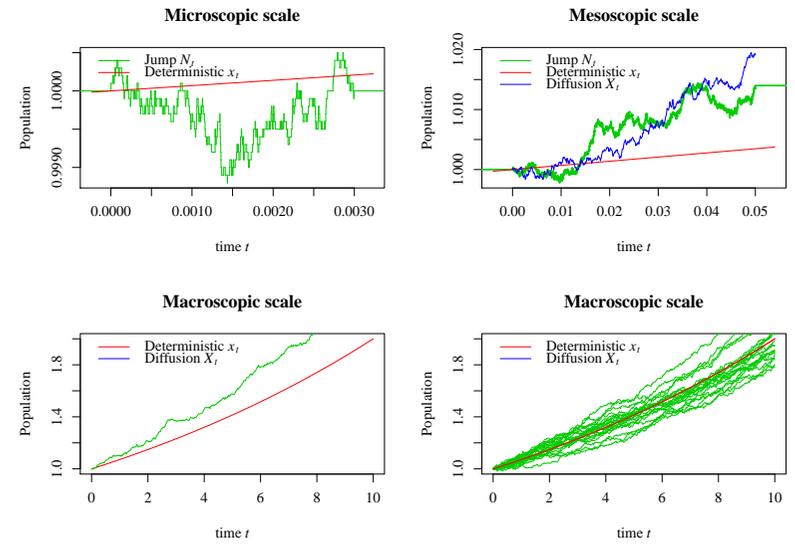


un peu de R IV

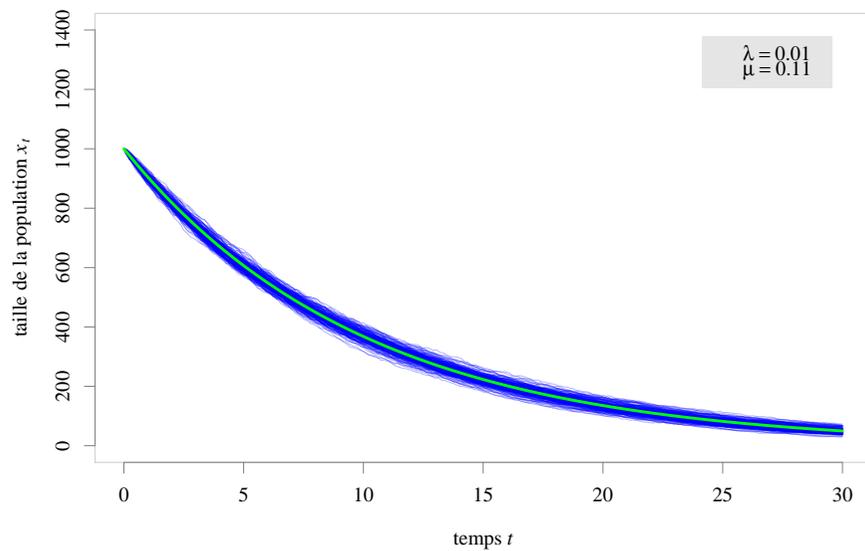
► `tmax <- 50; lambda <- 10; mu <- 10+01`



