

# Chapitre 2 : Systèmes dynamiques

# Sommaire I

- ▶ Systèmes qui évoluent dans le temps.
- ▶ La variable temps est soit continue ou discrete
- ▶ Exemple : marche aléatoire
- ▶ Propriétés : points fixes, cycles limites et chaos

## Sommaire II

- ▶ Réseaux de Hopfield
- ▶ Exemple de systèmes dynamiques, équation de bilan, taux de changement
- ▶ Modélisation par le concept de réservoirs et vannes
- ▶ Résolution d'équations différentielles : Méthode d'Euler et de Runge-Kutta
- ▶ Equation aux dérivées partielles, équations de conservation

## 2.1 Que sont les systèmes dynamiques ?

- ▶ Systèmes qui évoluent dans le temps
- ▶ On décrit comment l'état  $s$  de tels systèmes change au cours du temps
- ▶  $s$  peut être multivarié : un vecteur :  $s = (s_1, s_2, \dots, s_N)$
- ▶ système dynamique à **temps discret** :

$$s(t + \Delta t) = F(s(t))$$

où  $\Delta t$  est la durée du pas de temps. Si on mesure le temps en unités de  $\Delta t$ , on obtient

$$s(t + 1) = F(s(t))$$

# Systèmes dynamiques

On peut aussi écrire :

$$s(t + \Delta t) = \underbrace{s(t) + f(s(t))\Delta t}_{F(s)}$$

ou encore

$$\frac{s(t + \Delta t) - s(t)}{\Delta t} = f(s(t))$$

- ▶ système dynamique à **temps continu** :  $\Delta t \rightarrow 0$  et l'évolution temporelle est décrite par une équation différentielle ordinaire

$$\frac{ds}{dt} \equiv \dot{s}(t) = f(s(t))$$

# Remarques

- ▶ Le temps discret est souvent un choix du modélisateur pour simplifier la représentation en temps continu.
- ▶ le temps discret est ainsi souvent une façon numérique de résoudre un système à temps continu.
- ▶ Mais pas seulement : on peut s'intéresser à ce qui se passe à l'échelle de la seconde, du jour, du mois, de l'année : taux d'intérêt, systèmes électroniques cadencés, transactions.

# Généralité de la description

Qu'en est-il si on a des dérivées secondes ?

$$\dot{s} + \ddot{s} = f(s)$$

On pose alors

$$\begin{cases} \dot{s} = y \\ \dot{y} = f(s) - y \end{cases}$$

et on a à nouveau une expression de la forme  $\dot{s} = f(s)$  où  $s$  à été étendu avec une composante de plus

$$s \rightarrow \begin{pmatrix} s \\ y \end{pmatrix}$$

## Cas discret

Pour les système discret, une «dérivée seconde» est reflété par la présence d'un terme en  $t - 1$  :

$$s(t + 1) = s(t - 1) + F(s(t))$$

On peut faire le même truc :

$$\begin{cases} y(t + 1) = s(t) \\ s(t + 1) = F(s(t)) + y(t) \end{cases}$$

NB Certains systèmes sont **réversibles** ou **inversibles** dans le temps



## Cas où $f = f(s, t)$

Il se peut que la fonction qui décrit l'évolution dépende elle-même du temps (p. ex par une contrainte extérieure qui varie).

$$\dot{s} = f(s, t)$$

peut s'écrire comme

$$\begin{cases} \dot{s} = f(s, y) \\ \dot{y} = 1 \end{cases}$$

avec  $y(0) = 0$ .

## Cas discret $F = F(s, t)$

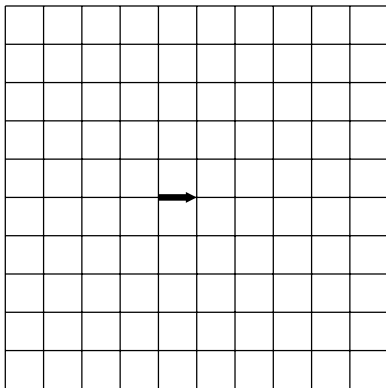
$$s(t+1) = F(s(t), t)$$

peut s'écrire comme

$$\begin{cases} s(t+1) = F(s, y) \\ y(t+1) = y(t) + 1 \end{cases}$$

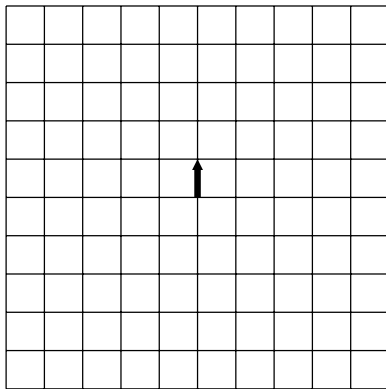
avec  $y(0) = 0$ .

