Global Numerics I: Invariant Sets, GAIO

Discrete dynamical systems

$X \subset \mathbb{R}^d$ state space, $f : X \to X$ diffeomorphism.

For $x_0 \in X$ consider the iteration

$$x_{k+1} = f(x_k), \quad k = 0, 1, 2, \ldots$$

Invariant sets

- A set A is called invariant, if $f(A) = A$.
- A point $x \in X$ is called periodic if there is a k such that $f^k(x) = x$. The smallest such k is the period of x.

  $\mathrm{Per}(f)$: set of periodic points of f.

- The sets

$$\begin{aligned} \omega_f(x) &= \{y \in X : \exists (n_i)_i \to +\infty \text{ s.t. } f^{n_i}(x) \to y\} \\ \alpha_f(x) &= \{y \in X : \exists (n_i)_i \to -\infty \text{ s.t. } f^{n_i}(x) \to y\} \end{aligned}$$

  are the ω-limit set and the α-limit set of x.

  $L(f, X) = \overline{\bigcup_{x \in X} \omega_f(x)} \cup \overline{\bigcup_{x \in X} \alpha_f(x)}$.

  Exercise 1 Show that $L(f, X)$ is invariant.

- A point x is called nonwandering if for every neighbourhood U of x there is a $k > 0$ such that $f^k(U) \cap U$ is nonempty.

  $\Omega(f, X)$: set of nonwandering points.

- A sequence $(x_i)_i$ of points $x_i \in X$ is an ε-pseudo-orbit for f, if

$$d(f(x_i), x_{i+1}) < \varepsilon.$$

An ε-pseudo-orbit is ε-pseudo-periodic if there is an n such that $x_i = x_{i+n}$ for all i. A point is chain recurrent if it is element of an ε-pseudo-periodic orbit for all positive ε.

$R(f, X)$: set of chain recurrent points.

- Maximal invariant set:

$$\mathrm{Inv}(f, X) = \{x \in X : f^k(x) \in X \text{ for all } k\}.$$

Lemma 1

$$\mathrm{Per}(f) \subset L(f) \subset \Omega(f) \subset R(f) \subset \mathrm{Inv}(f).$$

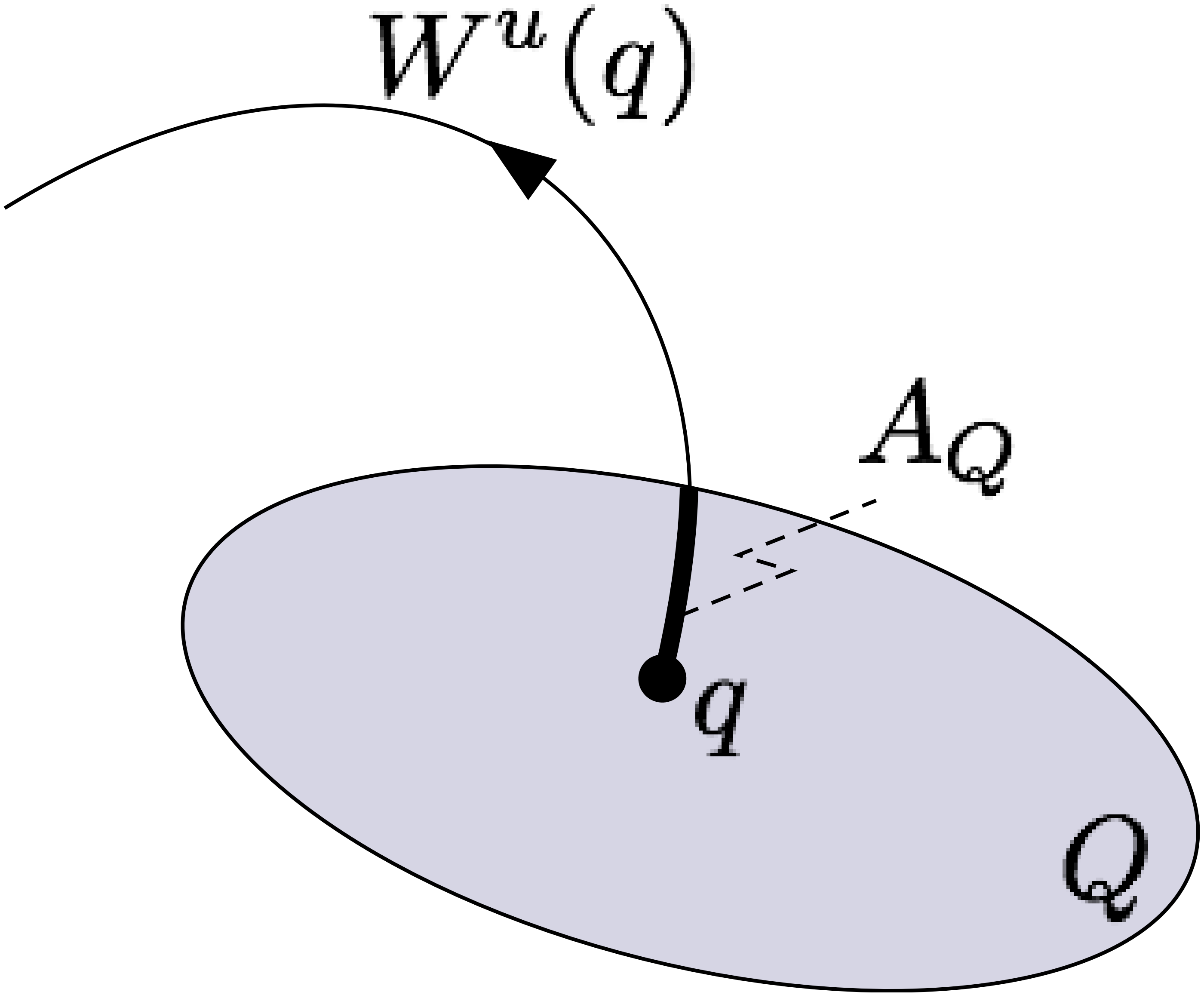
The relative global attractor

$Q \subset X$ compact, the set

$$A_Q = \bigcap_{k=0}^{\infty} f^k(Q)$$

is the global attractor relative to Q.

