A new approach for guaranteed integration of nonlinear ODE using positive-invariant set theory

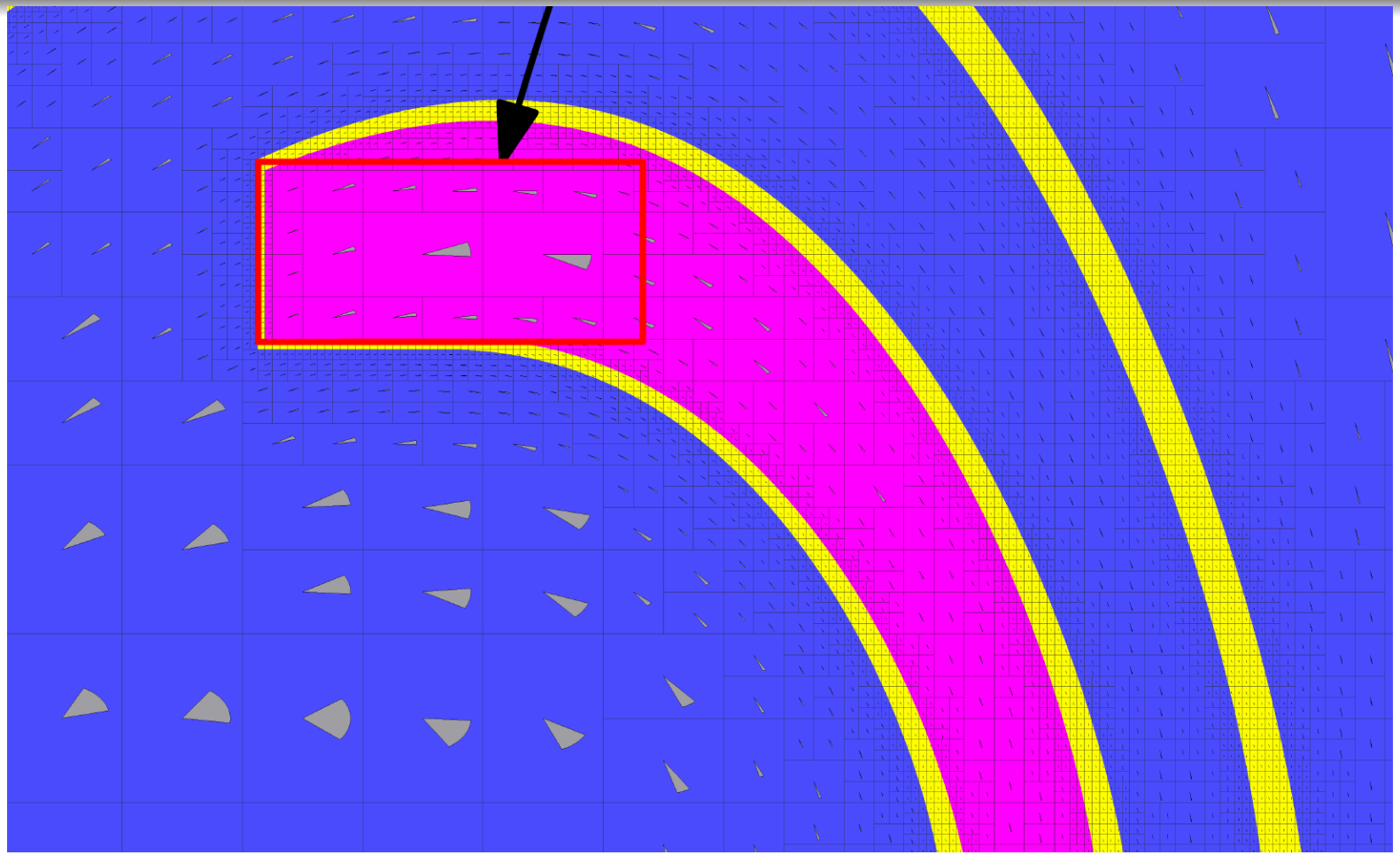
L. Jaulin, T. Le Mézo, B. Zerr, D. Massé, Lab-STICC, ENSTA-Bretagne
ModeliScale, Paris, 2017 December 5