## 2.2 Exemple de système dynamique à temps discret : Marche aléatoire



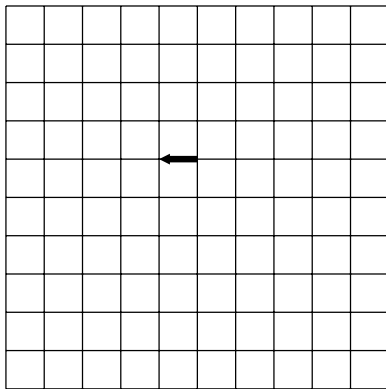
Random Walk

## 2.2 Exemple de système dynamique à temps discret : Marche aléatoire



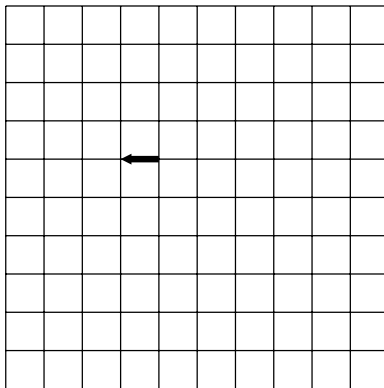
Random Walk

## 2.2 Exemple de système dynamique à temps discret : Marche aléatoire



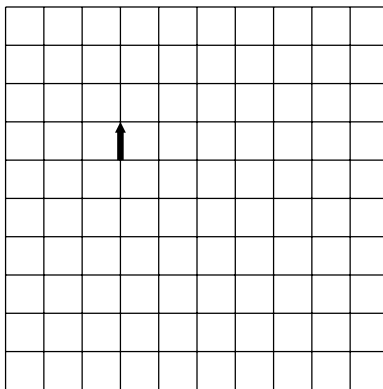
Random Walk

## 2.2 Exemple de système dynamique à temps discret : Marche aléatoire



Random Walk

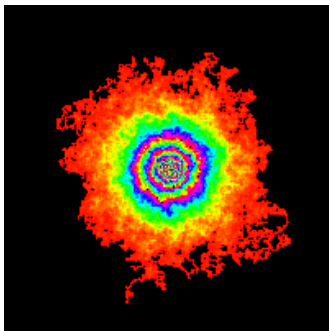
## 2.2 Exemple de système dynamique à temps discret : Marche aléatoire



Random Walk

# Simulation numérique

On simule 1000 répétitions d'une marche aléatoire de 1000 pas



Voir demo : C++/RandomWalk\$ ./randomWalk 300 300 1000



# Marche aléatoire

C'est un système **stochastique**

- ▶ Cela décrit le mouvement d'un ivrogne, d'animaux, de particules, etc
- ▶ Facile à simuler dans un ordinateur.
- ▶ On peut se poser des questions comme «quand est ce que, en moyenne, la particule atteindra pour la première fois une région donnée».
- ▶ Cela donne une implémentation microscopique du processus de **diffusion**

# Marche aléatoire : description mathématique

$$x(t + \Delta t) = x(t) + v_x(t)$$

$$y(t + \Delta t) = y(t) + v_y(t)$$

où  $(v_x, v_y)$  est choisi **aléatoirement** à chaque itération dans l'ensemble  $\{(\Delta x, 0), (-\Delta x, 0), (0, \Delta y), (0, -\Delta y)\}$  de façon uniforme.

## Analyse mathématique

La position  $x(t)$  après un temps  $t$  vaut

$$\begin{aligned}x(t) &= x(t-1) + v_x(t-1) = x(t-2) + v_x(t-2) + v_x(t-1) \\&= x(0) + \sum_{\tau=0}^{t-1} v_x(\tau)\end{aligned}\tag{1}$$

Quelle est la position moyenne (moyenne d'ensemble) ?

$$\langle x(t) \rangle = x_0 + \sum_{\tau=0}^{t-1} \langle v_x(\tau) \rangle$$

avec, pour tout temps

$$\langle v_x(\tau) \rangle = \frac{1}{4}(-\Delta x) + \frac{1}{2} \times 0 + \frac{1}{4}\Delta x = 0$$

Donc  $\langle x(t) \rangle = x_0$  En moyenne la particule est symétriquement placée par rapport à  $x_0$

## Ecart type

Mais, en moyenne, à quelle distance  $d$  se trouve-t-elle de  $x_0$  après un temps  $t$  ?

$$d \approx \sqrt{\langle (x(t) - x_0)^2 \rangle}$$

D'où

$$d^2 = \langle \sum_{i=0}^{t-1} v_x(i) \sum_{j=0}^{t-1} v_x(j) \rangle = \sum_{i,j=0}^{t-1} \langle v_x(i) v_x(j) \rangle$$

Pour  $i \neq j$ ,  $v_x(i)$  et  $v_x(j)$  sont statistiquement indépendants et

$$\langle v_x(i) v_x(j) \rangle = 0, \quad i \neq j$$

Pour  $i = j$  on a que  $\langle v_x(i) v_x(i) \rangle = \langle v_x^2 \rangle$  et que

$$\langle v_x^2 \rangle = \frac{1}{4}(\Delta x^2) + \frac{1}{2} \times 0 + \frac{1}{4} \Delta x^2 = \frac{\Delta x^2}{2}$$

## Ecart type

Donc

$$\begin{aligned} d^2 &= \sum_{i \neq j}^{t-1} \langle v_x(i) v_x(j) \rangle + \sum_{i=0}^{t-1} \langle v_x^2 \rangle \\ &= \sum_{i=0}^{t-1} \frac{\Delta x^2}{2} = \frac{\Delta x^2}{2} t \end{aligned} \tag{2}$$

D'où

$$d = \sqrt{\frac{\Delta x^2}{2} t}$$

Il faut attendre 4 fois plus de temps pour que la particule s'éloigne deux fois plus de son point de départ.

## Description en terme de probabilité

On verra que la probability  $\rho(x, y, t)$  d'avoir en  $(x, y)$  et au temps  $t$  une particule qui bouge aléatoirement est donnée par **l'équation de diffusion**

$$\partial_t \rho = D \nabla^2 \rho$$

On montre que

$$d(t) = \sqrt{\int x^2 \rho(x, t) dx} = \sqrt{2Dt} = \sqrt{\frac{\Delta x^2}{2} t}$$

On en déduit que pour notre marche aléatoire, le coefficient de diffusion  $D$  vaut

$$D = \frac{1}{2t} d^2(t) = \Delta x^2 / 4$$

## 2.3 Points fixes, cycles limites et chaos

- ▶ Est-ce que l'évolution d'un système dynamique se stabilise, après une période de transition (transient) ?
- ▶ Cela peut être un **point fixe** de la dynamique
- ▶ Ou éventuellement un **cycle limite** : après un certain temps, le système oscille entre plusieurs états possibles, toujours les mêmes (ex une balançoire).