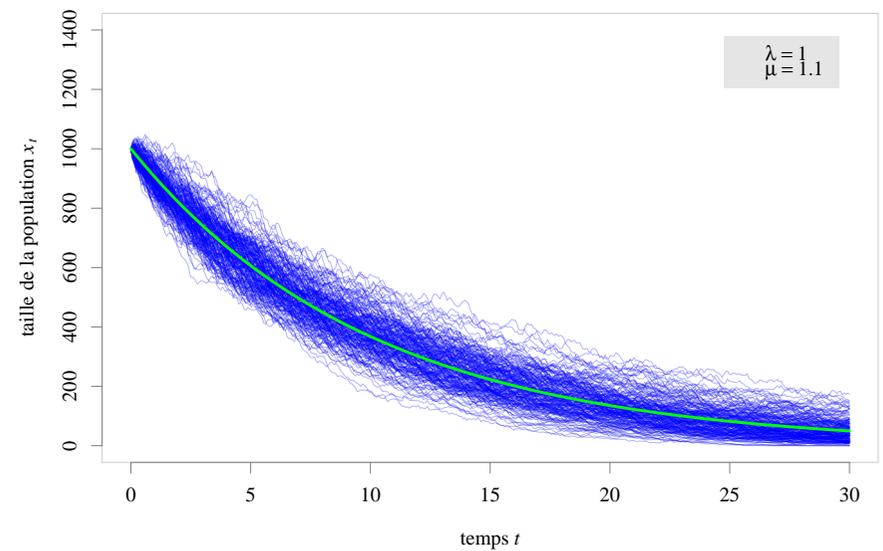
une question d'échelle

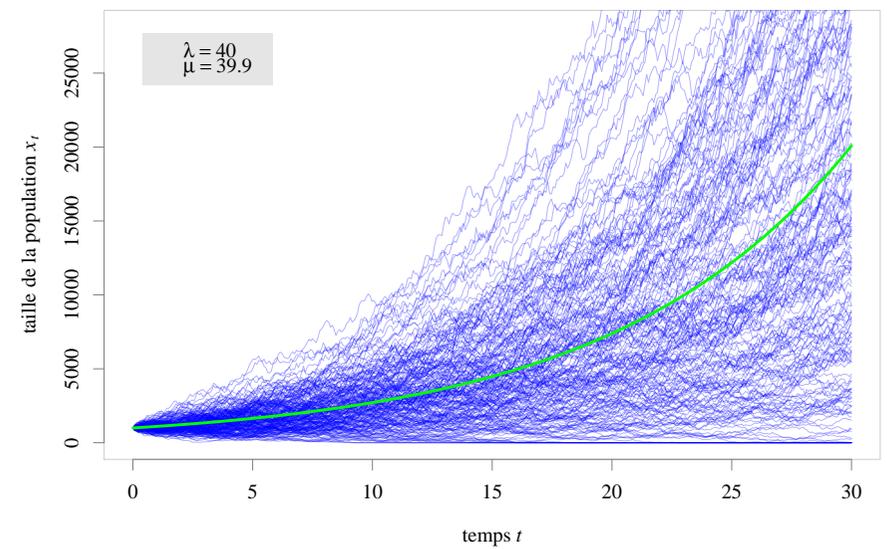
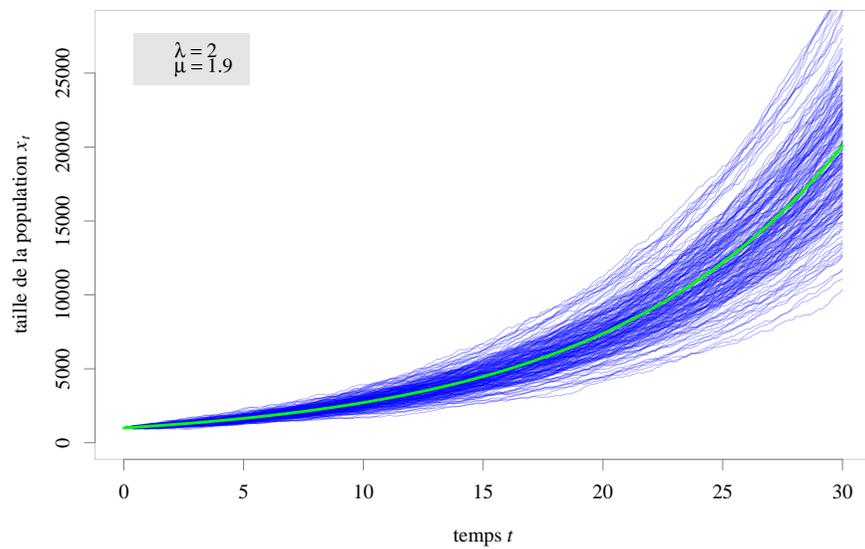
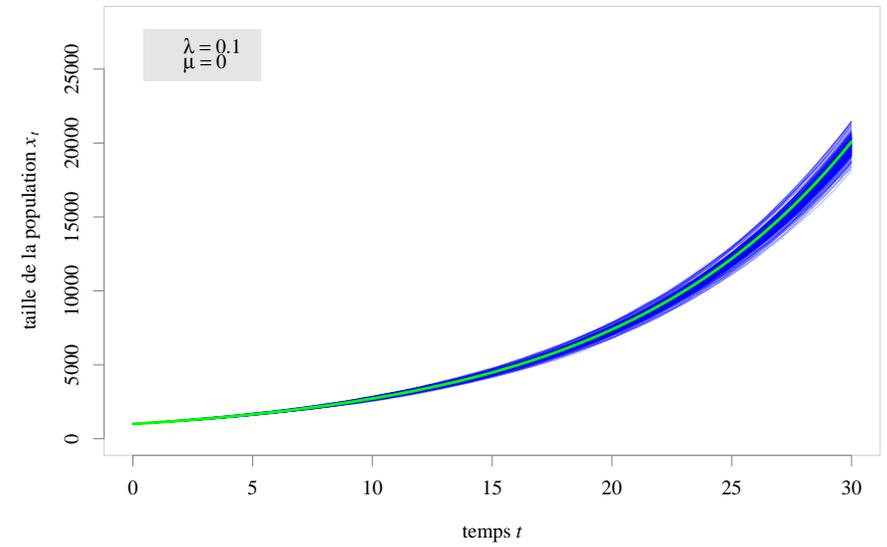
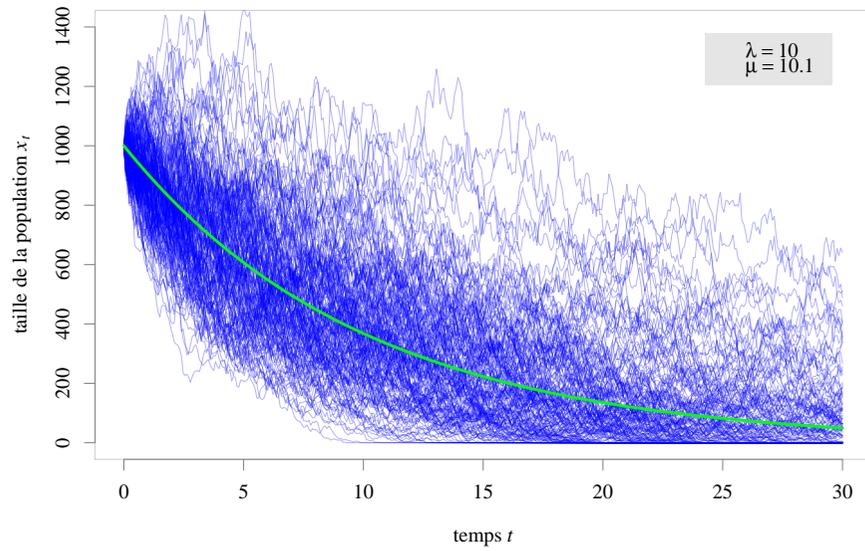


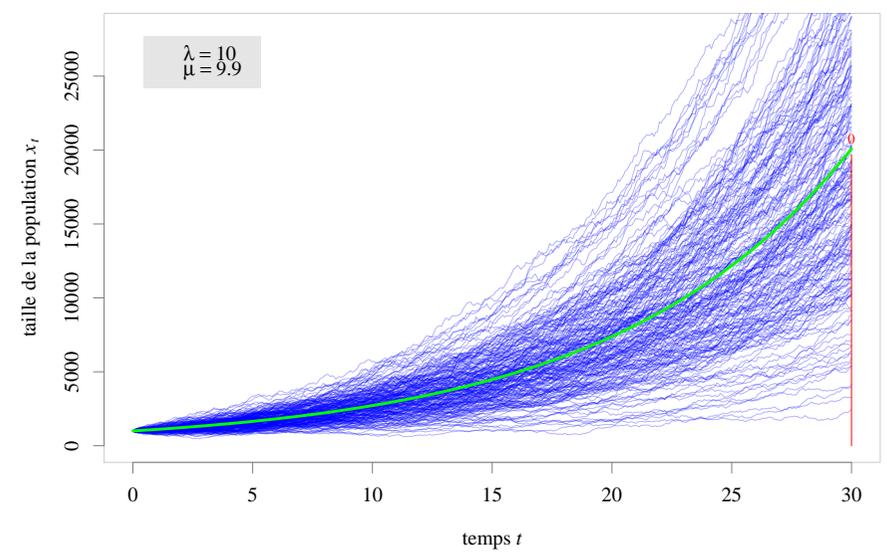
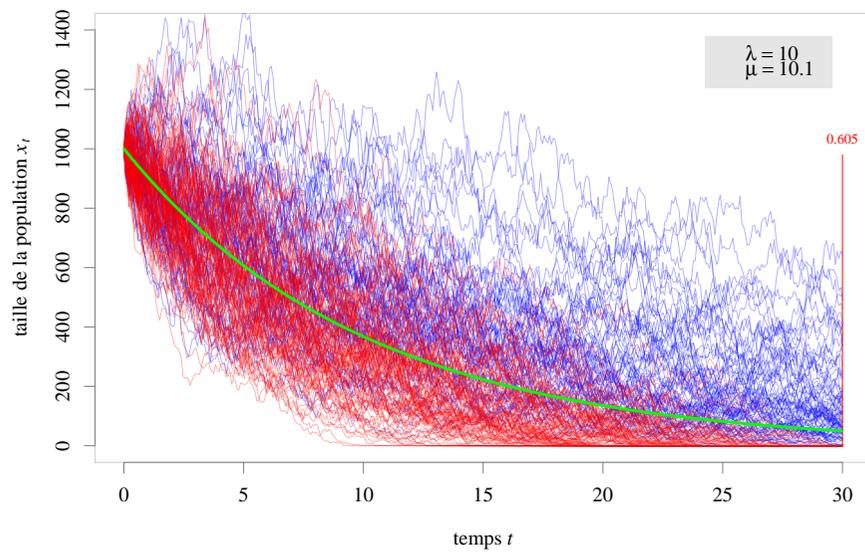
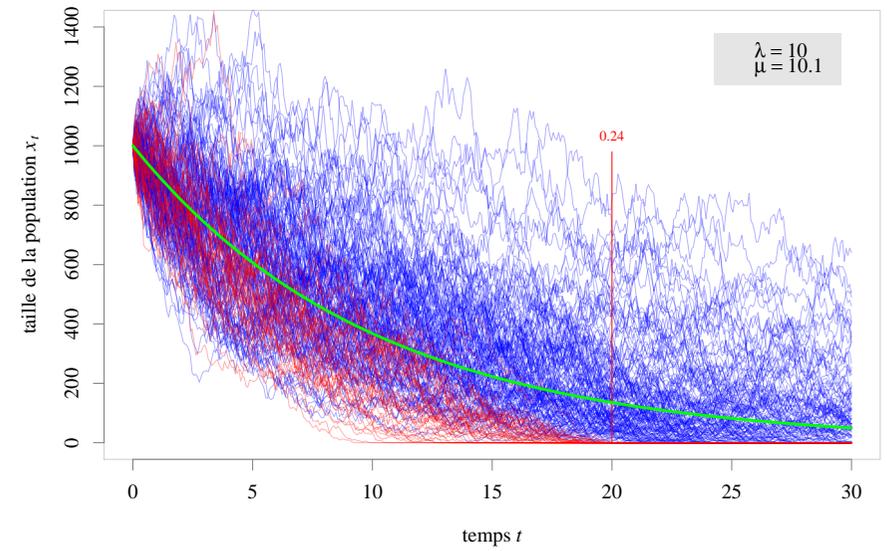
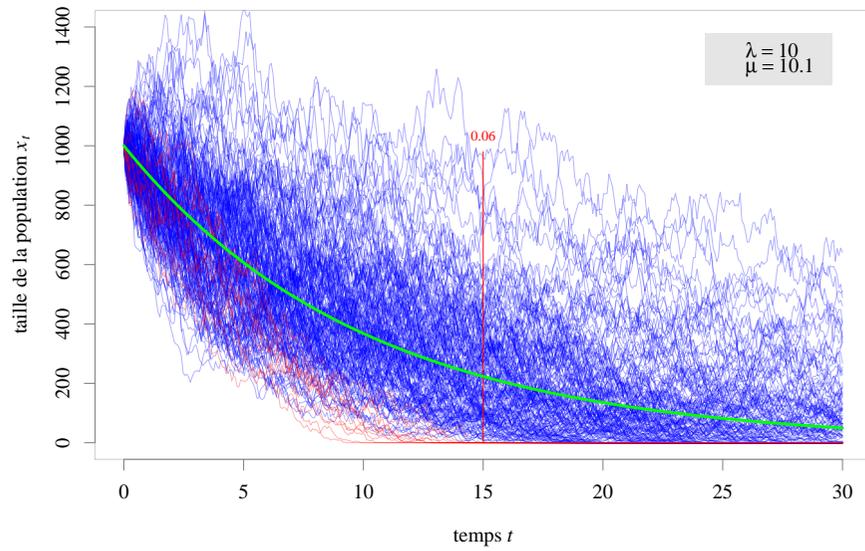
script9.R