- A_Q is compact;
- A_Q contains all invariant sets in Q;
- $A_Q \subset f(A_Q)$, but in general $f(A_Q) \not\subset A_Q$.
- A_Q contains (part of) the unstable set of an invariant set in Q;


$W^u(q)$
A_Q
q
Q

A subdivison algorithm [Dellnitz, Hohmann, 97]

Input: $Q \subset X$ compact; Output: sequence $\mathcal{B}_0, \mathcal{B}_1, \ldots$ of finite collections of compact subsets of Q.

Set $\mathcal{B}_0 = \{Q\}$. Given $\mathcal{B}_{k-1}$, $\mathcal{B}_k$ is constructed in two steps:

Subdivision: construct a collection $\hat{\mathcal{B}}_k$ such that

$$\bigcup_{B \in \hat{\mathcal{B}}_k} B = \bigcup_{B \in \mathcal{B}_{k-1}} B \quad \text{and} \quad \operatorname{diam}(\hat{\mathcal{B}}_k) \leq \theta \operatorname{diam}(\mathcal{B}_{k-1})$$

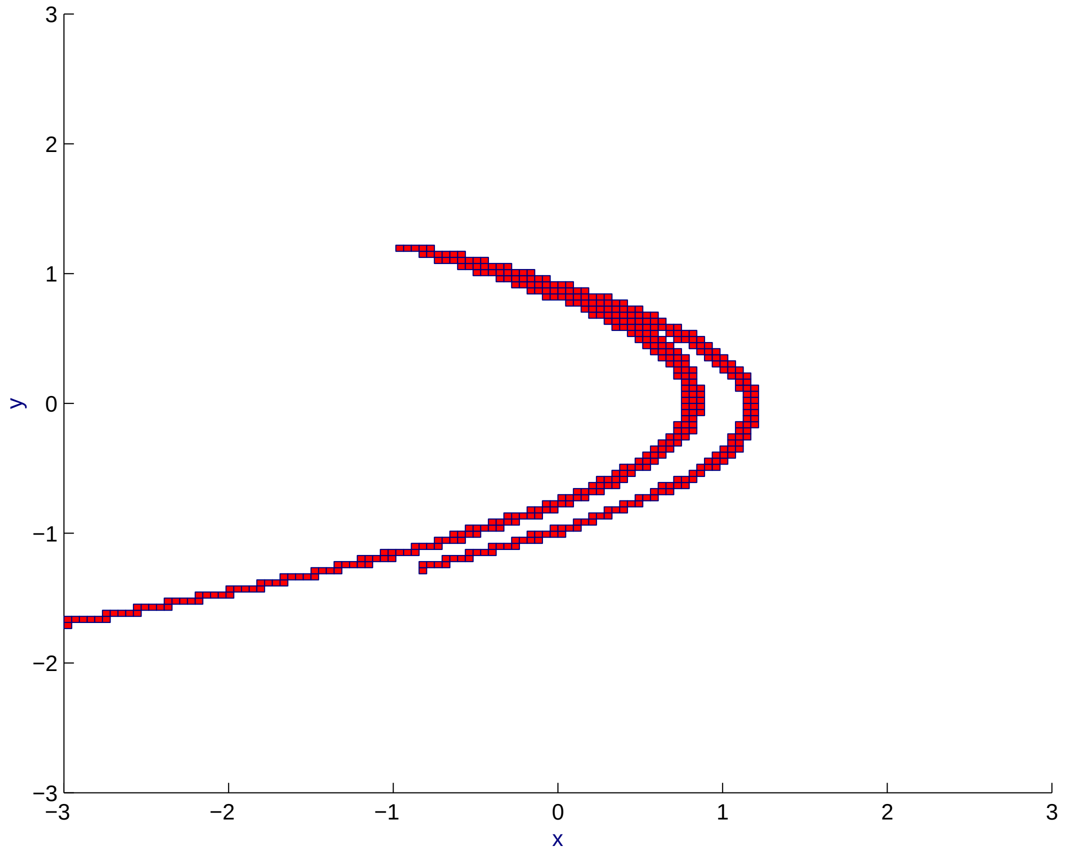
for some $0 < \theta_{min} \leq \theta \leq \theta_{max} < 1$.

Selection: Set

$$\mathcal{B}_k = \{B \in \hat{\mathcal{B}}_k : f(B') \cap B \neq \emptyset \text{ for some } B' \in \hat{B}_k\}.$$

Example: The Hénon-map

$$f(x, y) = (1 - ax^2 + y, bx)$$



Convergence

Let

$$Q_k = \bigcup_{B \in \mathcal{B}_k} B,$$

obviously $Q_{k+1} \subset Q_k$.