Cas discret :

$$s(t+1) = s(t) \quad \text{donc } F(s) = s$$

Cas continu :

$$\dot{s} = 0 \quad \text{donc } f(s) = 0$$

## Points Fixes attractifs ou répulsifs

Soit  $f(s^*) = 0$  pour un  $s^*$  donné. Proche de  $s^*$ , on peut écrire

$$s(t) = s^* + \epsilon(t) \quad \dot{s} = \dot{\epsilon}$$

Si  $\epsilon$  est assez petit, par un développement de Taylor, on a

$$f(s^* + \epsilon) = f(s^*) + \epsilon \left( \frac{df}{ds} \right)_{s=s^*} \equiv \lambda \epsilon$$

Donc, dans le voisinage de  $s^*$ , le système dynamique s'écrit

$$\dot{s} = f(s) \implies \dot{\epsilon} = \lambda \epsilon$$



# Points Fixes attractifs ou répulsifs

La solution est

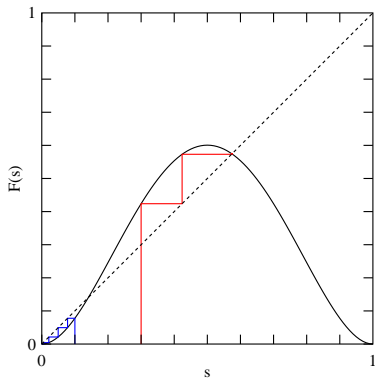
$$\epsilon(t) = \epsilon(0)e^{\lambda t}$$

- ▶ Si  $\lambda < 0$  on revient vers  $s^*$ , et  $s^*$  est un point fixe attractif.
- ▶ Si  $\lambda > 0$  on s'éloigne de  $s^*$ , et  $s^*$  est un point fixe répulsif.
- ▶ Si  $\lambda = 0$ , il faut voir le signe de la dérivée 2e de  $f$

**NB :** Si  $s$  est un vecteur, la discussion est plus riche : Il faut prendre le Jacobien de  $f$  et il peut y avoir des directions répulsives et d'autres attractives

## Cas discret : représentation graphique

Dans le cas discret, on peut utiliser le concept de «iterative map» ou encore «carte itérative» pour représenter l'évolution de façon graphique



## Cas d'une map linéaire

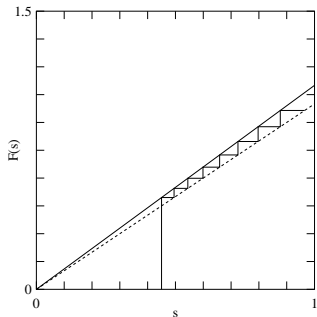
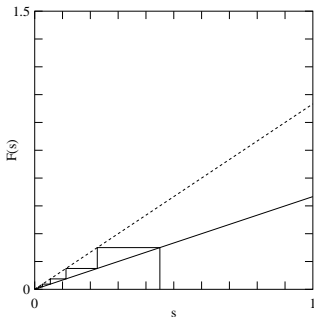
Soit

$$s(t+1) = F(s(t)) = \gamma s(t)$$

Pour que  $s^*$  soit un point fixe, il faut que  $F(s^*) = s^*$  donc que

$$s^* = \gamma s^* \quad \text{donc} \quad s^* = 0$$

Si  $0 < \gamma < 1$ , ce point est attractif, et si  $\gamma > 1$  il est répulsif.



On montre aussi que si  $-1 < \gamma < 0$ , il est attractif et si  $\gamma < -1$ , il est répulsif.

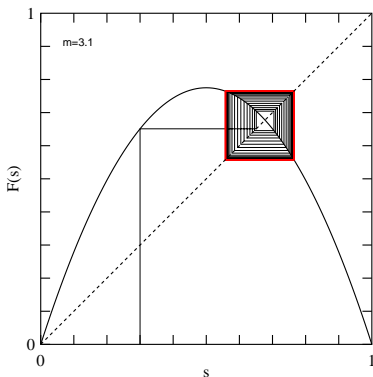
## Cycle limite : la «logistic Map»

On considère maintenant

$$s(t+1) = F(s(t)) = \gamma s(1-s)$$

avec  $0 < \gamma < 4$ , ce système reste dans  $[0, 1] \times [0, 1]$ .

$$\gamma = 3.1$$

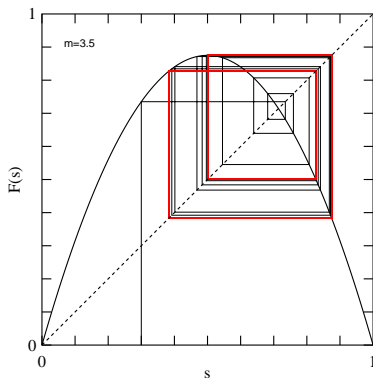


On observe un cycle de longueur 2 plutôt qu'un point fixe (les 40 premières itérations sont en noir, les 20 suivantes en rouge). Ce cycle de longueur 2 peut être obtenu comme points fixes de

$$F \circ F = \gamma^2 s(1-s)(1-\gamma s(1-s))$$

Que se passe-t-il quand  $\gamma$  augmente ?

$$\gamma = 3.5$$

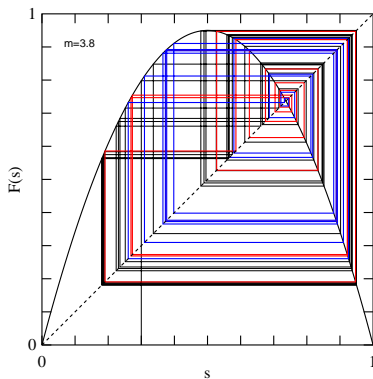


Il y a maintenant un cycle de longueur 4.

