

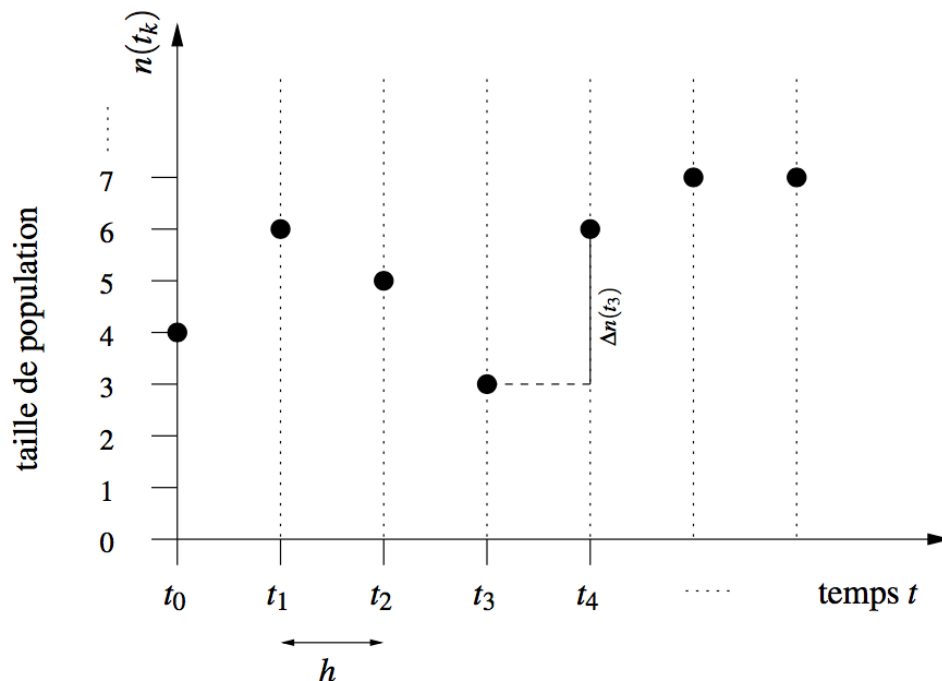
# Les bases de la dynamique des populations

À voir:

- [Bacaer 2009] très intéressante perspective historique
- [Boularas et al 2009] présentation très vivante et accessible des modèles différentiels
- [Otto et Dray2007] très complet et tourné vers les biologistes

## Modèles différentiels

Nous voulons modéliser l'évolution de la taille d'une population composée d'une seule espèce. Notons  $n(t)$  la taille de cette population à l'instant  $t$ , il s'agit d'une quantité entière. Nous allons modéliser l'évolution de cette taille à des instants  $t_k$ , que nous supposons pour simplifier équirépartis, i.e.  $t_k = k h$  avec  $h > 0$ :



Modéliser l'évolution de la taille de population consiste à définir la variation  $\Delta n(t_k)$  de cette taille entre les instants  $t_k$  et  $t_{k+1}$ :

$$n(t_{k+1}) = n(t_k) + \Delta n(t_k).$$

On suppose donc que ces accroissements dépendent de la taille courante de la population. Il est pertinent de s'intéresser à la variation de la taille de la population par unité de temps:

$$\frac{n(t_{k+1}) - n(t_k)}{h} = \frac{\Delta n(t_k)}{h}.$$

On fait l'hypothèse que les instants  $t_k$  sont rapprochés, i.e.  $h$  petit.

Dans l'équation précédente on fait tendre  $h$  vers 0 et  $k$  vers l'infini de telle sorte que  $t_k \rightarrow t$  pour  $t$  donné. On suppose aussi que  $\Delta n(t_k)$  tend vers l'infini de telle sorte que le rapport  $\Delta n(t_k)/h$  tende vers un certain  $F(n(t))$ :

$$\dot{n}(t) = F(n(t)).$$

Enfin, la taille  $n(t)$  de la population est supposée très grande et nous faisons le *changement* d'échelle suivant:

$$x(t) := \frac{n(t)}{M}$$

Ce changement de variable peut s'interpréter de différentes façons. Par exemple pour une population de bactéries:

- $M$  peut être vu comme l'inverse de la masse d'une bactérie, alors  $x(t)$  désigne la *biomasse* de la population;
- $M$  peut être le *volume* dans lequel vit cette population,  $x(t)$  est alors une *densité* de population;
- $M$  peut être simplement un *changement d'échelle*, si la taille de la population est de l'ordre de  $10^9$  individus et si  $M = 10^3$  alors  $x(t)$  désignera la taille de la population de méta-individus (1 méta-individu =  $10^3$  individus).

