

Définitions

(E, d) métrique compact (espace des phases)

$f : E \rightarrow E$ continue (loi d'évolution du système)

$$d_{n,f}(x, y) = \max_{0 \leq j < n} d(f^j(x), f^j(y))$$

$B_{n,f}(x, \varepsilon)$ boule ouverte pour la distance $d_{n,f}$.

Approximation de la dynamique ; ensembles couvrants

$\varepsilon > 0, n \in \mathbb{N}$

$A \subset E$ (n, ε) couvrant :

$$\bigcup_{x \in A} B_{n,f}(x, \varepsilon) = E$$

$r(n, \varepsilon, f)$ cardinal maximal d'un ensemble couvrant.

$$h(\varepsilon, f) \stackrel{def}{=} \limsup_{n \rightarrow \infty} \left(\frac{1}{n} \ln r(n, \varepsilon, f) \right)$$

$$h(f) \stackrel{def}{=} \lim_{\varepsilon \rightarrow 0^+} h(\varepsilon, f)$$

$h(f)$ est l'entropie topologique de l'application f .

Dispersion des états proches ; ensembles séparés

$A \subset E$ (n, ε) séparé :

$$\forall (x, y) \in A^2, x \neq y, d_{n,f}(x, y) \geq \varepsilon$$

$s(n, \varepsilon, f)$ cardinal maximal d'un ensemble séparé.

$$h_{sep}(\varepsilon, f) \stackrel{def}{=} \limsup_{n \rightarrow \infty} \left(\frac{1}{n} \ln s(n, \varepsilon, f) \right)$$

$$h_{sep}(f) \stackrel{def}{=} \lim_{\varepsilon \rightarrow 0^+} h_{sep}(\varepsilon, f)$$

L'encadrement $s(n, 2\varepsilon, f) \leq r(n, \varepsilon, f) \leq s(n, \varepsilon, f)$ montre que $h_{sep} = h$.

Une application est chaotique si son entropie est non nulle

→ démultiplication rapide du nombre d'orbites visibles à chaque itération.

Théorèmes de calcul

Entropie et conjugaison : l'invariance de l'entropie

$$\begin{array}{ccc}
 E & \xrightarrow{f} & E \\
 \Psi \downarrow & & \downarrow \Psi \\
 F & \xrightarrow{g} & F
 \end{array}
 \quad g \circ \Psi = \Psi \circ f$$

Ψ surjective $h(f) \geq h(g)$.

Ψ bijective $h(f) = h(g)$.

Moins facile : si Ψ est surjective et le cardinal des fibres est majoré (« presque conjugaison »), alors $h(f) = h(g)$.

Petits théorèmes

- 1) si $k \in \mathbb{N}$: $h(f^k) = kh(f)$.
- 2) si f est bijective : $h(f^{-1}) = h(f)$.
- 3) si $E = E_1 \cup E_2 \cup \dots \cup E_p$ est une décomposition de E en sous-ensembles fermés stables deux à deux disjoints alors $h(f) = \max_i h(f|_{E_i})$
- 4) si f est expansive ($\exists \delta > 0 \setminus \forall x \neq y, \exists n \in \mathbb{N}, d(f^n(x), f^n(y)) \geq \delta$), et p_n le nombre de points de période n au sens large, alors $h(f) \geq \limsup_{n \rightarrow \infty} \frac{1}{n} p_n$

Exemples

- Les applications ayant une itérée **1-Lipschitziennes** ont une entropie nulle.
- L'**application doublement des angles**, définie sur $S^1 = \mathbb{R} / \mathbb{Z}$ muni de la distance naturelle par $x \mapsto 2x$ est un exemple classique de **sensibilité aux conditions initiales** : son entropie est $\ln 2$.
- L'**application logistique** $f_\mu(x) = \mu x(1 - x)$, dans le cas $\mu > 2 + \sqrt{5}$, a une entropie $\ln 2$ sur l'ensemble $\Lambda = \{x \setminus \forall n \in \mathbb{N}, f_\mu^n(x) \in [0, 1]\}$ (qui est un ensemble de Cantor).
- Dans le cas $\mu = 4$, f_μ est « presque » conjuguée à l'application doublement des angles par $\Psi : S^1 \rightarrow [0, 1], x \mapsto \sin^2(\pi x)$. Ainsi $h(f_4) = \ln 2$.

Tentative de calcul numérique approché de l'entropie

```
(* Constantes *)
let pas=0.0001 and eps=0.1 and nmax=5;;

(* Coeur de l'algorithme *)
let rneps f n eps a b =
  let rec dnf f n x y = if n= -1 then 0. else max (abs_float (x-.y)) (dnf f (n-1) (f x) (f y))
  in
  let rec aux nb x y = if y>b then nb else
    if (dnf f n x y>eps) then aux (nb+1) y (y+.pas) else aux nb x (y+.pas)
  in aux 1 a (a+.pas);;

(* Calcul de la suite des cardinaux des ensembles couvrants *)
let rec entr f n p eps a b =
  if n=p then [] else (rneps f p eps a b)::(entr f n (p+1) eps a b);;

(* [a0, a1, a2, ..] -> [a1/a0, a2/a1, ..] *)
let rec accr = function a::b::q -> (float_of_int b/. float_of_int a)::(accr (b::q)) | _ -> [];;

(* Moyenne de la liste *)
let moyenne l =
  let (s,n)=it_list (fun (somme,nb) x -> somme+.x, nb+1) (0.,0) l
  in s/.(float_of_int n);;

(* Entropie de la fonction f [a,b]->[a,b] *)
let entropie f a b = moyenne (tl (accr (entr f nmax 0 eps a b))));;

(* Fonctions de test *)
let logistique mu x = let z=mu*.x*(1.-.x) in if z<1. then z else 0.;;
let boulanger x = if x>0.5 then 2.*(1.-.x) else 2.*.x;;
let compose f g x = f (g x);;
```

FIG. 1: Le code CAML de la fonction de calcul de l'entropie

Application	Valeur calculée	Valeur théorique
boulanger	1.9688809213	2.0
compose boulanger boulanger	4.0308417489	4.0
logistique 2.	1.15498575499	1.0
logistique 4.	1.9535671351	2.0
logistique 6.	2.06222550767	2.0
fun x->sin (x*.3.1415)	1.98804742781	2.0
fun x->x	1.0	1.0

TAB. 1: Résultats de la fonction entropie