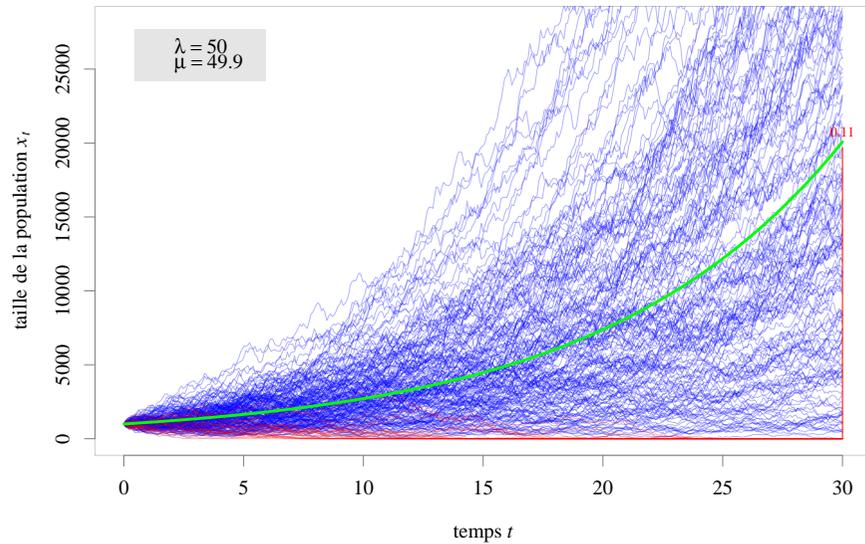


script9.R



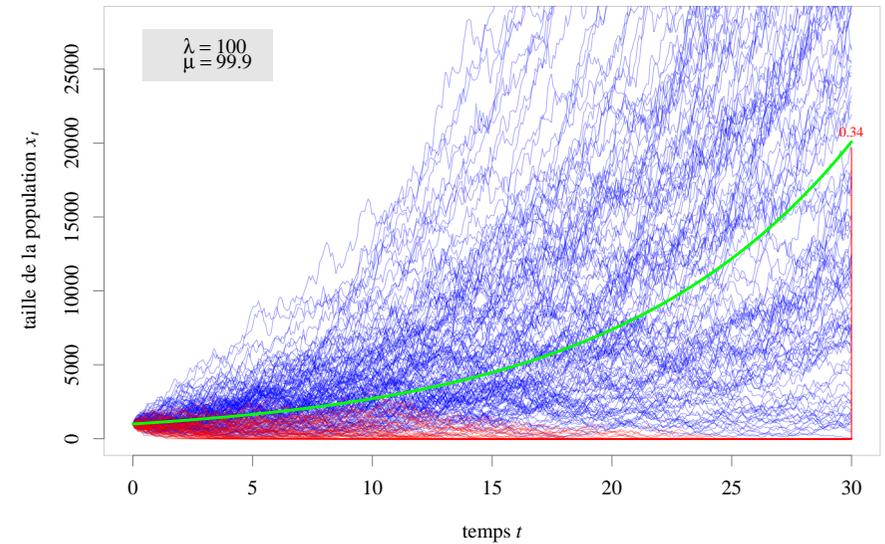






extinction

- ▶ modèle EDO : la population ne s'éteint jamais
- ▶ modèle stochastique : la population s'éteint en temps fini
 - temps d'extinction $\tau = \inf(t; N_t = 0)$, sa loi ?
 - loi stationnaire : δ_0
 - loi quasi-stationnaire : loi stationnaire conditionnellement à la non-extinction



plan général

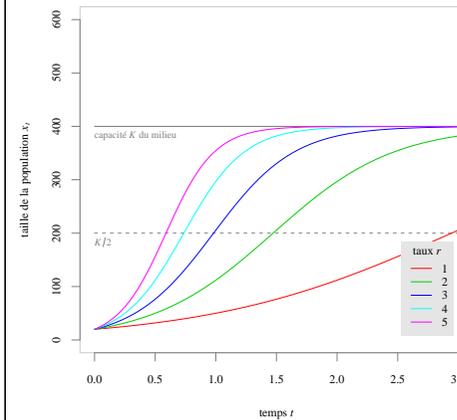
- Introduction
- Premiers pas en R
- Processus de naissance et mort
- Processus de naissance/mort linéaire
- Processus de naissance/mort logistique
- Processus de Markov de saut pur $\rightarrow \mathbb{N}$
- Chemostat
- IBM
- Perspectives
- Bibliographie

logistique (EDO) I

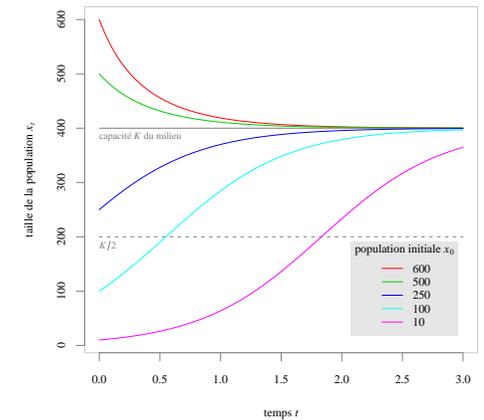
$$\frac{dx_t}{dt} = r \times \left(1 - \frac{x_t}{K}\right) \times x_t \quad (\text{densité})$$

$$\frac{dn_t}{dt} = r \times \left(1 - \frac{n_t/V}{K}\right) \times n_t \quad \text{avec } x_t = \frac{n_t}{V} \quad (\text{taille})$$