L'équation (???) devient:

$$\frac{\dot{n}(t)}{M} = \frac{1}{M} F\left(M \frac{n(t)}{M}\right)$$

et en posant  $f(x) := \frac{1}{M} F(Mx)$  on obtient l'équation différentielle ordinaire (EDO):

$$\dot{x}(t) = f(x(t)), \quad x(0) = x_0$$

et son état  $x(t)$  peut donc désigner la *taille* d'une population, sa *biomasse*, sa *densité* (nombre d'individus par unité de volume), ou bien encore sa *concentration* (massique ou molaire); pour simplifier nous dirons que  $x(t)$  "est" la population;  $x_0$  désigne la population initiale, supposée connue.

Dans beaucoup d'exemples de dynamique de population  $f$  est de la forme:

$$f(x) = r(x) x$$

où  $r(x)$  s'interprète comme un **taux de croissance per capita** (par individu). En effet si  $x(t+h) = x(t) + f(x(t))$  ( $h = 1$  unité de temps) et si par exemple  $f(x(t)) = 5$  il y alors eu un accroissement de 5 individus (dans l'échelle  $x$ ) sur la période de temps  $h$ : est-ce grand ou petit ? Cela est relatif à la taille  $x(t)$  de la population, c'est donc le rapport  $\frac{f(x(t))}{x(t)} = r(x(t))$  qui importe.

## Croissance exponentielle

Division cellulaire pouvant être vue comme un modèle d'ordre 1:

$$X \longrightarrow 2 X$$

La première étape consiste à appréhender la croissance géométrique (temps discret) et exponentielle (temps continu).

On considère une population dont la taille évolue de la façon suivante:

$$n(t_{k+1}) = n(t_k) + \lambda n(t_k) h - \mu n(t_k) h$$

où  $\lambda$  est le taux de naissance et  $\mu$  celui de mort. Il est nécessaire ici que l'intervalle de temps  $[t_k, t_{k+1}]$  soit suffisamment petit pour que  $n(t_k)$  évolue peu, mais aussi suffisamment grand pour que des événements de naissance et mort surviennent. Après changement d'échelle, l'équation précédente devient:

$$\dot{x}(t) = (\lambda - \mu) x(t), \quad x(0) = x_0$$

taux de naissance  $\lambda > 0$ , taux de mort  $\mu > 0$ .

qui admet la solution explicite suivante:

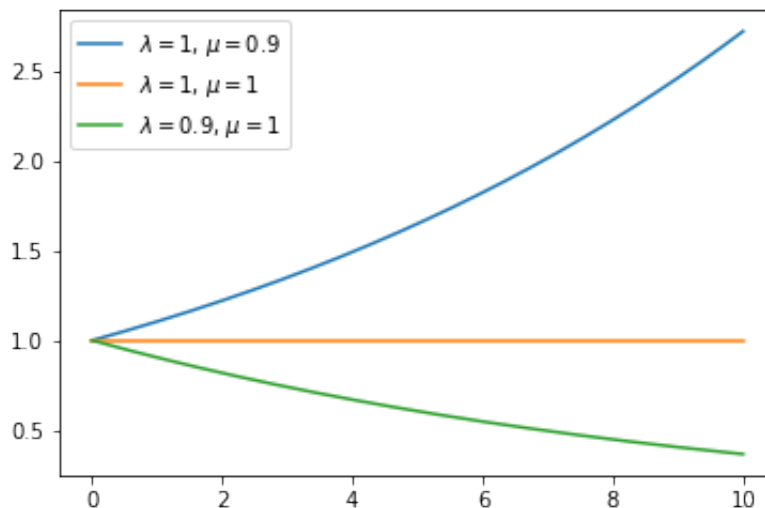
$$x(t) = x_0 e^{(\lambda - \mu) t}, \quad t \geq 0.$$

```
In [2]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

t0, t1 = 0, 10
temps = np.linspace(t0,t1,200, endpoint=True)

population = lambda t: x0*np.exp((rb-rd)*t)

legende = []
for x0, rb, rd in zip([1, 1, 1], [1, 1, 0.9], [0.9, 1, 1]):
    plt.plot(temps, population(temps))
    legende = legende + [r'\lambda=$'+str(rb)+r', $\mu=$'+str(rd),
]
plt.legend(legende, loc='upper left')
plt.show()
```