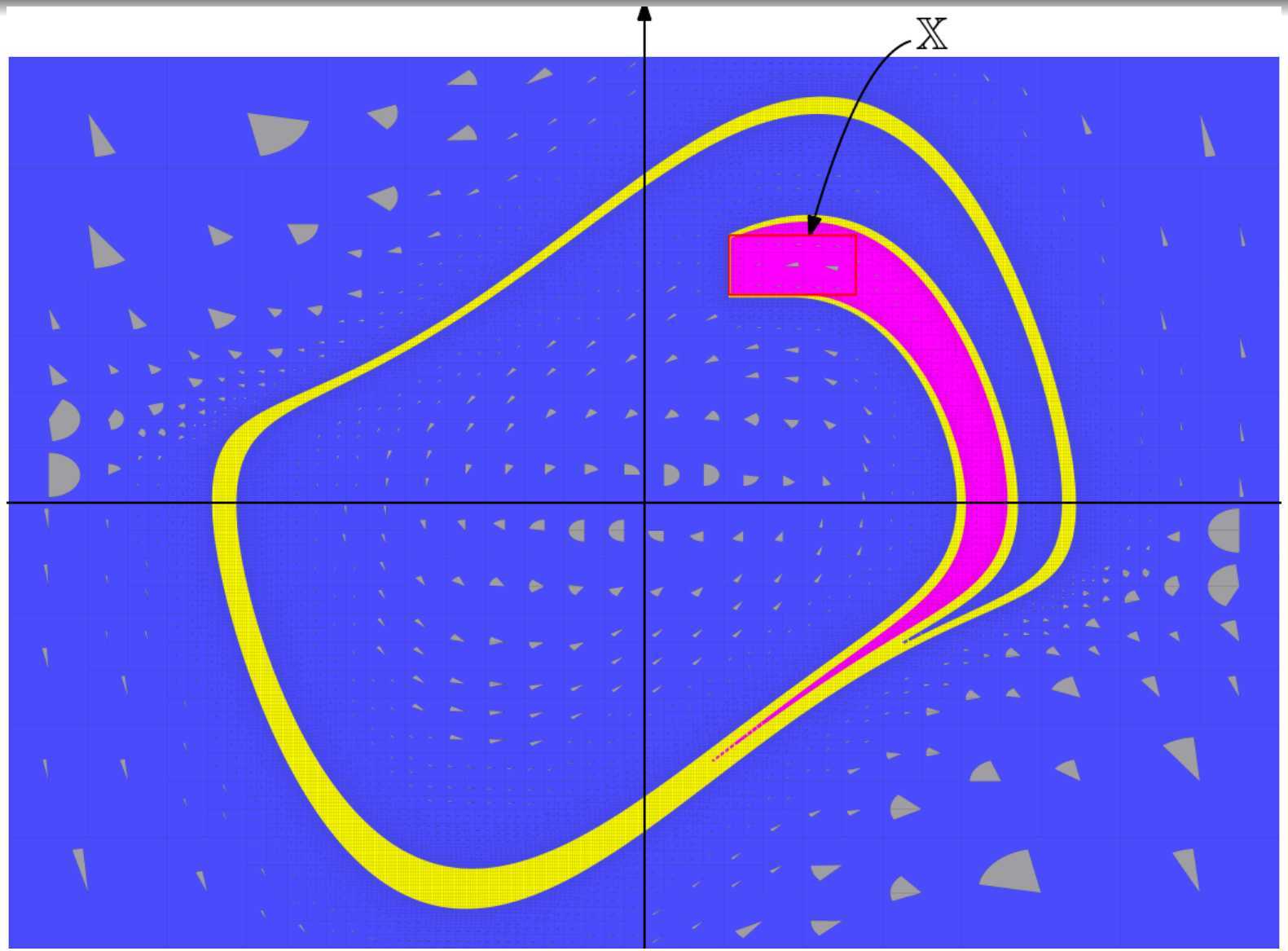
Guaranteed integration

We consider a state equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$.

Let φ be the flow map.
The *forward reach set* of $\mathbb{X} \subset \mathbb{R}^n$ is:

$$\overrightarrow{\mathbb{T}}_{\varphi}(\mathbb{X}) = \{\mathbf{x} \mid \exists \mathbf{x}_0 \in \mathbb{X}, \exists t \geq 0, \mathbf{x} = \varphi(t, \mathbf{x}_0)\}.$$