- ▶ K capacité du milieu, r taux per capita
- ▶ EDO non-linéaire
- ▶ solution explicite $x_t = \frac{KP_0e^{rt}}{K + P_0(e^{rt} - 1)}$



logistique (EDO) II



logistique stochastique

- ▶ avec h petit et conditionnellement à $N_t = i$

$$N_{t+h} = \begin{cases} i+1 & \text{avec proba } \lambda_i \\ i-1 & \text{avec proba } \mu_i \\ i & \text{avec proba } 1 - (\lambda_i + \mu_i) \end{cases}$$

avec

- $\lambda_i > 0$ taux de **naissance**
- $\mu_i > 0$ taux de **mort**

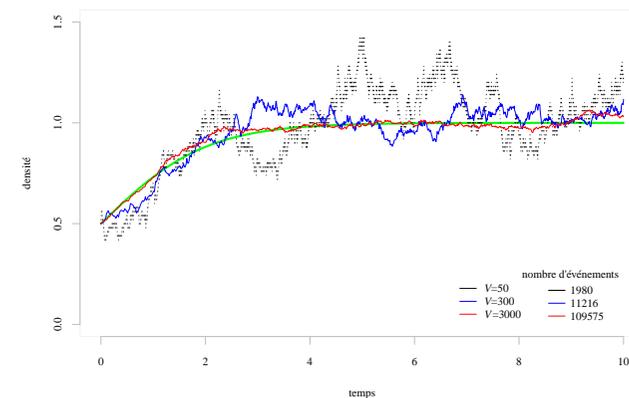
- ▶ **logistique**

$$\lambda_i - \mu_i = r i \left(1 - \frac{i/V}{K}\right) \quad \text{e.g. } \begin{cases} \lambda_i = \lambda i \\ \mu_i = (\mu + di) i \end{cases}$$

$r = \lambda - \mu$ taux de croissance; $KV = (\lambda - \mu)/d$, K capacité du milieu, V paramètre de renormalisation (volume, surface, taille de population...) $\rightarrow \infty$

limite fluide $V \rightarrow \infty$

$$\frac{1}{V} N_t \xrightarrow{V \rightarrow \infty} x_t \quad (\text{densité})$$



- ▶ pourquoi ?

approximation de Poisson I

$$N_{t_{k+1}} = N_{t_k} + B - D \quad \begin{cases} B & \text{nb naissances} \\ D & \text{nb morts} \end{cases} \text{ sur } [t_k, t_{k+1}]$$

- ▶ Hyp : h suffisamment petit afin que les taux ne changent pas significativement dans $[t_k, t_{k+1}]$, alors

$$B \simeq \text{Poisson}(h \lambda N_{t_k})$$

$$D \simeq \text{Poisson}(h (\mu + d N_{t_k}) N_{t_k})$$

approximation normale I

- ▶ on souhaite quand même que h ne soit pas trop petit (trop lent)
- ▶ Hyp : h suffisamment petit afin que les taux ne changent pas significativement dans $[t_k, t_{k+1}]$ mais suffisamment grand pour qu'il y ait de nombreux événements, alors

$$B \simeq \text{Poisson}(h \lambda N_{t_k}) \simeq \mathcal{N}(h \lambda N_{t_k}, h \lambda N_{t_k})$$

$$D \simeq \text{Poisson}(h (\mu + d N_{t_k}) N_{t_k})$$

$$\simeq \mathcal{N}(h (\mu + d N_{t_k}) N_{t_k}, h (\mu + d N_{t_k}) N_{t_k})$$

ainsi $B - D$ est approximativement

$$h r N_{t_k} \left(1 - \frac{N_{t_k}/V}{K}\right) + \sqrt{h \left(\lambda + \mu + r \frac{N_{t_k}/V}{K}\right) N_{t_k}} \times \mathcal{N}(0, 1)$$

approximation de Poisson II

$$k \leftarrow 0$$

$$t_k \leftarrow 0$$

$$\tilde{N}_k \leftarrow \ell$$

while $t_k + h \leq t_{\max}$ **do**

$$k \leftarrow k + 1$$

$$t_k \leftarrow t_{k-1} + h$$

$$B \sim \text{Poisson}(h \lambda \tilde{N}_{t_k})$$

$$D \sim \text{Poisson}(h (\mu + d N_{t_k}) \tilde{N}_{t_k})$$

$$\tilde{N}_{t_k} \leftarrow \tilde{N}_{t_{k-1}} + B - D$$

end while

approximation normale II

- ▶ approximation diffusion de N_t

$$\begin{aligned} \bar{N}_{t_{k+1}} &= \bar{N}_{t_k} + h r \bar{N}_{t_k} \left(1 - \frac{\bar{N}_{t_k}/V}{K}\right) \\ &\quad + \sqrt{h \left(\lambda + \mu + r \frac{\bar{N}_{t_k}/V}{K}\right) \bar{N}_{t_k}} \times \mathcal{N}(0, 1) \end{aligned}$$

extinction : si $\bar{N}_{t_{k+1}} < 0$ alors $\bar{N}_{t_{k+1}} = 0$

- ▶ renormalisation $\bar{X}_{t_k} \stackrel{\text{déf}}{=} \bar{N}_{t_k}/V$

$$\begin{aligned} \frac{\bar{N}_{t_{k+1}}}{V} &= \frac{\bar{N}_{t_k}}{V} + h r \frac{\bar{N}_{t_k}}{V} \left(1 - \frac{\bar{N}_{t_k}/V}{K}\right) \\ &\quad + \frac{1}{\sqrt{V}} \sqrt{h \left(\lambda + \mu + r \frac{\bar{N}_{t_k}/V}{K}\right) \bar{N}_{t_k}} \times \mathcal{N}(0, 1) \end{aligned}$$

► après renormalisation

$$\bar{X}_{t_{k+1}} = \bar{X}_{t_k} + h r \bar{X}_{t_k} \left(1 - \frac{\bar{X}_{t_k}}{K}\right) + \frac{1}{\sqrt{V}} \sqrt{h \left(\lambda + \mu + r \frac{\bar{X}_{t_k}}{K}\right)} \bar{X}_{t_k} \times \mathcal{N}(0, 1)$$