Ce processus de d'augmentation de la longueur des cycles quand un paramètre varie, s'appelle **bifurcation**

# Chaos

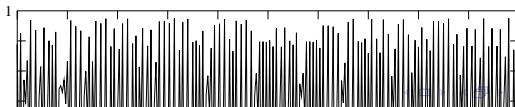
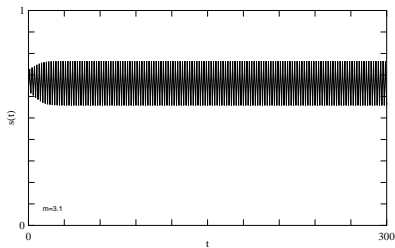
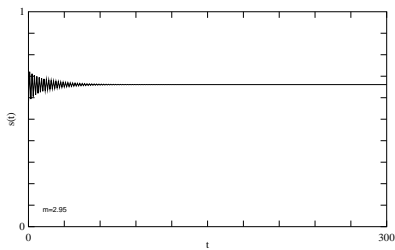
$$\gamma = 3.8 > 3.5699\dots$$



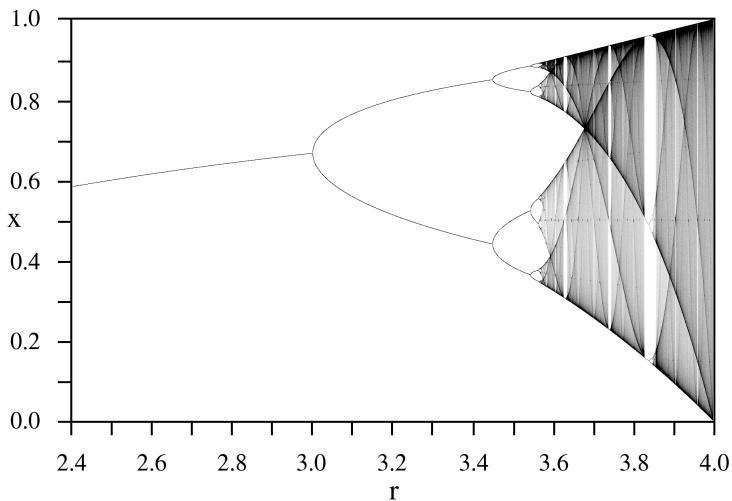
Il n'y a maintenant plus de cycles mais du chaos (imprédictibilité et sensibilité aux conditions initiales.)

NB les 40 premières itérations sont en noir, les 20 suivantes en rouge et les 20 dernières en bleu.

# Graphes de $s(t)$



# Diagramme de Bifurcation



$$s(t+1) = F(s(t)) = rs(1-s)$$



## 2.4 Réseaux de Hopfield

C'est un exemple de système dynamique à temps discret.

- ▶ Un réseau de Hopfield est un réseau totalement connectés de  $N$  neurones artificiels.
- ▶ Mémoire des informations et peut les restituer si soumis à une information partielle, bruitée ou dégradée.
- ▶ Modèle de mémoire associative.

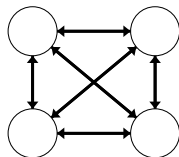
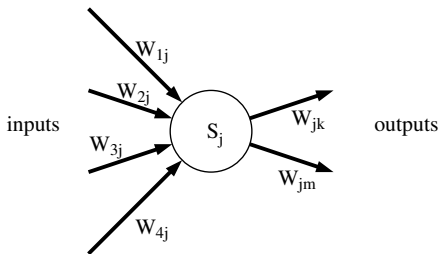


(a) 8개의 대표패턴들



(b) 노이즈가 있는 '3'의 입력에 대한 출력

# Neurone artificiel



Etats possibles :  $s_i(t) = \begin{cases} 1 & \text{noir} \\ -1 & \text{blanc} \end{cases}$

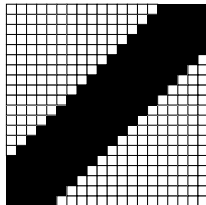
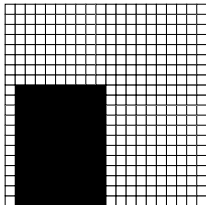
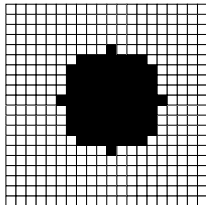
Evolution

$$s_i(t+1) = F_w(s(t))$$



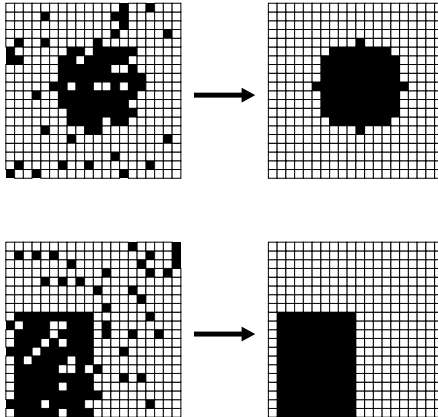
## Exemple : images apprises

Point fixe :  $s_i(t+1) = s_i(t) \quad \forall i$

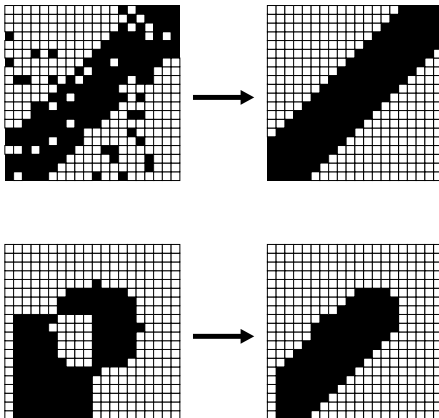


Pour mémoriser des images il faut ajuster les poids  $w_{ij}$

# Images reconstruites : convergence vers le point fixe



## Images reconstruites (suite)



Dans le dernier cas, l'image reconstruite n'est pas une image apprise

# Neurones

- ▶ Un neurone  $n_i$  est une entité ayant des entrées et des sorties
- ▶ Il possède un état interne  $s_i(t) \in \{-1, 1\}$
- ▶ Il est relié à d'autres neurones  $n_j$  avec une connexion (synapse) de poids  $w_{ij}$
- ▶ L'état des neurones évolue dans le temps par la relation

$$s_i(t+1) = F_w(s(t)) = T(h_i(s(t)))$$

- ▶ Où  $T$  s'appelle la **fonction de transfert** et  $h_i$  est le **potentiel synaptique**