Theorem 1 (Dellnitz, Hohmann, 97) *The sets Q_k converge to A_Q in Hausdorff-distance as $k \to \infty$, i.e.*

$$\lim_{k\to\infty} h(Q_k, A_Q) = 0.$$

Equivalently let $Q_\infty = \bigcap_{k=0,1,\ldots} Q_k$, then

$$Q_\infty = A_Q.$$

Proof of Theorem 1 (1)

Lemma 2 *$A_Q \subset Q_k$ for all $k \in \mathbb{N}$.*

Proof:

By definition $A_Q \subset Q = Q_0$. Suppose there is an $x \in A_Q \subset Q_{k-1}$ such that $x \notin Q_k$ for some $k > 0$. Then the set $B \in \hat{\mathcal{B}}_k$ containing x is removed in the selection step, i.e.

$$f(Q_{k-1}) \cap B = \emptyset.$$

Hence, $x \notin f(Q_{k-1})$, but this contradicts the fact that $x \in A_Q \subset f(A_Q) \subset f(Q_{k-1})$.

Proof of Theorem 1 (2)

Lemma 3 *Let $B \subset Q$ be a subset such that*

$$B \subset f(B).$$

Then B is contained in the global attractor A_Q relative to Q.

<u>Proof:</u>

We know that $B \subset f^j(B)$ for all $j \geq 0$. Therefore,

$$B \subset \bigcap_{j \geq 0} f^j(B) \subset \bigcap_{j \geq 0} f^j(Q) = A_Q.$$

Proof of Theorem 1 (3)

Lemma 4 $Q_\infty \subset f(Q_\infty)$.

Proof: Suppose that there is a point $x \in Q_\infty$, such that $x \notin f(Q_\infty)$. Since Q is compact there is $\delta > 0$ such that

$$d(x, f(Q_\infty)) > \delta.$$

Hence there is an integer $K > 0$ such that

$$d(x, f(Q_k)) > \delta/2 \quad \text{for } k > K.$$

For each k let $B_k(x) \subset \mathcal{B}_k$ be a set containing x. By continuity there is an $\ell > K$ such that

$$B_\ell(x) \cap f(Q_\ell) = \emptyset.$$

By construction of the algorithm this is impossible. $\square$

Hyperbolic sets

A compact invariant set A is called hyperbolic, if for every $x \in A$ there is a decomposition $T_xX = E^s_x \oplus E^u_x$ such that

(i) $Df_x E^{s,u}_x = E^{s,u}_{f(x)}$;

(ii) There are $C > 0$, $\lambda < 1$, such that

$$\begin{aligned} \|Df^k_x v\| &\leq C\lambda^k \|v\| \quad \text{for } v \in E^s_x, k \in \mathbb{N}, \\ \|Df^{-k}_x v\| &\leq C\lambda^k \|v\| \quad \text{for } v \in E^u_x, k \in \mathbb{N}; \end{aligned}$$

(iii) $E^{s,u}_x$ depend continuously on x.

Speed of convergence

Theorem 2 (Dellnitz, Hohmann, 97) *Let A_Q be the global attractor relative to Q with respect to the diffeomorphism f^q, $q \in \mathbb{N}$. If A_Q is attractive and hyperbolic and [...] then*

$$h(A_Q, Q_k) \leq \mathrm{diam}(\mathcal{B}_k)(1 + \alpha + \cdots + \alpha^k),$$