► \bar{X}_{t_k} est une approximation en temps discret de l'équation différentielle stochastique suivante

$$dX_t = r X_t \left(1 - \frac{X_t}{K}\right) dt + \frac{1}{\sqrt{V}} \sqrt{\left(\lambda + \mu + r \frac{X_t}{K}\right)} X_t dB_t$$

où B_t est un mouvement brownien

```

k ← 0
t_k ← 0
X̄_k ← ℓ
while t_k + h ≤ T_max do
  k ← k + 1
  t_k ← t_{k-1} + h
  ξ ~ N(0, 1)
  derive ← h r X̄_{t_k} (1 - X̄_{t_k} / K)
  diffusion ← 1/√V h (λ + μ + r X̄_{t_k} / K) X̄_{t_k}
  X̄_{t_k} ← X̄_{t_{k-1}} + derive + √diffusion × ξ
  if X̄_{t_k} < 0 then
    X̄_{t_k} ← 0 {extinction}
  end if
end while
    
```

fonction/graphiques (script11.R) I

```

tmax <- 3; npts <- 2000; h <- tmax/npts
temps <- seq(0,tmax,length=npts+1)
lambda <- 10
mu <- 7
x0 <- 1 # population initiale
K <- 2 # capacité
V <- 1000 # volume (parametre de renormalisation)
r <- lambda - mu
d <- r / K

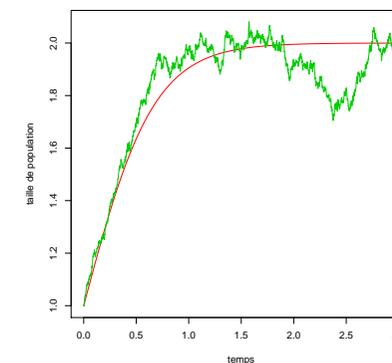
logistique<- function(x) {
  K / (1 + (K/x0-1) * exp(-r * x))
}

xt <- numeric(npts)
xt[1] <- x0
for (i in (2:(npts+1))) {
  derive <- h * r * (1 - xt[i-1] / K) * xt[i-1]
  diffusion <- h * (lambda + mu + d * xt[i-1]) * xt[i-1]/V
  xt[i] <- xt[i-1] + derive + sqrt(diffusion) * rnorm(1)
  if (xt[i] < 0) xt[i] = 0 # test d'extinction
}
    
```

fonction/graphiques (script11.R) II

```

matplot((0:npts)*h,cbind(logistique((0:npts)*h),xt),
  type="l",lty=1,col=c(2,3),
  xlab="temps",ylab="taille de population")
dev.copy2pdf(file="script11.pdf")
    
```



- ▶ lorsque $V \rightarrow \infty$ (i.e. grande population, densité grande etc.) X_t converge vers la solution de l'EDO :

$$\frac{d}{dt}x_t = r x_t \left(1 - \frac{x_t}{K}\right) \quad (\text{logistique})$$

\bar{X}_{t_k} converge vers l'approximation par un schéma d'Euler de l'EDO logistique

- ▶ $\mathbb{E}X_t < x_t$ (dans le cas linéaire $\mathbb{E}X_t = x_t$)

- Introduction
- Premiers pas en R
- Processus de naissance et mort
- Processus de naissance/mort linéaire
- Processus de naissance/mort logistique
- Processus de Markov de saut pur $\rightarrow \mathbb{N}$
- Chemostat
- IBM
- Perspectives
- Bibliographie

Markov

- ▶ $(X_t)_{t \geq 0}$ processus en temps continu à valeurs dans \mathbb{N} vérifie la propriété de Markov lorsque

$$\mathbb{P}(X_{t_{n+1}} = j | X_{t_n} = i_n, \dots, X_{t_0} = i_0) = \mathbb{P}(X_{t_{n+1}} = j | X_{t_n} = i_n)$$

pour tout $n, i_k \in \mathbb{N}, 0 \leq t_0 \leq \dots \leq t_{n+1}$

loi(futur|passé) = loi(futur|présent)

- ▶ la loi de $(X_t)_{t \geq 0}$ est donc définie par :

- loi initiale $\nu = \text{loi}(X_0)$
- probabilités de transition $P_{ij}(t) = \mathbb{P}(X_{t+s} = j | X_s = i)$
(hyp : X_t homogène en temps, i.e. P ne dépend pas de s)

en effet

$$\mathbb{P}(X_{t_n} = i_n, \dots, X_{t_0} = i_0) = \nu_{i_0} P_{i_0 i_1}(t_1 - t_0) \cdots P_{i_{n-1} i_n}(t_n - t_{n-1})$$

notation matricielle

$$\mathbb{E}_\nu f(X_t) = \nu P(t) f$$

$$= \underbrace{[\mathbb{P}(X_t=0) \ \mathbb{P}(X_t=1) \ \mathbb{P}(X_t=2) \ \cdots]}_{\nu P(t) = \pi(t) = \text{loi}(X_t)} \times \underbrace{[P(t)]_{ij \in \mathbb{N}}}_{P(t) f = \mathbb{E}(f(X_t) | X_0)} \times \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \end{bmatrix}$$

- ▶ si $X_0 \equiv \ell$, i.e. $\nu = \delta_\ell = [0 \cdots 0 1 0 \cdots]$ (1 à la place ℓ)

$$\mathbb{E}_\ell f(X_t) = [0 \cdots 0 1 0 \cdots] \times [P(t)]_{ij \in \mathbb{N}} \times \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \end{bmatrix}$$

propriété de semi-groupe

$$P(t+s) = P(t)P(s) \quad P(0) = I$$

- ▶ c'est à dire

$$\mathbb{P}(X_{t+s} = j | X_0 = i) = \sum_{k \in \mathbb{N}} \mathbb{P}(X_s = k | X_0 = i) \mathbb{P}(X_{t+s} = j | X_s = k)$$

- ▶ générateur infinitésimal

$$A = \lim_{t \downarrow 0} \frac{P(t) - I}{t} \quad \begin{cases} A_{ij} \geq 0 \text{ pour } i \neq j \\ A_{ii} \leq 0 \\ \sum_j A_{ij} = 0 \end{cases}$$

- ▶ posons

$$r(i) = -A_{ii} = \sum_{j \neq i} A_{ij}$$

équation Fokker-Planck (master equation)

- ▶ $\frac{d}{dt} \nu P(t) = \nu P(t) A$ mais $\pi(t) = \text{loi}(X_t) = \nu P(t)$ donc

$$\dot{\pi}(t) = \pi(t) A, \quad \pi(0) = \nu$$

$$\text{ou } \dot{\pi}_j(t) = \sum_{i \in \mathbb{N}} [\pi_i(t) A_{ij} - \pi_j(t) A_{ji}]$$

- ▶ $\tilde{\pi}$ est dite stationnaire lorsque $\tilde{\pi} A = 0$

- ▶ $\frac{d}{dt} \nu P(t) f = \nu P(t) A f$ mais $\mathbb{E}_\nu f(X_t) = \nu P(t) f$ donc

$$\frac{d}{dt} \mathbb{E}[f(X_t)] = \mathbb{E}[A f(X_t)], \quad \mathbb{E}f(X_0) = \nu f$$

équation de Kolmogorov progressive

- ▶ d'après la propriété de semi-groupe

$$\frac{P(t+h) - P(t)}{h} = P(t) \frac{P(h) - I}{h} \xrightarrow{h \downarrow 0} P_t A$$