# Fonction de Transfert et potentiel synaptique

Règle d'évolution :

$$s_i(t+1) = F_w(s(t)) = T(h_i(s(t)))$$

Potentiel synaptique :

$$h_i(t) = \sum_j w_{ij} s_j(t)$$

Fonction de transfert :

$$T(x) = \text{sgn}(x) = \begin{cases} 1 & \text{if } x \geq 1 \\ -1 & \text{if } x < 1 \end{cases}$$

NB :  $T$  s'appelle la fonction «signe»

## Règle de Hebb (apprentissage)

- ▶ On veut choisir les poids synaptiques  $w_{ij}$  pour qu'une *image*,  $S^{(0)} = [s_1^{(0)}, s_2^{(0)}, \dots, s_N^{(0)}]$  soit un **point fixe** de l'évolution



$$s_i(t+1) = s_i(t) \quad \text{si} \quad s_i(t) = s_i^{(0)}$$

- ▶ On veut aussi que si on prend comme condition initiale une image proche de  $S^{(0)}$ , le réseau la reconstruise, c'est à dire que

$$s_i(t) \rightarrow s_i^{(0)} \quad \text{si} \quad t \rightarrow \infty$$

### Règle de Hebb :

On choisit les poids ainsi :

$$w_{ij} = \frac{1}{N} s_i^{(0)} s_j^{(0)}$$



## Point Fixe

On suppose que  $s_j(t) = s_j^{(0)}$  et on veut vérifier que  $s_i(t+1) = s_i^{(0)}$

$$s_i(t+1) = \operatorname{sgn}\left(\sum_j w_{ij} s_j(t)\right) = \operatorname{sgn}\left(\sum_j w_{ij} s_j^{(0)}\right)$$

Mais

$$\sum_j w_{ij} s_j^{(0)} = \frac{1}{N} s_i^{(0)} \sum_j (s_j^{(0)})^2 = s_i^{(0)}$$

et, donc,

$$s_i(t+1) = \operatorname{sgn}(s_i^{(0)}) = s_i^{(0)}$$



## Reconstruction de l'image

Considérons maintenant le cas où l'évolution du réseau ne commence pas dans l'état mémorisé mais dans un état qui en diffère par  $n$  pixels faux

$$s_i(t) = -s_i^{(0)} \quad \text{pour} \quad i = i_1, \dots, i_n$$

Alors

$$\begin{aligned} h_i(t) &= \sum_j w_{ij} s_j(t) = \frac{1}{N} s_i^{(0)} \sum_j s_j^{(0)} s_j(t) \\ &= \frac{1}{N} s_i^{(0)} (-n + N - n) = \left(1 - \frac{2n}{N}\right) s_i^{(0)} \end{aligned}$$

car il y a  $n$  valeurs qui donneront  $s_j^{(0)} s_j(t) = -1$  et  $N - n$  valeurs qui donneront  $s_j^{(0)} s_j(t) = 1$ .

Ainsi, pour  $n < N/2$ ,

$$s_i(t+1) = \text{sgn}(h_i(t)) = s_i^{(0)}$$

et on retrouve l'image originale en **1 itération**.

## Mémorisation de plusieurs images

Pour mémoriser  $p$  images, la règle de Hebb s'écrit :

$$w_{ij} = \frac{1}{N} \sum_{\ell=1}^p s_i^{(\ell)} s_j^{(\ell)}$$

Montrons à quelle condition  $s = s^{(k)}$ ,  $1 \leq k \leq p$  sera effectivement un point fixe :

$$\begin{aligned} s_i(t+1) &= \operatorname{sgn} \left( \sum_j w_{ij} s_j^{(k)} \right) = \operatorname{sgn} \left( \frac{1}{N} \sum_j \sum_{\ell} s_i^{(\ell)} s_j^{(\ell)} s_j^{(k)} \right) \\ &= \operatorname{sgn} \left( \frac{1}{N} \sum_j [s_i^{(k)} s_j^{(k)} s_j^{(k)} + \sum_{\ell \neq k} s_i^{(\ell)} s_j^{(\ell)} s_j^{(k)}] \right) \\ &= \operatorname{sgn} \left( s_i^{(k)} + \frac{1}{N} \sum_j s_i^{(k)} \sum_{\ell \neq k} s_i^{(\ell)} s_j^{(\ell)} \right) \end{aligned}$$

## Mémorisation de plusieurs images (suite)

Pour que  $s_i(t+1) = s_i^{(k)}$ , il faut que

$$-1 < \frac{1}{N} \sum_j s_i^{(k)} \sum_{\ell \neq k} s_i^{(\ell)} s_j^{(\ell)} < 1$$

soit compris entre -1 et 1, strictement. Cela est plus probable si les images sont bien équilibrées en pixels blancs et noirs.

Mais il faut aussi que  $p$  reste petit devant  $N$ . En pratique, cette limite est environ  $p/N = 0.1$  avec la règle de Hebb.