X

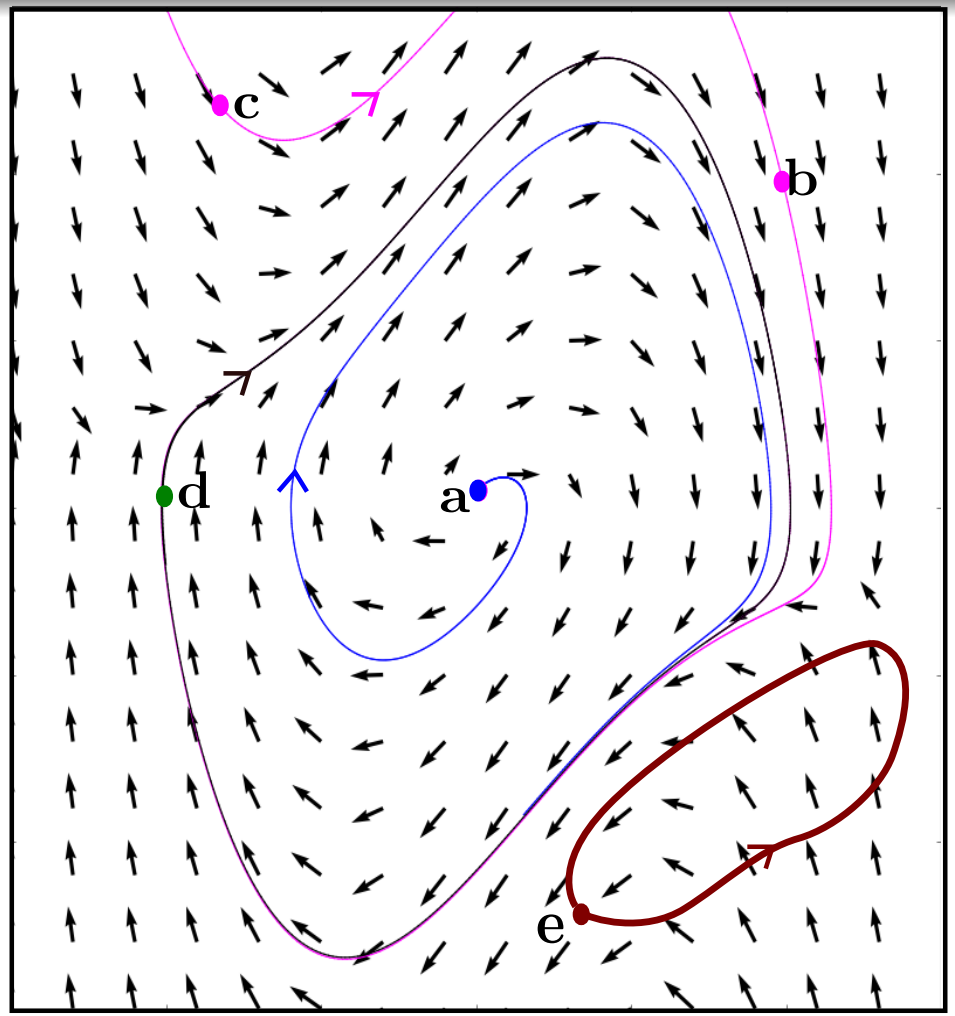
The vector field corresponds to the Van der Pol system

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \left(1 - x_1^2\right) \cdot x_2 - x_1 \end{cases}$$

Largest positive invariant sets

Example: Consider

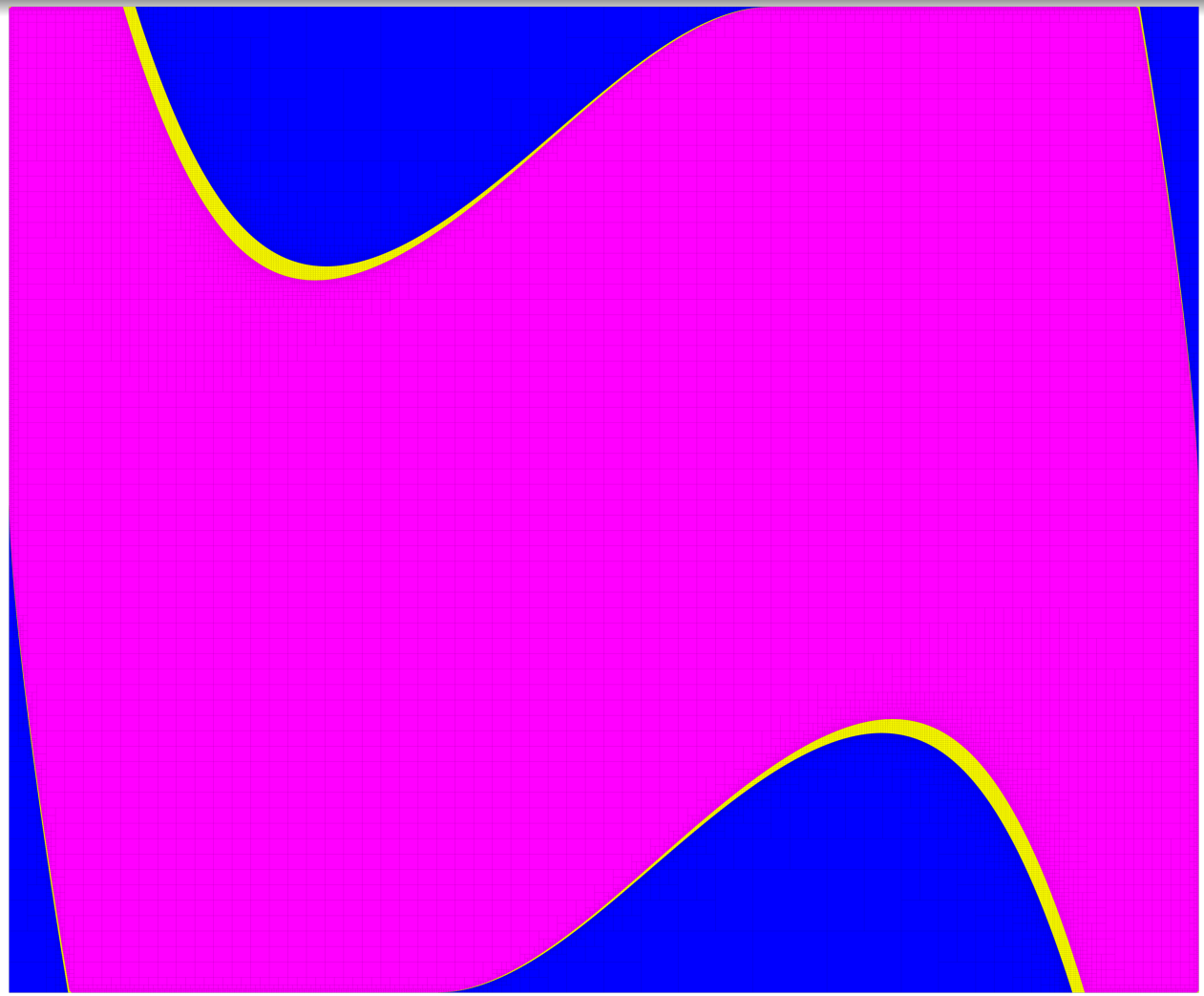
$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \left(1 - x_1^2\right) \cdot x_2 - x_1 \end{cases}$$