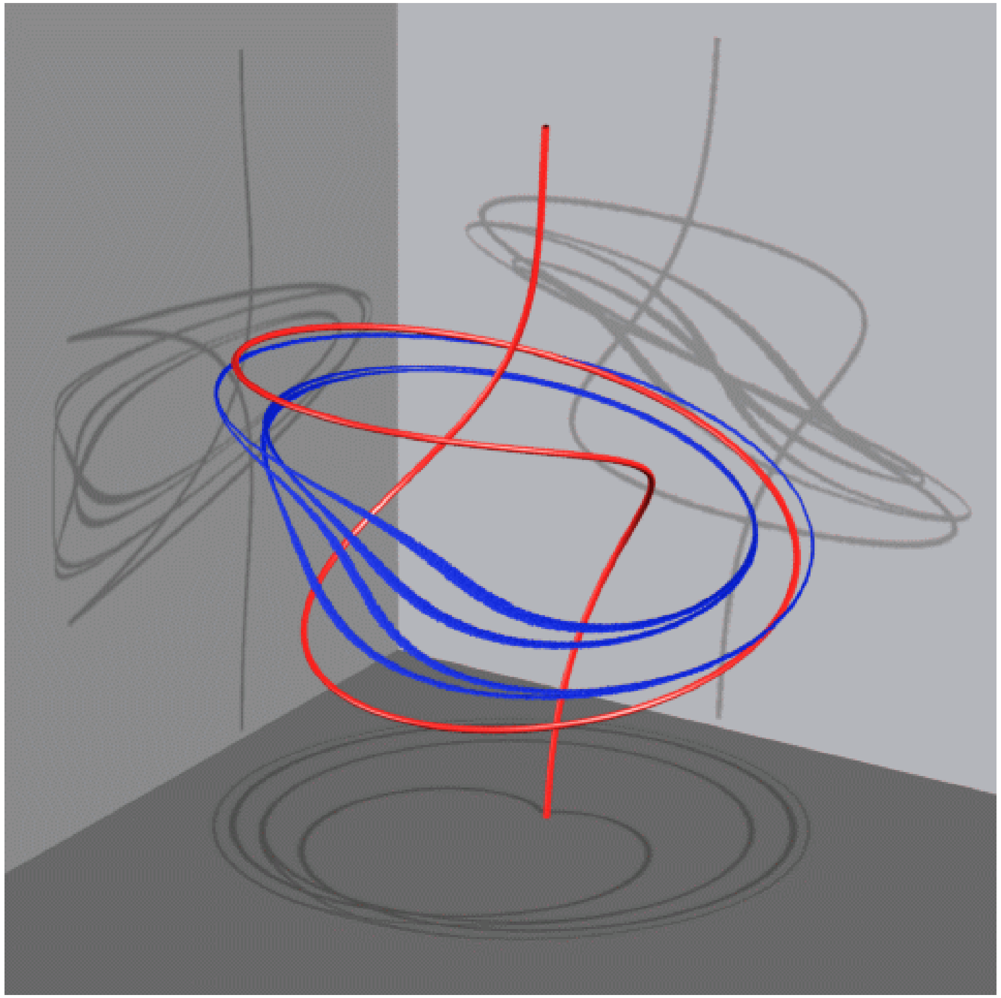
where

$$\alpha = C \frac{\lambda^q}{\theta_{min}}.$$

If $\alpha < 1$ then

$$h(A_Q, Q_k) \leq \mathrm{diam}(\mathcal{B}_k) \frac{1}{1 - \alpha}.$$

Example: A "knotted" flow



(joint work with M. Dellnitz, R. Strzodka and M. Rumpf, visualization by GRAPE)

The stable manifold theorem

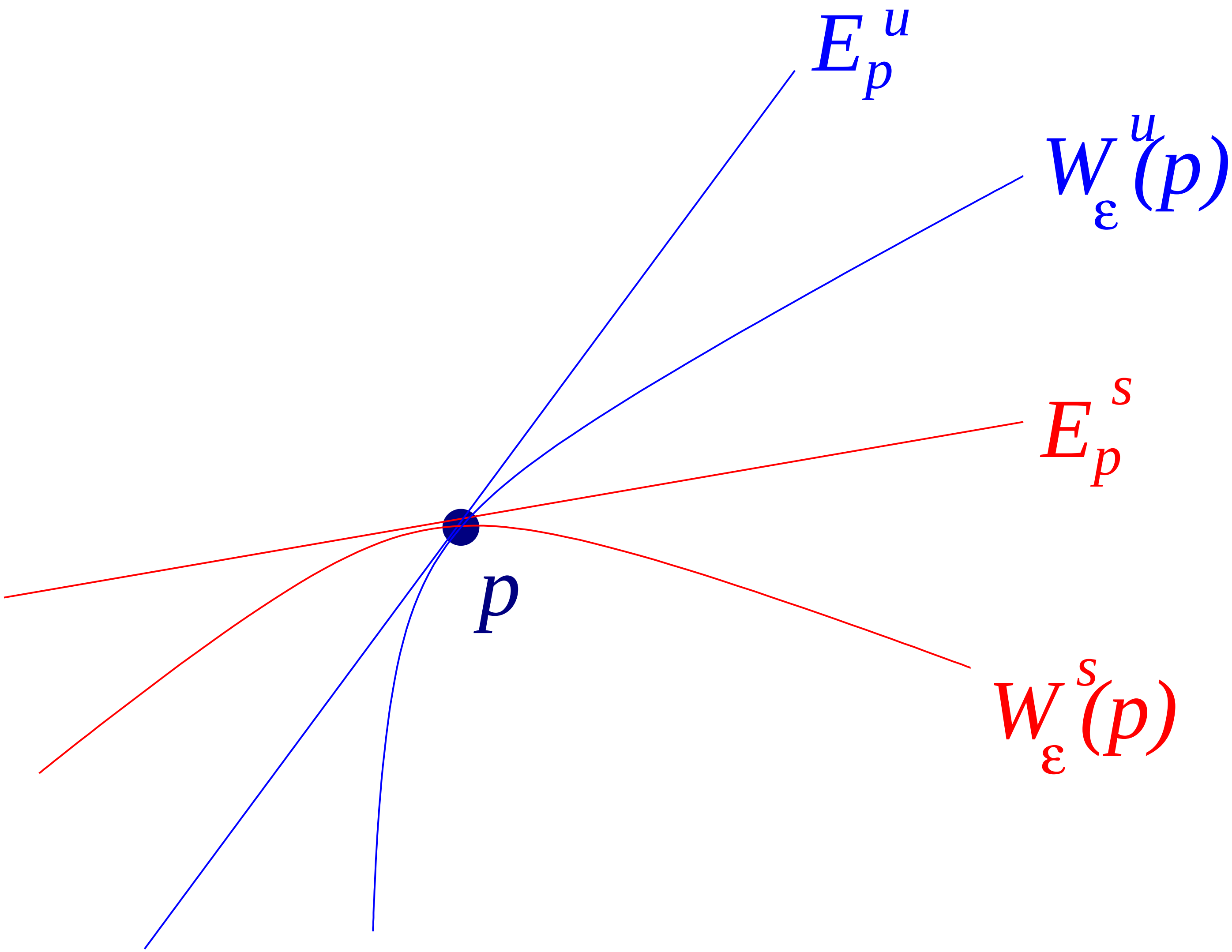
For a fixed point $p = f(p)$ let

$$W^s_\varepsilon(p) = \{x : \|f^k(x) - p\| \leq \varepsilon \text{ for } k \in \mathbb{N}\}$$

Let E^s_p be the subspace spanned by the eigenspaces to eigenvalues λ with $|\lambda| < 1$ of Df_p.