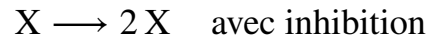
Ainsi

- lorsque  $\mu > \lambda$  la population croît exponentiellement
- lorsque  $\lambda < \mu$  la population tend exponentiellement vers 0.

On parlera de croissance (ou décroissance) malthusienne.

Lorsque  $\lambda < \mu$  la population décroît exponentiellement vite vers 0 mais à tout instant fini cette population est strictement positive, pourtant si  $M = 10^3$  et si  $x(t)$  descend en dessous de  $10^{-3}$  alors  $x(t)$  représentera moins d'un individu. Ce point n'est pas cohérent avec l'hypothèse de population grande et donc limite l'intérêt de ce modèle pour les petites tailles de population.

## Croissance logistique



En 1838, Pierre François Verhulst ([https://en.wikipedia.org/wiki/Pierre\\_Francois\\_Verhulst](https://en.wikipedia.org/wiki/Pierre_Francois_Verhulst)) (1804-1849) proposa un modèle de croissance dont le taux de croissance diminue linéairement en fonction de la taille de la population rendant ainsi compte de la capacité maximale d'accueil du milieu.

$$\dot{x}(t) = r \times \left(1 - \frac{x(t)}{K}\right) x(t), \quad x(0) = x_0$$

admet l'unique solution:

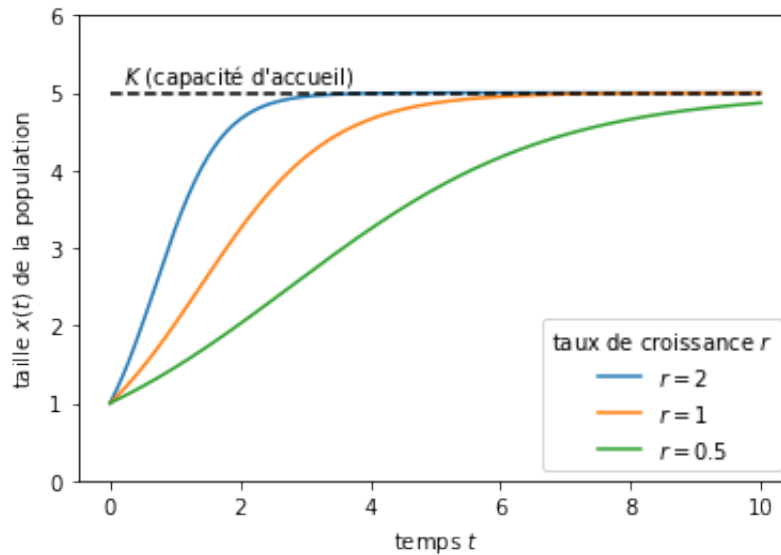
$$x(t) = K \frac{1}{1 + \left(\frac{K}{x_0} - 1\right) e^{-rt}} = \frac{1}{\frac{x_0}{K} (e^{rt} - 1) + 1} x_0 e^{rt}$$

```
In [3]: t0, t1 = 0, 10

temps = np.linspace(t0,t1,300, endpoint=True)

population = lambda t: K*1/(1+ (K/x0-1) * np.exp(-r*t))

x0, K = 1, 5
legende = []
for r in [2, 1, 0.5]:
    plt.plot(temps, population(temps))
    legende = legende + [r'$r=$'+str(r),]
plt.ylim([0,K*1.2])
plt.legend(legende, loc='lower right',title=r'taux de croissance $r$')
plt.plot([t0, t1], [K, K], color="k", linestyle='--')
plt.text((t1-t0)/50, K, r"$K$ (capacité d'accueil)",
        verticalalignment='bottom', horizontalalignment='left')
plt.xlabel(r'temps $t$')
plt.ylabel(r'taille $x(t)$ de la population')
plt.show()
```