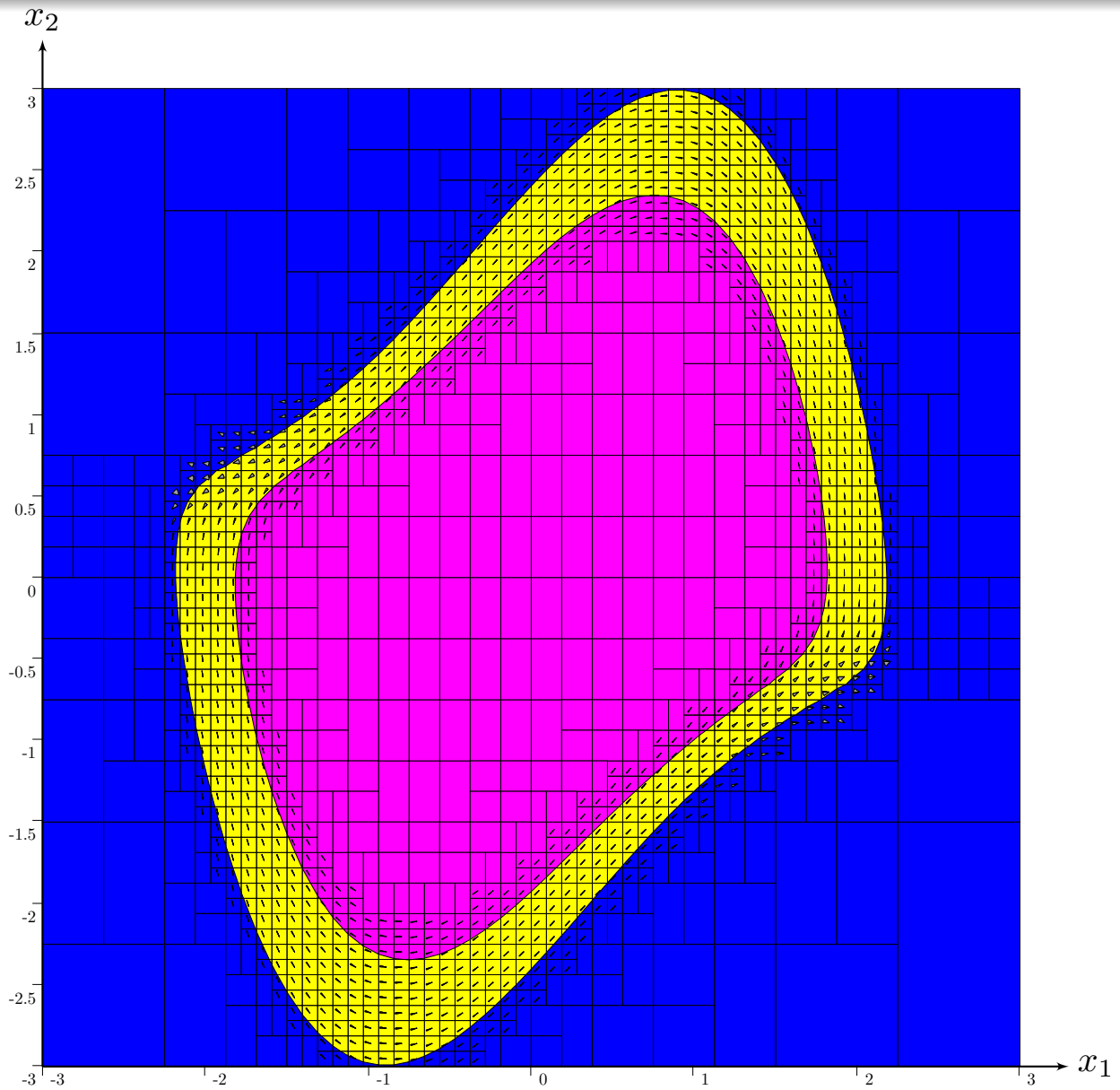
Positive invariant sets : $\mathbb{I}_{\varphi}^{+}(\mathbb{X})$ with $\mathbb{X} = [-4,4] \times [-4,4]$.