Theorem 3 *If p is hyperbolic then for ε small enough $W^s_\varepsilon(p)$ is a smooth submanifold of X. $W^s_\varepsilon(p)$ is tangent to E^s_p in p and of the same dimension.*

$W^s_\varepsilon(p)$: local stable manifold of p.


E_p^u
$W_\varepsilon^u(p)$
E_p^s
p
$W_\varepsilon^s(p)$

Global (un-)stable manifolds

The sets

$$W^u(p) = \bigcup_{k\in\mathbb{N}} f^k(W^u_\varepsilon(p)), \quad W^s(p) = \bigcup_{k\in\mathbb{N}} f^{-k}(W^s_\varepsilon(p))$$

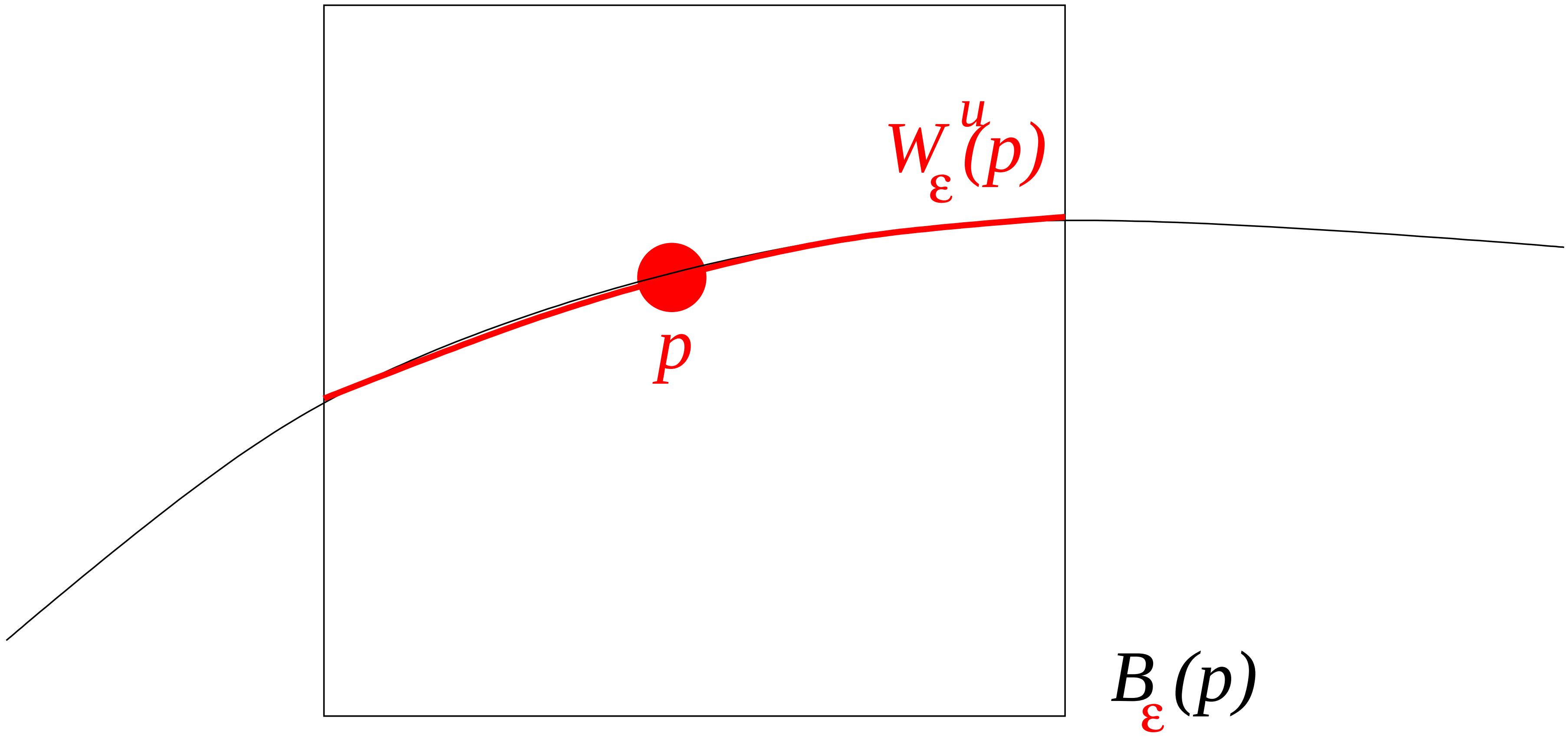
are the (global) unstable/stable manifold of p.

Computing the local unstable manifold

p hyperbolic fixed point, $B_\varepsilon(p)$ ε-neighbourhood of p.

Observation: If ε is small enough, then

$$A_{B_\varepsilon(p)} = W^u_\varepsilon(p).$$


$W^u_\varepsilon(p)$
p
$B_\varepsilon(p)$

The continuation algorithm [Dellnitz, Hohmann, 96]

Input: $Q \subset X$ compact, $p \in Q$ hyperbolic fixed point.

Output: collection $\mathcal{B}$, covering part of $W^u(p)$.

$\mathcal{P}_0 < \mathcal{P}_1 < \ldots$: nested sequence of finite partitions of Q.

$\mathcal{P}_\ell(p)$: set of $\mathcal{P}_\ell$ containing p.