- ▶ équation de Kolmogorov progressive

$$\dot{P}(t) = P(t) A, \quad P(0) = I$$

représentation de X_t

$$\mathbb{E}f(X_t) = \mathbb{E}f(X_0) + \int_0^t \mathbb{E}[A f(X_s)] ds \quad (*)$$

- ▶ d'après (*)

$$M_t^f \stackrel{\text{déf}}{=} f(X_t) - f(X_0) - \int_0^t A f(X_s) ds$$

est une martingale de moyenne nulle : c'est à dire un "bruit"

- ▶ représentation de $f(X_t)$

$$f(X_t) = f(X_0) + \int_0^t A f(X_s) ds + M_t^f$$

- ▶ représentation de X_t , on pose $\psi(x) = x$, alors

$$X_t = X_0 + \int_0^t A \psi(X_s) ds + M_t^\psi$$

exemple : processus de naissance/mort

$$\mathbb{P}(N_{t+h} = j | N_t = i) \stackrel{h \downarrow 0}{\simeq} \begin{cases} \lambda_i h & \text{si } j = i + 1 \\ \mu_i h & \text{si } j = i - 1 \\ 1 - (\lambda_i + \mu_i) h & \text{si } j = i \end{cases}$$

$$A = \begin{pmatrix} -\lambda_0 & \lambda_0 & & & & \\ \mu_1 & -\mu_1 - \lambda_1 & \lambda_1 & & & \\ & \mu_2 & -\mu_2 - \lambda_2 & \lambda_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & & \ddots \end{pmatrix}$$

$$\begin{aligned} Af(i) &= \lambda_i [f(i+1) - f(i)] + \mu_i [f(i-1) - f(i)] \\ &= r(i) \left\{ \frac{\lambda_i}{r(i)} [f(i+1) - f(i)] + \frac{\mu_i}{r(i)} [f(i-1) - f(i)] \right\} \end{aligned}$$

exemple : logistique

- ▶ $\lambda_i = i \lambda$, $\mu_i = (\mu + d i) i$, $r(i) = i(\lambda + \mu) + d i^2$
- ▶ générateur

$$\begin{aligned} Af(i) &= \lambda i (f(i+1) - f(i)) + (\mu + d i) i (f(i-1) - f(i)) \\ &= r(i) \left[\frac{\lambda i}{r(i)} (f(i+1) - f(i)) + \frac{(\mu + d i) i}{r(i)} (f(i-1) - f(i)) \right] \end{aligned}$$

avec taux global $r(i) = i(\lambda + \mu + d i)$.

- ▶ équation

$$N_t = N_0 + \int_0^t r N_s \left(1 - \frac{N_s}{K}\right) ds + M_t$$

exemple : cas linéaire

- ▶ $\lambda_i = i \lambda$, $\mu_i = i \mu$, $r(i) = i(\lambda + \mu)$ donc

$$Af(i) = i(\lambda + \mu) \left\{ \frac{\lambda}{\lambda + \mu} [f(i+1) - f(i)] + \frac{\mu}{\lambda + \mu} [f(i-1) - f(i)] \right\}$$

$$A\psi(i) = i(\lambda + \mu) \left\{ \frac{\lambda}{\lambda + \mu} [i+1 - i] + \frac{\mu}{\lambda + \mu} [i-1 - i] \right\} = i(\lambda - \mu)$$

donc

$$N_t = N_0 + \int_0^t (\lambda - \mu) N_s ds + M_t^\psi$$

en prenant l'espérance

$$\mathbb{E}N_t = \mathbb{E}N_0 + \int_0^t (\lambda - \mu) \mathbb{E}N_s ds$$

solution de l'EDO $\dot{x}_t = (\lambda - \mu) x_t$ (croissance malthusienne)

plan général

Introduction

Premiers pas en R

Processus de naissance et mort

Processus de naissance/mort linéaire

Processus de naissance/mort logistique

Processus de Markov de saut pur $\rightarrow \mathbb{N}$

Chemostat

IBM

Perspectives

Bibliographie

$$\begin{aligned}\dot{b}_t &= \mu(b_t, s_t) b_t - D b_t \\ \dot{s}_t &= -c_1 \mu(b_t, s_t) b_t - D (s_t - s_{\text{in}})\end{aligned}$$

- somme de 3 vecteurs

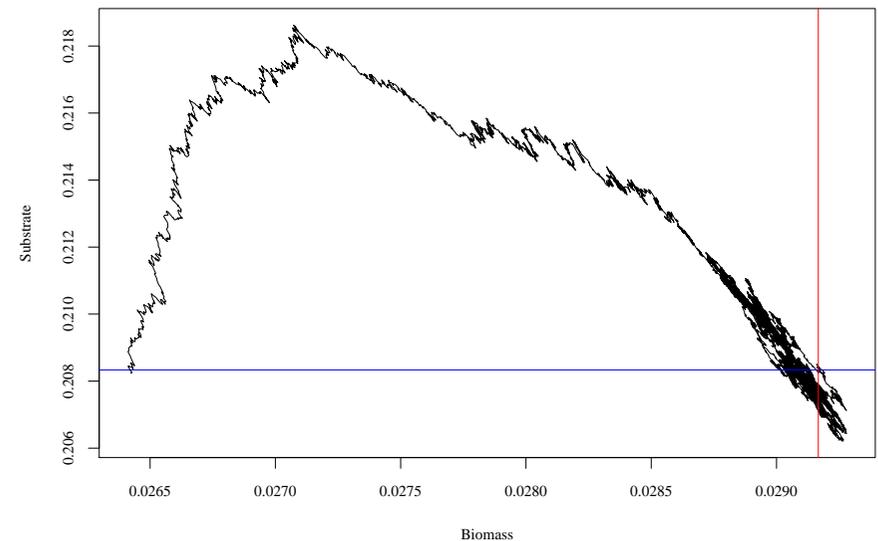
$$\begin{pmatrix} \dot{b}_t \\ \dot{s}_t \end{pmatrix} = \mu(b_t, s_t) b_t \begin{pmatrix} 1 \\ -c_1 \end{pmatrix} + D \begin{pmatrix} 0 \\ s_{\text{in}} \end{pmatrix} + D \begin{pmatrix} -b_t \\ -s_t \end{pmatrix}$$

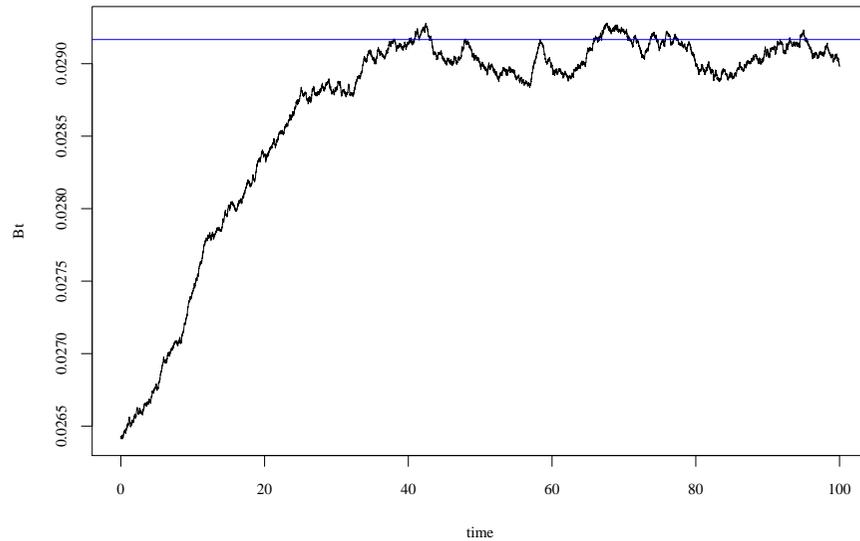
correspondant au 3 différentes sources de variation : bio, in et out