The *largest positive invariant set* in $\mathbb{X} \subset \mathbb{R}^n$ is:

$$\mathbb{I}_{\varphi}^{+}(\mathbb{X}) = \{\mathbf{x}_0 \mid \forall t \geq 0, \varphi(t, \mathbf{x}_0) \in \mathbb{X}\}.$$



Positive invariant sets : $\mathbb{I}_{\varphi}^{+}(\mathbb{X})$ with $\mathbb{X} = [-4,4] \times [-4,4]$.

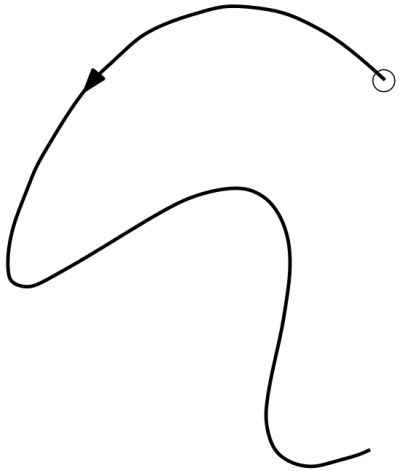

x_2
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3
-2
-1
0
1
2
3
x_1

Positive and negative invariant sets : $\mathbb{I}^+_{\varphi^{-1}}(\mathbb{X}) \cap \mathbb{I}^+_{\varphi}(\mathbb{X})$

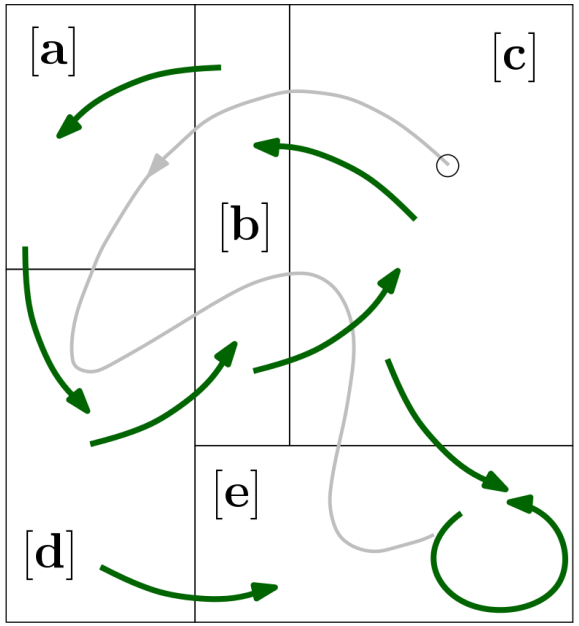
$$
\begin{aligned}
\overrightarrow{\mathbb{T}}_{\varphi}(\mathbb{X}) &= \{\mathbf{x} \mid \exists \mathbf{x}_0 \in \mathbb{X}, \exists t \geq 0, \mathbf{x} = \varphi(t, \mathbf{x}_0)\} \\
&= \{\mathbf{x} \mid \exists \mathbf{x}_0 \in \mathbb{X}, \exists t \geq 0, \mathbf{x}_0 = \varphi^{-1}(t, \mathbf{x})\} \\
&= \overline{\{\mathbf{x} \mid \exists t \geq 0, \varphi^{-1}(t, \mathbf{x}) \in \mathbb{X}\}} \\
&= \{\mathbf{x} \mid \forall t \geq 0, \varphi^{-1}(t, \mathbf{x}) \in \overline{\mathbb{X}}\} \\
&= \overline{\mathbb{I}^{+}_{\varphi^{-1}}(\overline{\mathbb{X}})}
\end{aligned}
$$

Mazes

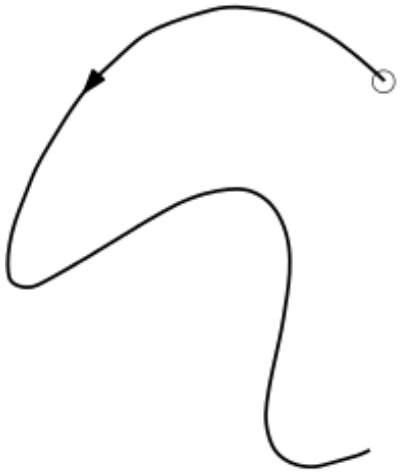
A maze [2][1] is a set of trajectories.