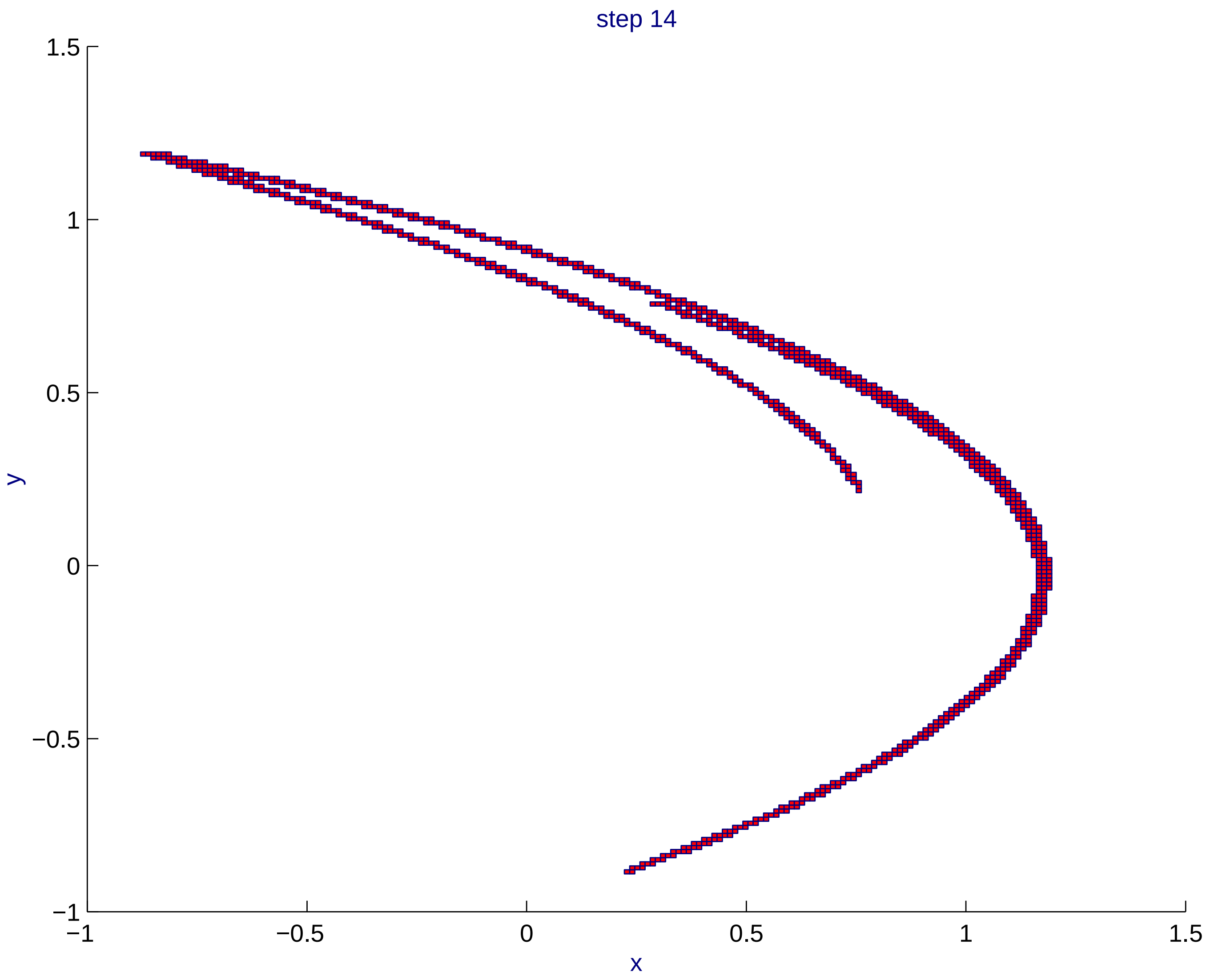
Initialization: For ℓ big enough approximate $W^u_{loc}(p) \cap \mathcal{P}_\ell(p)$ by subdivision, yielding a collection $\mathcal{B}_k^{(0)} \subset \mathcal{P}_{\ell+k}$.

Continuation: From $\mathcal{B}_k^{(j-1)}$ compute

$$\mathcal{B}_k^{(j)} = \{B \in \mathcal{P}_{\ell+k} : f(B') \cap B \neq \emptyset \text{ for some } B' \in \mathcal{B}_k^{(j-1)}\}.$$

until $\mathcal{B}_k^{(j)} = \mathcal{B}_k^{(j-1)}$. Set $\mathcal{B} = \mathcal{B}_k^{(j)}$.