## Mémorisation de plusieurs images : règle de projection

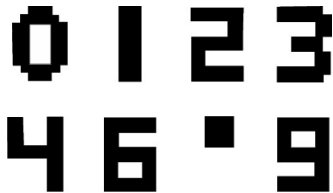
La règle dite de projection permet des valeurs  $p/N$  plus grande. On définit une matrice auxiliaire  $Q$  telle que

$$Q_{nm} = \frac{1}{N} \sum_{i=1}^N s_i^{(n)} s_i^{(m)}$$

et on calcule les poids synaptiques selon la relation

$$w_{ij} = \frac{1}{N} \sum_{m,n=0}^{p-1} s_i^{(m)} Q_{mn}^{-1} s_j^{(n)}$$

## Illustration



(a) 8개의 대표패턴들



(b) 노이즈가 있는 '3'의 입력에 대한 출력

## 2.5 Modélisation par des systèmes dynamiques

L'idée centrale de la modélisation avec

$$s(t + \Delta t) = F(s(t)) \quad \text{ou} \quad \dot{s} = f(s)$$

est d'écrire une **équation de bilan**.

Par exemple  $s(t + \Delta t) - s(t)$  est la variation de la quantité  $s$  durant l'intervalle de temps  $\Delta t$ . Il y a des processus qui créent du  $s$ , d'autres qui en détruisent.

$$s(t + \Delta t) - s(t) = [\text{creation rate} - \text{destruction rate}] \Delta t$$

Pour spécifier un modèle il faut donc définir les taux de création et de destruction.

Cela définit  $F(s)$  comme

$$F(s) = s + [\text{creation rate} - \text{destruction rate}] \Delta t$$

# Equation de bilan

Dans la limite  $\Delta t \rightarrow 0$  on obtient l'eq. différentielle de bilan



$$\dot{s} = f(s) = \text{creation rate} - \text{destruction rate}$$

La variation de  $s$  est donnée par la quantité de  $s$  créée moins la quantité de  $s$  détruite.

Exemple : croissance d'une population décrite par le nombre  $P(t)$  d'individus qui la compose. On a donc le bilan

$$\dot{P}(t) = N(t) - D(t)$$

où  $N(t)$  est le taux de naissances au temps  $t$  et  $D(t)$  le taux de décès.



# Croissance d'une population

Un exemple classique est  $N(t) = r_1 P(t)$  et  $D(t) = r_2 P(t)$ , d'où

$$\dot{P} = rP$$

où  $P(t)$  est la population d'une espèce donnée et  $r = r_1 - r_2$ .  
Le résultat est

$$P(t) = P_0 e^{rt}$$

où  $P_0$  est la population au temps  $t = 0$ .

C'est une croissance exponentielle, sans limite, si  $r > 0$ .

## Capacité maximum

Plus adaptée est la croissance avec «capacité» maximum  $M$  de l'environnement. Le taux de croissance décroît à mesure que le nombre d'individus s'approche de  $M$ . Du coup, on pose

$$r \rightarrow r \left( 1 - \frac{P}{M} \right)$$

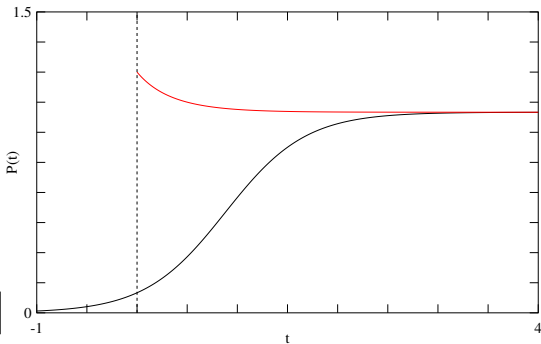
Le modèle dynamique devient

$$\dot{P} = r \left( 1 - \frac{P}{M} \right) P$$

## Croissance d'une population (suite)

- La solution est une fonction dite **logistique** : croissance puis saturation :

$$P(t) = \frac{M}{1 + \frac{M-P_0}{P_0} e^{-rt}}$$



## Croissance d'une population (suite)

- ▶ Si  $P(0) < M$  il y a croissance de plus en plus lente à mesure que  $P$  s'approche de  $M$ .
- ▶ Si  $P(0) > M$ , il y a décroissance.
- ▶ Il y a un point fixe attracteur :

$$P = M$$

- ▶ La population  $P = 0$  est aussi un point fixe ( $f(P) = 0$  si  $P = 0$ ), mais instable, ou répulsif.

## Modèle de Lotka-Volterra (1910 et 1926)

Deux populations en interaction : «lapins»  $\ell(t)$  et «renard»  $r(t)$ .  
Les **proies** et les **prédateurs**.

Les  $r$  mangent les  $\ell$  et cela augmente leur taux de coissance. Cela augment aussi le taux de décès des  $\ell$ .

La quantité  $r(t)\ell(t)$  est le nombre total d'interactions entre les individus des deux populations.

$$\frac{d\ell}{dt} = k_{\ell}\ell(t) - k_{r\ell}r(t)\ell(t)$$

$$\frac{dr}{dt} = -k_r r(t) + k_{\ell r}\ell(t)r(t)$$

On a des coefficients qui décrivent l'habileté des uns à échapper aux prédateurs et des autres à détecter une proie.

# Modèle de Lotka-Volterra

## Comportement oscillant

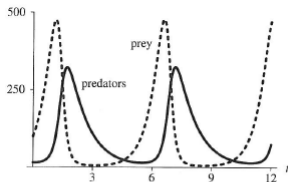
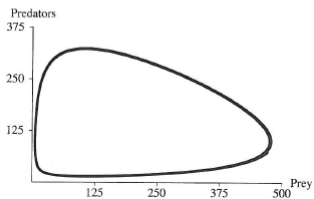
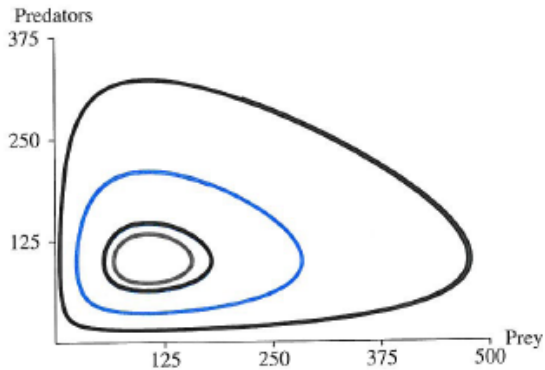