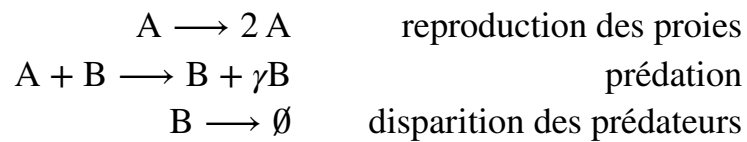


```
In [4]: from ipywidgets import interact, fixed

def pltlogistique(x0,K,r):
    population2 = K*1/(1+ (K/x0-1) * np.exp(-r*temps))
    plt.plot(temps, population2)
    plt.ylim([0,6])
    plt.plot([t0, t1], [K, K], color="k", linestyle='--')
    plt.show()

interact(pltlogistique, x0=(0.01,6,0.1), K=(0.01,6,0.1), r=(0.1,20,
0.1))
plt.show()
```

## Modèle de Lotka-Volterra



matrice de Petersen:

réaction	ordre	A	B	taux de réaction
reproduction des proies	1	+1	0	$k_1 [A]$
prédation	2	-1	$\gamma$	$k_2 [A][B]$
disparition des prédateurs	1	0	-1	$k_3 [B]$

$$\begin{aligned}
 \frac{d[A]}{dt} &= k_1 [A] - k_2 [A][B] \\
 \frac{d[B]}{dt} &= \gamma k_2 [A][B] - k_3 [B]
 \end{aligned}$$

[1kg d'herbe ne fait pas 1kg de vache]

Il existe de nombreuses présentations de ce modèles, pour un résumé mathématique précis voir par exemple [ce document PDF \(http://w3.bretagne.ens-cachan.fr/math/people/gregory.vial/files/cplts/volterra.pdf\)](http://w3.bretagne.ens-cachan.fr/math/people/gregory.vial/files/cplts/volterra.pdf).

Le [modèle de Lotka-Volterra \(https://fr.wikipedia.org/wiki/Équations\\_de\\_Lotka-Volterra\)](https://fr.wikipedia.org/wiki/Équations_de_Lotka-Volterra) représente deux populations en interaction:

- des proies, de taille  $x_1(t)$ , ayant accès à une ressource illimitée (non modélisée)
- et des prédateurs, de taille  $x_2(t)$ , se nourrissant de proies.

On suppose que:

- en l'absence de prédateurs, la population de proies croît de façon exponentielle selon un taux  $r_1$ ;
- en l'absence de proies, la population de prédateurs décroît de façon exponentielle selon un taux  $r_2$ .

On suppose que  $r_1$  dépend de  $x_2(t)$  et que  $r_2$  dépend de  $x_1(t)$ :

- $r_1 = a - b x_2(t)$ , où  $a$  est le taux de naissance des proies en l'absence de prédateurs et  $b x_2(t)$  est le taux de prédation que l'on suppose linéaire en  $x_2(t)$ ;
- $r_2 = c x_1(t) - d$ , où  $d$  est le taux de mort des prédateurs en l'absence de proies et  $c x_1(t)$  est le taux de naissance des prédateurs que l'on suppose linéaire en  $x_1(t)$ .

On obtient donc un système de deux équations différentielles couplées:

$$\begin{aligned}\dot{x}_1(t) &= [a - b x_2(t)] x_1(t) \\ \dot{x}_2(t) &= [c x_1(t) - d] x_2(t)\end{aligned}$$

ce système n'admet pas de solution explicite, on doit faire appel à une méthode numérique.

La solution est périodique de période  $\sqrt{a c}$ .

Voir par exemple dans le [SciPy Cookbook \(http://scipy-cookbook.readthedocs.io/items/LotkaVolterraTutorial.html\)](http://scipy-cookbook.readthedocs.io/items/LotkaVolterraTutorial.html).