Example: The Hénon map



Convergence

Set $W_0 = W^u_{loc}(p) \cap \mathcal{P}_\ell(p)$ and for $j = 0, 1, 2, \ldots$

$$W_{j+1} = f(W_j) \cap Q.$$

Denote

$$Q_k^{(j)} = \bigcup_{B \in \mathcal{B}_k^{(j)}} B.$$

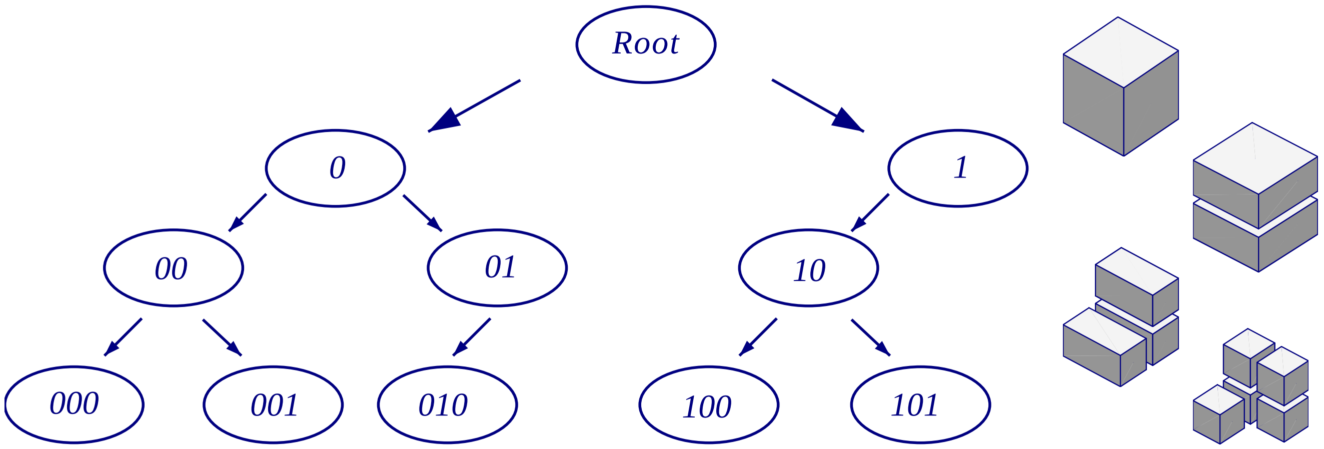
Lemma 5 (Dellnitz, Hohmann, 96) *The set $Q_k^{(j)}$ is a covering of W_j for all $j, k = 0, 1, \ldots$. Moreover, for fixed j, $Q_k^{(j)}$ converges to W_j in Hausdorff distance if the number k of subdivision steps in the initialzation goes to infinity.*

Storage of the set collections

Collections: set of boxes $B = B(c,r) = \{x : |x_i - c_i| \leq r_i, i = 1, \ldots, n\}$.

Subdivison: by bisection w.r.t to one coordinate direction.

Storage of the boxes: binary tree:



The multivalued box map

Recall the selection step of the subdivision algorithm:

$$\mathcal{B}_k = \{B' \in \hat{\mathcal{B}}_k : f(B) \cap B' \neq \emptyset \text{ for some } B \in \hat{B}_k\}.$$

Define the image of a box $B \in \mathcal{B}$ in some collection $\mathcal{B}$ as

$$\mathcal{F}_{\mathcal{B}}(B) = \{B' \in \mathcal{B} : f(B) \cap B' \neq \emptyset\}.$$

Then

$$\mathcal{B}_k = \bigcup_{B \in \hat{\mathcal{B}}_k} \mathcal{F}_{\hat{\mathcal{B}}_k}(B).$$

Discretization of the selection step

Question: How to compute $\mathcal{F}_{\mathcal{B}}(B)$?

Heuristic method: Choose a finite set $T \subset B$ of test points and compute

$$\tilde{\mathcal{F}}_{\mathcal{B}}(B) = \{B' \in \mathcal{B} : f(T) \cap B' \neq \emptyset\}.$$

Evidently

$$\tilde{\mathcal{F}}_{\mathcal{B}}(B) \subset \mathcal{F}_{\mathcal{B}}(B),$$

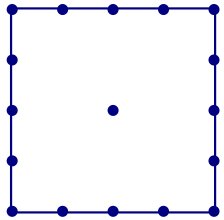
but possibly

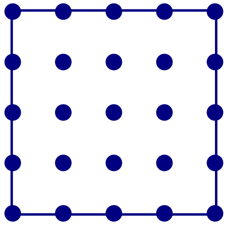
$$\tilde{\mathcal{F}}_{\mathcal{B}}(B) \neq \mathcal{F}_{\mathcal{B}}(B),$$

i.e. one is missing some boxes of the image.

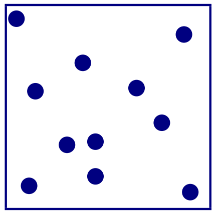
Choice of testpoints

In a low dimensional phase space ($d \leq 3$):

- on the edges of the box (+ center): 

- on a uniform grid within the box: 

In a higher dimensional phase space:

- randomly (from a uniform distribution): 

Invoking GAIO

GAIO: “Global Analysis of Invariant Objects”

```
shell> setenv GAIODIR  /software/gaio-2.1.3

shell> matlab -nojvm


                    < M A T L A B >
        Copyright 1984-2003 The MathWorks, Inc.
    Version 6.5.1.200223 Release 13 (Service Pack 1)
                      Aug 22 2003


>> addpath ~/software/gaio-2.1.3/matlab/
```

Defining a new model: the C-file

```
char *typ = "map";
char *name = "The Henon map";
int   dim  = 2;
int   paramDim = 2;
char *paramNames[] = { "a", "b" };
double a = 1.3, b = 0.2;
double c[2] = { 0, 0 };
double r[2] = { 3, 3 };
double tFinal = 1;

void rhs(double *x, double *u, double *y)
{
  y[0] = 1 - a*x[0]*x[0] + x[1]/5;
  y[1] = b*x[0]*5;
}
```

Compilation and loading

Compile the file

```
shell> cc -c henon.c
```

Make it a shared object (platform specific)

```
shell> ld -shared -o henon.so henon.o
```

Load it within GAIO

```
>> henon = Model('henon')
```

Parameters of a model

```
>> henon.info

Model at 1245c0:
  handle = ef4201c4
  filename = henon
  tmpname = /var/tmp/aaa0MiM9Y
  name = The Henon map
  type = map
  dim = 2
  uDim = 0
  center = 0 0
  radius = 3 3
  tFinal = 1
  number of parameters = 2
  parameters:  a = 1.3  b = 0.2
  rhs() at ef4426e8
  fixed_point() at ef442814: x = 0.621772767199582 0.621772767199582
```