Figure 6.4.2 Graph of populations versus time in months



## Rôle des conditions initiale



## 2.6 Modélisation par les réservoir et vannes

Une dynamique du type

$$\dot{s} = f(s)$$

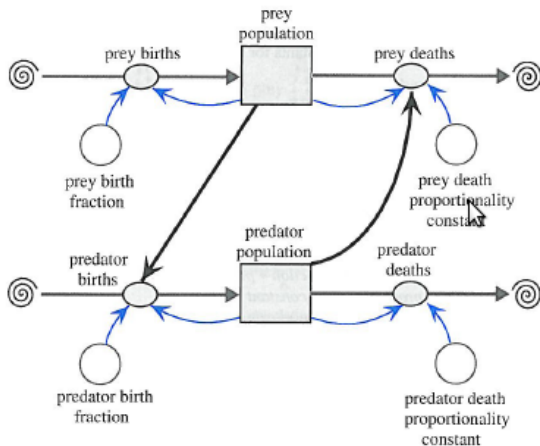
nous dit qu'on a une grandeur  $s$  qui croît ou décroît dans le temps.

- ▶ On peut voir  $s$  comme la quantité d'une substance dans un réservoir
- ▶  $\dot{s}$  est alors le taux de variation du réservoir.
- ▶ C'est une sorte de vanne qui spécifie combien de substance entre ou sort du réservoir. Cela peut dépendre de  $s$ .
- ▶ on suppose qu'il y a toujours de quoi alimenter la vanne en substance (source infinie)
- ▶ Ce qui sort du réservoir disparaît dans un puit sans fond.

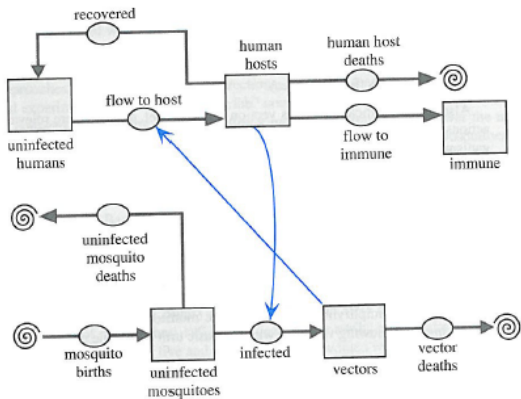
Voir diagrammes à la «Stella»



## Exemple pour le modèle de Lotka-Volterra



## Exemple pour la malaria



# Liste d'autres exemples de modèles dynamiques

- ▶ Dynamique des population
- ▶ Dosage de médicament et élimination par l'organisme
- ▶ Reaction chimiques
- ▶ propagation d'épidémie
- ▶ Radio-activité
- ▶ Effet de serre
- ▶ cycle du carbone
- ▶ ...

Mais attention : sans composante spatiale, sans tenir compte de tous les phénomènes possibles.

## 2.7 Résolution d'équations différentielles

$$\dot{s} = f(s)$$

Méthode d'Euler :

$$\dot{s}(t) \approx \frac{s(t + \Delta t) - s(t)}{\Delta t}$$

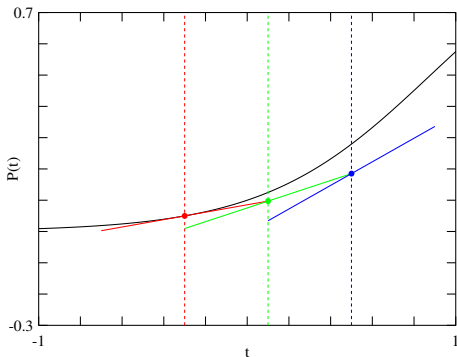
D'où

$$\frac{s(t + \Delta t) - s(t)}{\Delta t} \approx f(s(t))$$

et encore

$$s(t + \Delta t) = s(t) + f(s(t))\Delta t$$

## Méthode d'Euler : illustration graphique

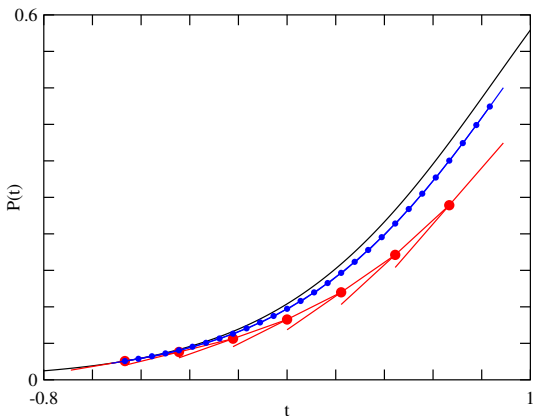


$$\dot{P} = r \left( 1 - \frac{P(t)}{M} \right) P(t) \quad P(t + \Delta t) = P(t) + \Delta t \times \dot{P}$$

$$P(t) = \frac{M}{1 + \frac{M - P_0}{P_0} e^{-rt}}$$

## Méthode d'Euler : précision

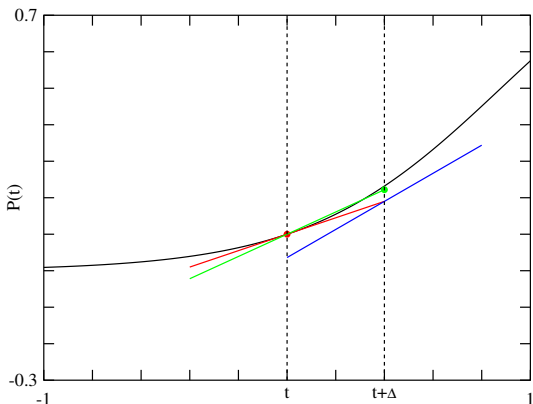
En diminuant  $\Delta t$  on améliore la précision. Mais on fait plus de pas de calcul.