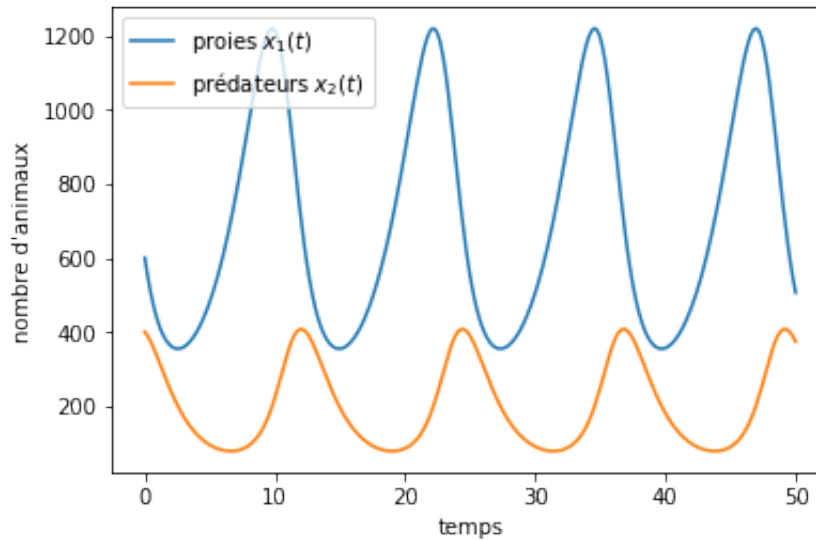
```
In [5]: from scipy.integrate import odeint

a, b, c, d = 0.4, 0.002, 0.001, 0.7
def f(x, t):
    x1, x2 = x
    return [a * x1 - b * x1 * x2,
            c * x1 * x2 - d * x2]

x0 = [600, 400]
t = np.linspace(0, 50, 250)
x_t = odeint(f, x0, t)
```



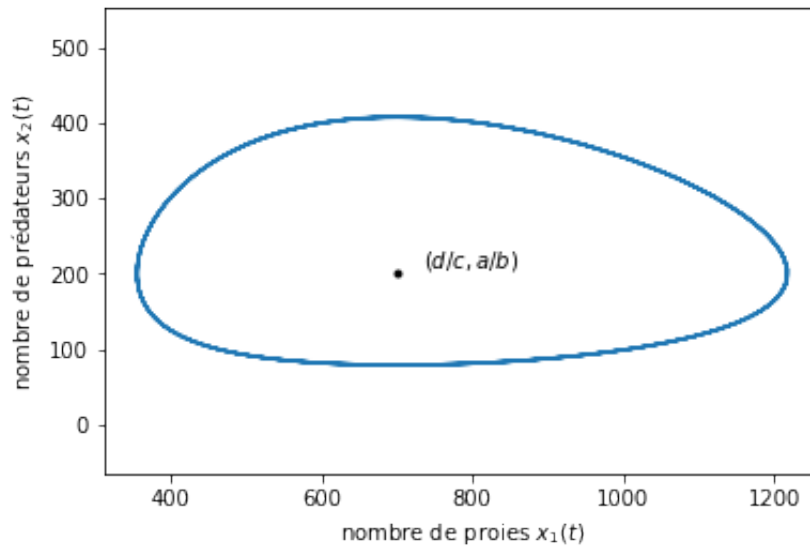
```
In [6]: plt.plot(t, x_t[:,0], label=r"proies  $x_1(t)$ ")
plt.plot(t, x_t[:,1], label=r"prédateurs  $x_2(t)$ ")
plt.xlabel("temps")
plt.ylabel("nombre d'animaux")
plt.legend()
plt.show()
```