Querying and changing parameters

```
>> henon.center

ans =

0

0

>> henon.a

ans =

1.3

>> henon.a = 1.4
```

Loading an integrator

```
>> map = Integrator('Map')
>> map.info
```

```
integrator at 815e260:
  handle = 815e078
  filename = Map
  tmpname = /tmp/08332aaa
  name = Map
  task at 0
  tFinal = 0
  h = 1
  h_min = 1
  h_max = 1
  eps = 1e-06
  count = 0
  Step() at 405ac980
  Init() at 0
```

Available integrators

```
>> map = Integrator('Map')
```

- `Map`: discrete mapping
- `Newton`: Newton's method (i.e. one or several Newton steps)
- `Euler`: the Euler scheme (constant stepsize)
- `RungeKutta4`: fourth order Runge-Kutta scheme (const. stepsize)
- `DormandPrince853`: embedded Runge-Kutta scheme of order 8(5,3) with adaptive stepsize control
- `MidpointRule`: simple implicit scheme
- `Gauss6`: implicit scheme of higher order

Using an integrator

```
>> map.model = henon
>> map.eval([1; 1])

1.0000  -0.2000
1.0000   1.0000

>> orbit = map.eval([1; 1], 100)
>> orbit

 ...
  Columns 99 through 101

   -0.0292    1.1768   -0.9448
    0.8902   -0.0292    1.1768
```

Creating a tree

```
>> tree = Tree([0;0], [3;3])
```

respectively

```
>> tree = Tree(henon.center, henon.radius)
```

Initially the tree consist only of the root box:

```
>> tree.depth
```

```
ans =
0
```

Tree methods (1)

```
>> tree.set_flags('all', to_be_subdivided)

>> tree.subdivide()

>> tree.depth

ans =

1

>> tree.count(-1)

ans =

2
```

Loading test points

```
>> edges = Points('Edges', 2, 20)

>> edges.info

points at 0x813f620:
  filename = Edges
  tmpname = /tmp/130573aa
  name = Edges
  dim = 2
  noOfPoints = 17
  info = 20
  Set() at 0x4079ac90
  Get() at 0x4079acd0
```

Available test points

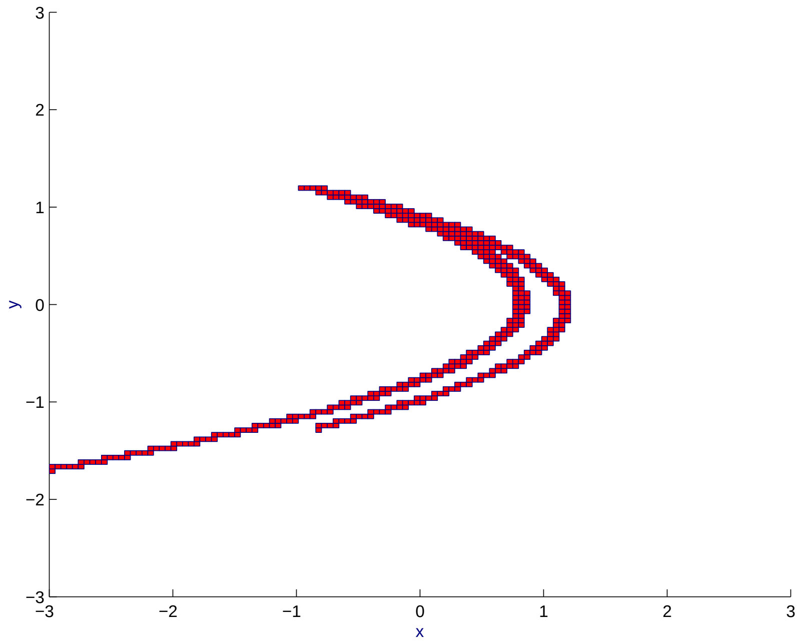
- `Edges`: edges + center
- `Grid`: uniform grid
- `MonteCarlo`: randomly (uniform distribution)
- `Vertices`: vertices
- `Center`: only the center
- `Data`: specified by a file
- `Lipschitz`: grid, depending on Lipschitz-estimate of the map

The subdivision algorithm

```
for i=1:no_of_steps
  tree.set_flags('all', to_be_subdivided);
  tree.subdivide;
  b = tree.first_box(-1);
  while (~isempty(b))
    center = b(1:d);
    radius = b(d+1:2*d);
    p = domain_points.get(center, radius, integrator, radius);
    fp = integrator.map(p);
    tree.set_flags(fp, hit);
    b = tree.next_box(-1);
  end
  tree.remove(hit);
end
```

Output

```
>> plotb(tree)
>> tree.boxes(-1)
>> b = tree.boxes(-1)
>> show2(b',2, 'r')
```



Inserting specific boxes into the tree

```
>> x = henon.fixed_point

ans =
0.6064
0.6064

>> tree.insert(x, 16)
>> tree.depth

ans =
16
>> tree.count(16)
ans =
1
```

The continuation algorithm

```
for i=1:no_of_steps
  tree.change_flags('all', inserted, to_be_expanded);
  b = tree.first_box(depth);
  while (~isempty(b))
    center = b(1:d);
    radius = b(d+1:2*d);
    flags = b(2*d+1);
    if (bitand(flags, to_be_expanded))
      p = domain_points.get(center, radius, integrator, radius);
      fp = integrator.map(p);
      tree.insert(fp, depth, inserted, none);
    end
    b = tree.next_box(depth);
  end
  tree.unset_flags('all', to_be_expanded);
end
```