La méthode d'Euler est précise au **première ordre**. Si on divise  $\Delta t$  par deux, l'erreur entre la solution numérique et la solution exacte diminue aussi par deux.

## Méthode de Runge-Kutta d'ordre 2

On cherche une meilleure approximation de la pente de la sécante  $[s(t + \Delta t) - s(t)]/\Delta t$



C'est la moyenne entre la pente  $f(s_0)$  (en rouge) où l'on se trouve et la pente  $f(s_1)$  (en bleu) où l'on serait si on avançait avec le schéma d'Euler. Cette pente est indiquée en vert.

## Méthode de Runge-Kutta d'ordre 2

$$\dot{s} = f(s)$$

Soit  $t_0$  et  $s_0 = s(t_0)$  une condition initial donnée.

Un pas de la méthode de Runge-Kutta est donc

$$\begin{cases} s_1 = s_0 + f(s_0)\Delta t \\ s(t_0 + \Delta t) = s(t_0) + (f(s_0) + f(s_1)) \frac{\Delta t}{2} \end{cases} \quad (3)$$

Puis on pose  $t_0 \rightarrow t_0 + \Delta t$ ,  $s_0 = s(t_0 + \Delta t)$  et on continue à itérer.  
Si on divise  $\Delta t$  par deux, l'erreur est divisée par 4.



## 2.8 Systèmes spatio-temporels

- ▶ Souvent il faut aussi introduire la composante spatiale en plus de la composante temporelle.
- ▶ Pression, température varient dans le temps et l'espace
- ▶ Le nombre d'individus par unité de surface aussi
- ▶ L'état du systèmes est maintenant représenté par une grandeur

$$s(t, x)$$

- ▶ où  $x$  est un point de l'espace en 1, 2 ou 3 dimensions
- ▶ C'est une représentation *Eulérienne* à contraster de la représentation *Lagrangienne*.

## Equation de bilan (ou conservation)

- ▶ Comme avant, la modélisation d'un processus se fera à travers une équation de bilan qui indique comment  $s(t, x)$  varie dans le temps en chaque point  $x$ .
- ▶ En plus de création et destruction de  $s$  en  $x$ , il y a maintenant le flux  $J$  de  $s$  qui arrive et repart de  $x$ .
- ▶ Forme générique :



$$\partial_t s + \operatorname{div} J_s = C - D$$

## Calcul de l'équation de conservation

Bilan de ce qui rentre moins ce qui sort.  $J_x(x, y)dy$  est la quantité qui traverse le segment  $dy$ .

$$\begin{aligned} J_x(x, y)dy - J_x(x + dx, y)dy + J_y(x, y)dx - J_y(x, y + dy)dx \\ = -(\partial_x J_x + \partial_y J_y)dx dy = -\text{div} J \times dx dy \end{aligned}$$

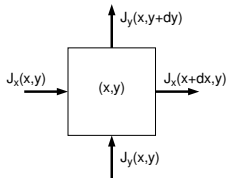
- $s dx dy$  est la quantité de  $s$  dans le volume  $dx dy$ . Donc

$$\partial_t s dx dy = -\text{div} J dx dy$$

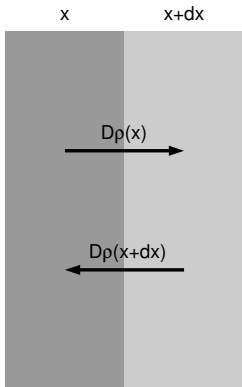
- Ou encore (forme souvent appelée équation de continuité) :

$$\partial_t s + \text{div} J = 0$$

Si la quantité  $s$  est transportée à vitesse  $u(x, y, t)$ , on aura que  $J = s \times u$



## Calcul du courant $J$ : cas de la diffusion



- ▶ Avec une marche aléatoire, la quantité de particules qui se déplacent à gauche et à droite de  $x$  est proportionnelle au nombre de particules présentes en  $x$ .
- ▶ Ce nombre est donc  $D\rho(x)$  où  $D$  est la constante de proportionnalité.
- ▶ Le flux  $J$  de particule à travers la ligne qui sépare  $x$  et  $x + dx$  est donc

$$J = D\rho(x) - D\rho(x+dx) \sim -D \frac{d\rho(x)}{dx}$$

# Equations de Diffusion

L'équation de diffusion pour  $s(t, x)$ , qui contient des dérivées par rapport à  $t$  et à  $x$  s'écrit donc (en 1 dimension) à partir de l'équation de bilan et de l'expression de  $J_x$  :

$$\partial_t s + \partial_x J_x = 0 \quad J_x = -D \partial_x s$$

D'où

$$\partial_t s = D \partial_x^2 s$$

# Equations aux dérivées partielles : différence finie

La méthode de résolution numérique la plus simple est la méthode des **différences finies** : utiliser des différences finies comme approximation des opérateurs différentiels (avec une asymétrie temps – espace).

$$\partial_t s \approx \frac{s(t + \Delta t, x) - s(t, x)}{\Delta t}$$

et

$$\partial_x s \approx \frac{s(t, x + \Delta x) - s(t, x - \Delta x)}{2\Delta x}$$

d'où

$$\partial_x^2 s \approx \frac{s(t, x + \Delta x) - 2s(t, x) + s(t, x - \Delta x))}{\Delta x^2}$$

Ces relations se vérifient par un développement de Taylor

## Solution par différence finie (pour la diffusion)

$$s(t+\Delta t, x) = s(t, x) + \Delta t D \left( \frac{s(t, x + \Delta x) - 2s(t, x) + s(t, x - \Delta x)}{\Delta x^2} \right)$$

qu'on peut résoudre itérativement pour chaque pas de temps (schéma d'Euler). Il faut cependant spécifier des conditions aux bords et des conditions initiales.