- état : $X_t = (B_t, S_t) \in \mathbb{R}_+^2$
- transitions depuis (b, s) :
- Bio : saut de taille $\frac{1}{K_1}(1, -c_1)$ à un taux $K_1 \mu(b, s) b$
 - In : saut de taille $\frac{1}{K_2}(0, s_{\text{in}})$ à un taux $K_2 D$
 - Out : saut de taille $\frac{1}{K_3}(-b, -s)$ à un taux $K_3 D$

approximation diffusion

$$\begin{aligned}\begin{pmatrix} B_{t_{k+1}} \\ S_{t_{k+1}} \end{pmatrix} &= \begin{pmatrix} B_{t_k} \\ S_{t_k} \end{pmatrix} \\ &+ \left(\mu(B_{t_k}, S_{t_k}) B_{t_k} \begin{pmatrix} 1 \\ -c_1 \end{pmatrix} + D \begin{pmatrix} 0 \\ s_{\text{in}} \end{pmatrix} + D \begin{pmatrix} -B_{t_k} \\ -S_{t_k} \end{pmatrix} \right) h \\ &+ \sqrt{h \frac{\mu(B_{t_k}, S_{t_k}) B_{t_k}}{K_1}} \begin{pmatrix} 1 \\ -c_1 \end{pmatrix} \mathcal{N}(0, 1) \\ &+ \sqrt{h \frac{D}{K_2}} \begin{pmatrix} 0 \\ s_{\text{in}} \end{pmatrix} \mathcal{N}(0, 1) \\ &+ \sqrt{h \frac{D}{K_3}} \begin{pmatrix} -B_{t_k} \\ -S_{t_k} \end{pmatrix} \mathcal{N}(0, 1)\end{aligned}$$





plan général

Introduction

Premiers pas en R

Processus de naissance et mort

Processus de naissance/mort linéaire

Processus de naissance/mort logistique

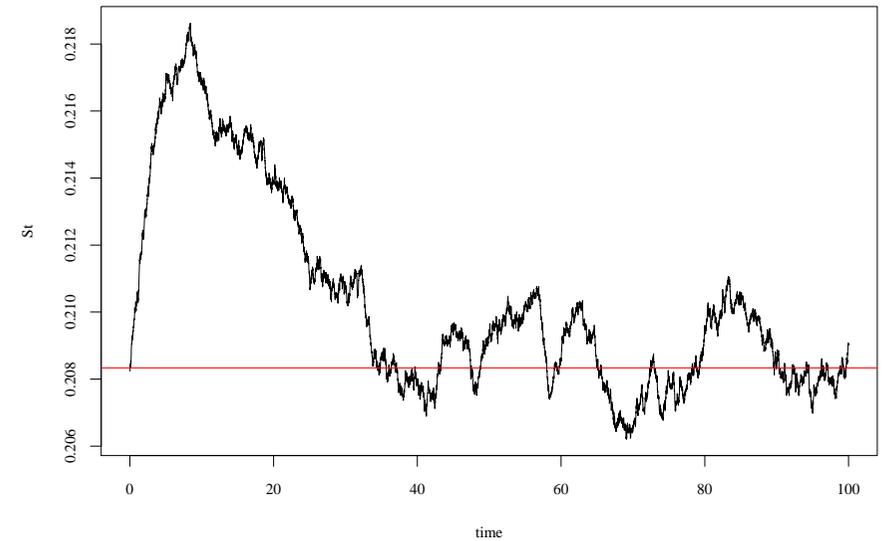
Processus de Markov de saut pur $\rightarrow \mathbb{N}$

Chemostat

IBM

Perspectives

Bibliographie

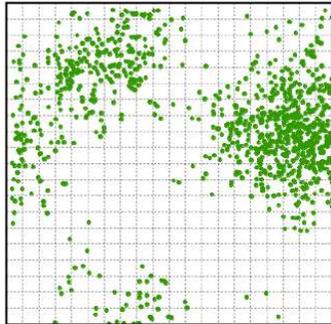


IBM

$$t \rightarrow \nu_t(dx) = \sum_{i=1}^{N_t} \delta_{x_t^i}(dx) \equiv \{x_t^1, x_t^2, \dots, x_t^{N_t}\}$$

- ▶ N_t individus i positionnés en x_t^i
- ▶ on décrit explicitement tous les mécanismes agissant sur chaque individu i (naissance/mort) et modifiant son état x_t^i
- ▶ pourquoi mesures? $\nu_t(A)$ = taille de la population dans A ;
 $\int \nu_t(dx) = N_t$ taille de la population

- ▶ chaque individu x dans la population ν_t
 - se reproduit



$$\nu_t \rightarrow \nu_t + \delta_y \quad (\text{naissance})$$

selon un taux λ^b avec $y \sim D(y - x^i) dy$ noyau de dispersion

- disparaît

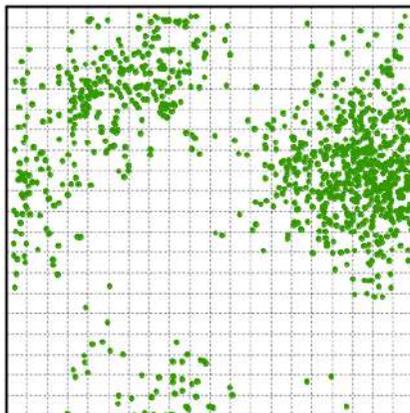
$$\nu_t \rightarrow \nu_t - \delta_{x^i} \quad (\text{mort})$$

selon un taux $\lambda^d(x, \nu_t) = \alpha + \sum_{y \in \nu_t} U(y - x)$

U noyau de compétition
 α taux de mort intrinsèque

- ▶ $(\nu_t)_{t \geq 0}$ est un processus de Markov (à valeurs mesures)

simulation



- ▶ on a un générateur / simulateur (exact) \mathcal{A}
- ▶ représentation

$$\langle \nu_t, \varphi \rangle = \langle \nu_t, \varphi \rangle + \int_0^t \mathcal{A}\varphi(\nu_s) ds + M_t^\varphi$$

M_t^φ martingale dont on connaît l'épaisseur $\langle M^\varphi \rangle_t$ ($\forall \varphi$)