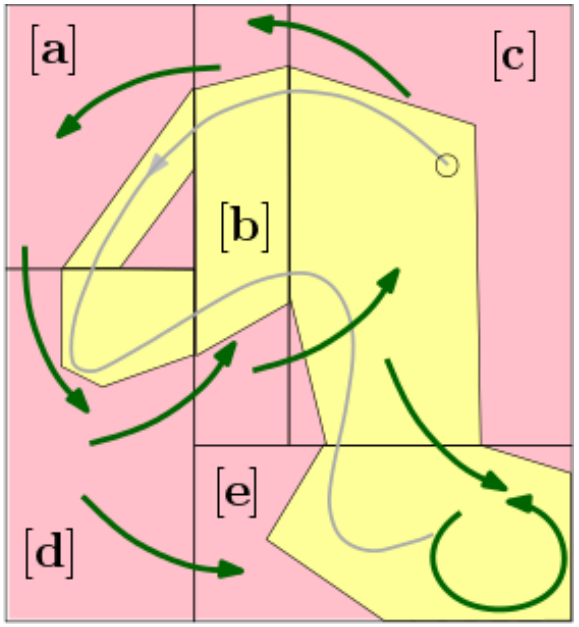
$\in$



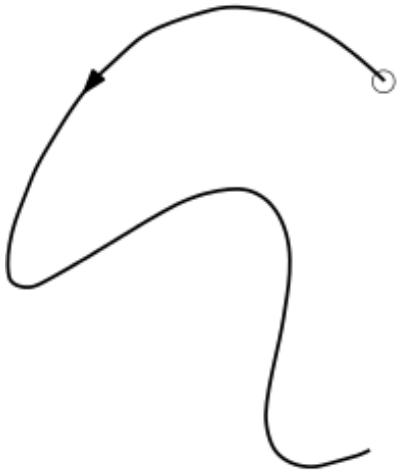
Mazes can be made more accurate by adding polygones.