## Espace des phases

Au lieu de tracer  $t \rightarrow x_1(t)$  et  $t \rightarrow x_2(t)$ , on trace les points  $(x_1(t), x_2(t))$  lorsque  $t$  varie, donc le temps n'apparaît plus, il s'agit d'une courbe dans l'espace des phases ([https://fr.wikipedia.org/wiki/Espace\\_des\\_phases](https://fr.wikipedia.org/wiki/Espace_des_phases)).

```
In [7]: plt.plot(x_t[:,0], x_t[:,1])
plt.xlabel(r"nombre de proies  $x_1(t)$ ")
plt.ylabel(r"nombre de prédateurs  $x_2(t)$ ")
marker_style = dict(linestyle=':', markersize=10)
equilibre = [d/c, a/b]
plt.plot(equilibre[0], equilibre[1], marker='.', color="k")
plt.text(1.05*equilibre[0], 1.05*equilibre[1], r' $(d/c, a/b)$ ')
plt.xlim(300, 1300)
plt.ylim(0, 500)
plt.axis('equal') # les échelles en x et y sont égales
plt.show()
```



```

In [8]: echelle = np.linspace(0.3, 0.9, 5)
couleurs = plt.cm.winter(np.linspace(0.3, 1., len(echelle)))

for v, col in zip(echelle, couleurs):
    val_ini = np.multiply(v, equilibre)
    X = odeint( f, val_ini, t)
    plt.plot( X[:,0], X[:,1], lw=1, color=col,
              label=r'$({:.f}, {:.f})$' % tuple(val_ini) )

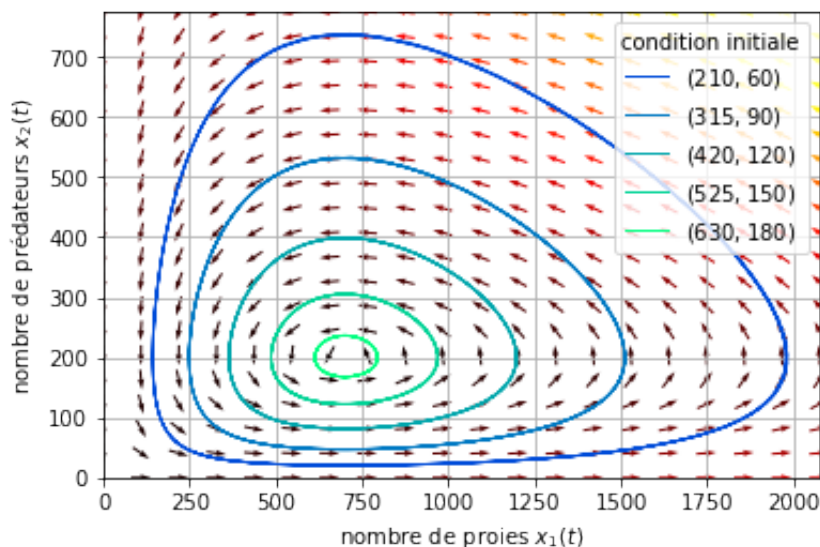
x1max = plt.xlim(xmin=0)[1]
x2max = plt.ylim(ymin=0)[1]

nb_points = 20

x1 = np.linspace(0, x1max, nb_points)
x2 = np.linspace(0, x2max, nb_points)
X1, X2 = np.meshgrid(x1, x2)
DX1, DX2 = f([X1, X2], 0)
vecteurs = np.hypot(DX1, DX2)      # norme du taux de croissance
vecteurs[vecteurs == 0] = 1.        # éviter la division par 0
DX1 /= vecteurs                    # normalisation de chaque vecteur
DX2 /= vecteurs

plt.quiver(X1, X2, DX1, DX2, vecteurs, pivot='mid', cmap=plt.cm.hot)
plt.xlabel(r"nombre de proies  $x_1(t)$ ")
plt.ylabel(r"nombre de prédateurs  $x_2(t)$ ")
plt.legend(title="condition initiale")
plt.grid()
plt.xlim(0, x1max)
plt.ylim(0, x2max)
plt.show()

```



# References

[^](#ref-1) Nicolas Bacaër. 2009. *Histoires de mathématiques et de populations*.

[^](#ref-2) Driss Boularas and Daniel Fredon and Daniel Petit. 2009. *Mini Manuel de Mathématiques pour les sciences de la vie et de l'environnement*.

[^](#ref-3) Otto, Sarah P. and Day, Troy. 2007. *A Biologist's Guide to Mathematical Modeling in Ecology and Evolution*.