- ▶ **limite fluide** : en population initiale grande $K \uparrow \infty$, si $\frac{U}{K} \rightarrow \bar{U}$ alors $\frac{1}{K} \nu_t$ converge vers la solution faible de

$$\partial_t p(t, x) = \lambda^b \int p(t, y) D(y - x) dy - \left[\alpha + \int p(t, y) \bar{U}(y - x) dy \right] p(t, x)$$

remarques

- ▶ **méthodes des moments** : calculer la taille moyenne de la population

$$n(t) \stackrel{\text{déf}}{=} \mathbb{E} \langle \nu_t, 1 \rangle$$

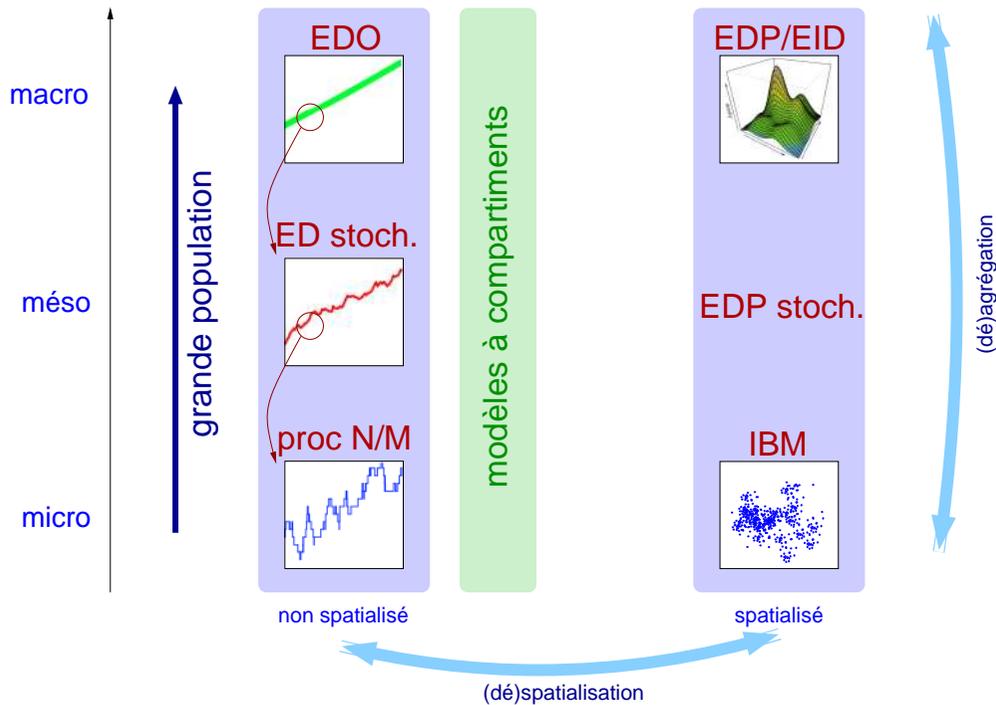
- une logistique faisant intervenir une covariance spatiale
- faisant intervenir un terme d'ordre 3 (que l'on néglige)

- ▶ il est possible de faire des **statistiques** à l'aide des modèles agrégés

- ▶ **simulateurs** : approximations poissonniennes / diffusions / EDO

Gillespie like algorithm

- ▶ $N \leftarrow N_{T_{k-1}}$
- ▶ compute **global rate** $\lambda = \sum_{i=1}^{N_{T_{k-1}}} \lambda(x_i)$
- ▶ next event time $T_K = T_{k-1} + S$ where $S \sim \text{Exp}(\lambda)$
- ▶ choose x with probability $\frac{\lambda(x)}{\lambda}$ then
 - remove x with probability $\frac{\lambda^d(x) + \lambda^c(x)}{\lambda(x)}$
 - add x' with probability $\frac{\lambda^b(x)}{\lambda(x)}$ where x' drawn from $D(x, dz)$



plan général

- Introduction
- Premiers pas en R
- Processus de naissance et mort
- Processus de naissance/mort linéaire
- Processus de naissance/mort logistique
- Processus de Markov de saut pur $\rightarrow \mathbb{N}$
- Chemostat
- IBM
- Perspectives
- Bibliographie

division cellulaire

- ▶ hypothèse simplificatrice : absence de mémoire / pas de vieillissement

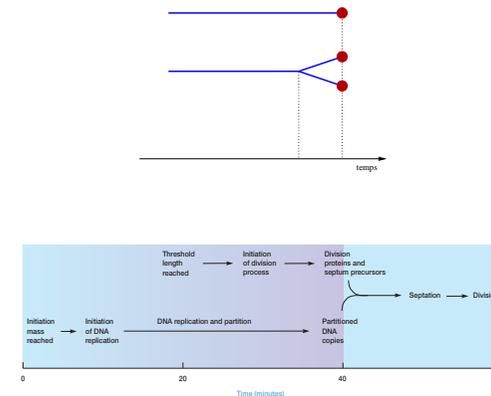


Figure 12.35 Control of the Cell Cycle in *E. coli*. A 60-minute interval between divisions has been assumed for purposes of simplicity (the actual time between cell divisions may be shorter). *E. coli* requires about 40 minutes to replicate its DNA and 20 minutes after termination of replication to prepare for division. The position of events on the time line is approximate and meant to show the general pattern of occurrences.

(illustrations "Prescott, Harley, Klein – Microbiology – McGraw-Hill 2002")

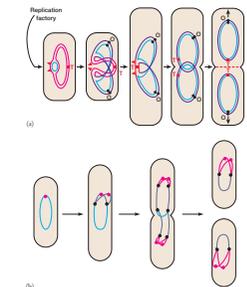


Figure 12.34 DNA Replication in Bacteria. (a) In slowly growing bacteria, the chromosome is replicated once before division. In this simplified illustration, the replicating chromosomal DNA is spooled through a membrane-bound replication factory, the factory then divides into two replication foci, and eventually the duplicated chromosomes separate and move to opposite ends of the cell. (b) DNA replication in rapidly growing bacteria is more complex. A new round of DNA replication is initiated before the original cell divides so that the DNA in daughter cells is already partially replicated. The colored circles at the ends of the DNA loops are replication origins, the black circles along the sides are replication forks. Newly synthesized DNA is in color, with the red representing the most recently synthesized DNA. Membrane attachments are not shown for sake of simplicity.

- ▶ modèles déterministes
 - + répandus
 - + valides en **grande population**
 - + plus simples que les modèles stochastiques
 - pas réalistes en petite population (extinction)
 - équilibre déterministe

- ▶ modèles stochastiques
 - + plus réalistes
 - + permet de traiter les problèmes d'extinction et de coexistence de façon différente et plus réalistes
 - + équilibre stochastique
 - plus complexes

- Introduction
- Premiers pas en R
- Processus de naissance et mort
- Processus de naissance/mort linéaire
- Processus de naissance/mort logistique
- Processus de Markov de saut pur $\rightarrow \mathbb{N}$
- Chemostat
- IBM
- Perspectives
- Bibliographie**

Bibliographie

- ▶ Allen, L. J. (2003).
An Introduction to Stochastic Processes with Biology Applications.
Prentice Hall.
- ▶ Campillo, F. and Joannides, M. (2009).
Processus stochastiques en temps continu pour la modélisation en écologie.
Master de Biostatistique, Montpellier I.
- ▶ Karlin, S. and Taylor, H. M. (1975).
A First Course in Stochastic Processes, Second Edition.
Academic Press, 2 edition.
- ▶ Klebaner, F. C. (2005).
Introduction to Stochastic Calculus with Applications.
Imperial College Press, 2 edition.
- ▶ Norris, J. R. (1998).
Markov chains.
Cambridge University Press.
- ▶ Van Kampen, N. G. (2007).
Stochastic Processes in Physics and Chemistry, Third Edition (North-Holland Personal Library).
North Holland.
- ▶ Wilkinson, D. J. (2006).
Stochastic Modelling for Systems Biology.
Chapman & Hall/CRC.