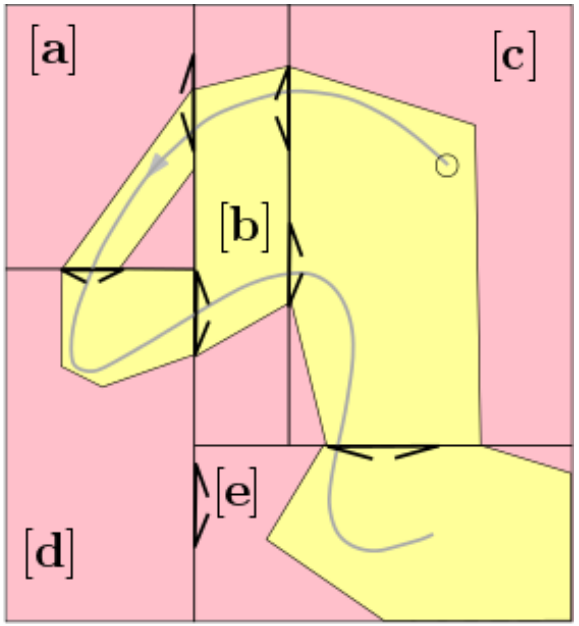
$\in$



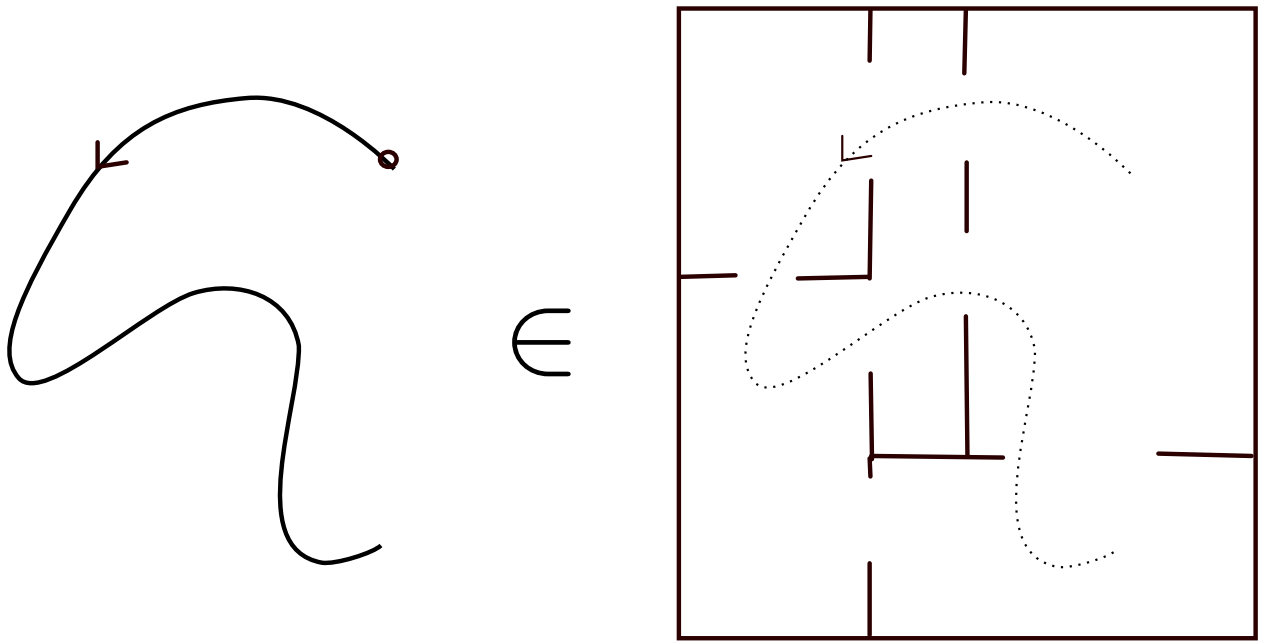
Or using doors instead of a graph



$\in$



Here, we use bi-directional doors



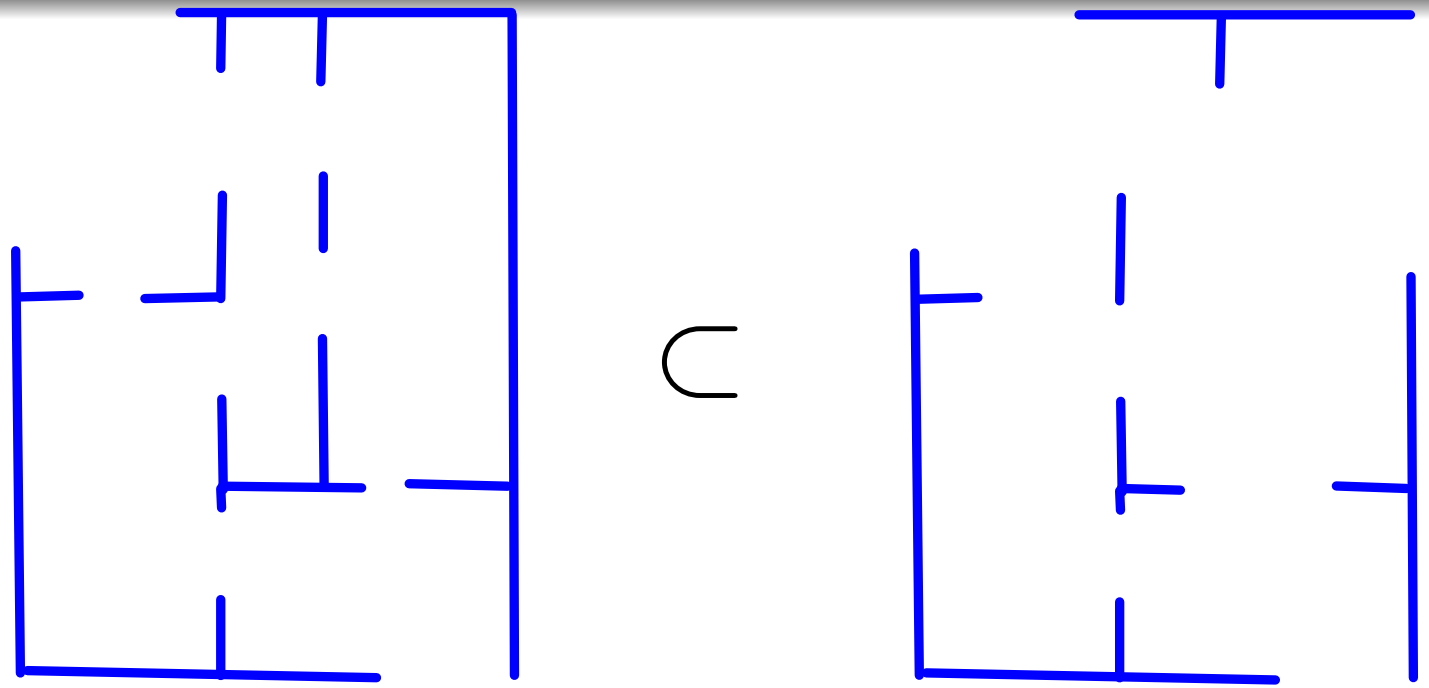
The trajectory $\mathbf{x}(\cdot)$ belongs to the maze $[\mathbf{x}](\cdot)$

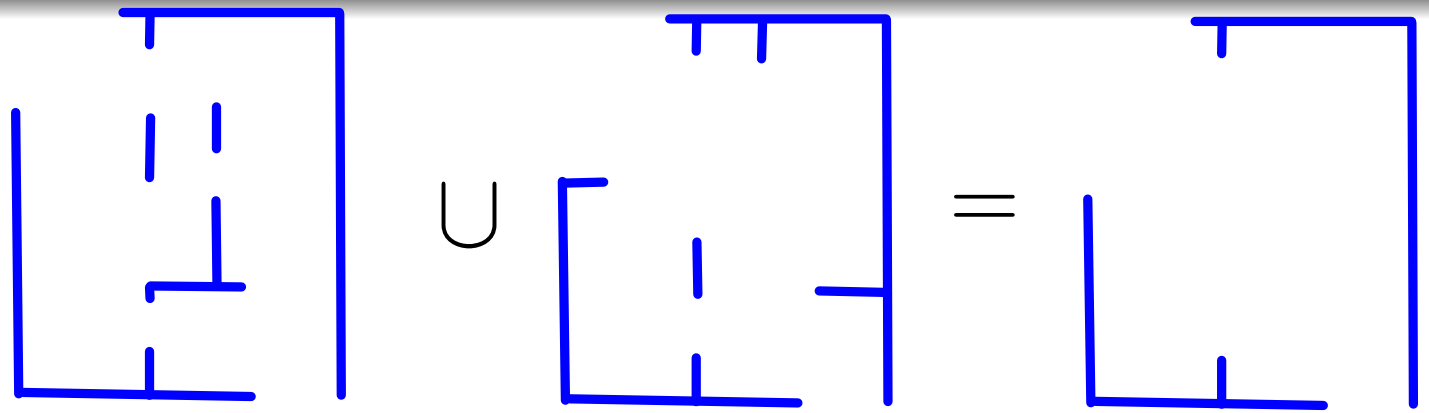
Here, a **maze** $\mathscr{L}$ is composed of

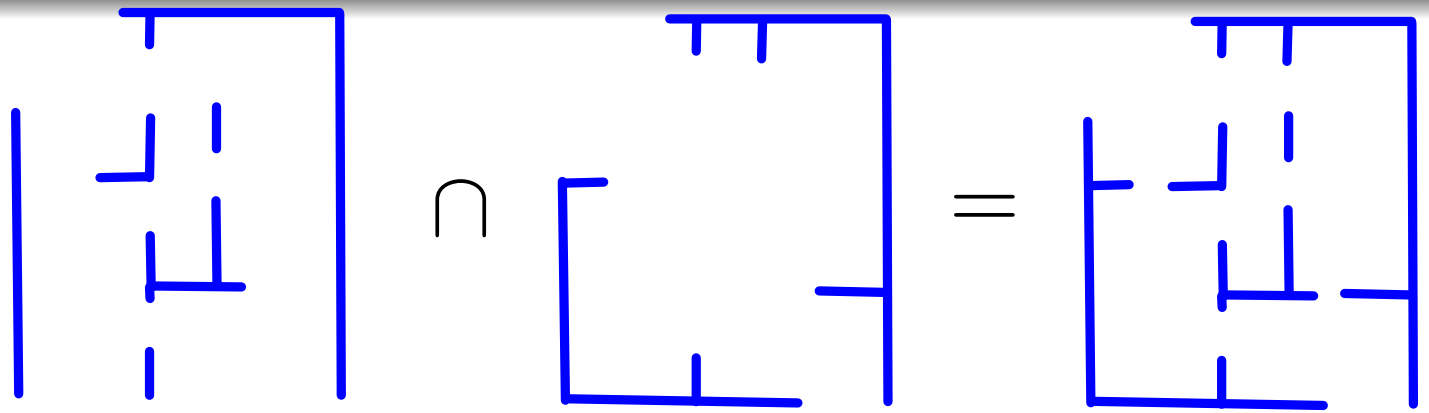
- A paving $\mathscr{P}$
- Doors between adjacent boxes

The set of mazes forms a lattice with respect to $\subset$.
$\mathscr{L}_a \subset \mathscr{L}_b$ means :

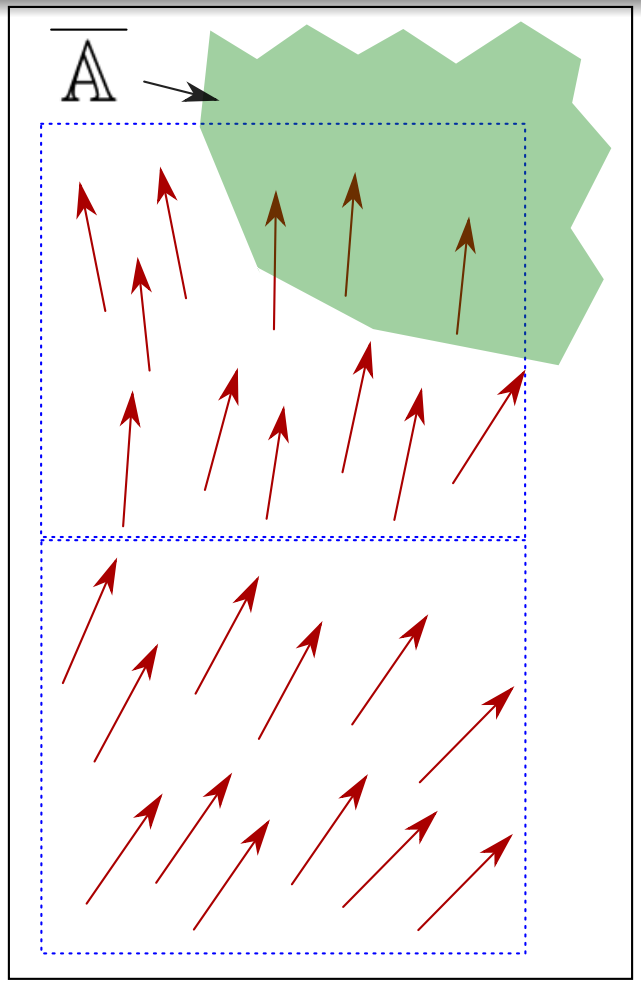
- the boxes of $\mathscr{L}_a$ are subboxes of the boxes of $\mathscr{L}_b$.
- The doors of $\mathscr{L}_a$ are thinner than those of $\mathscr{L}_b$.

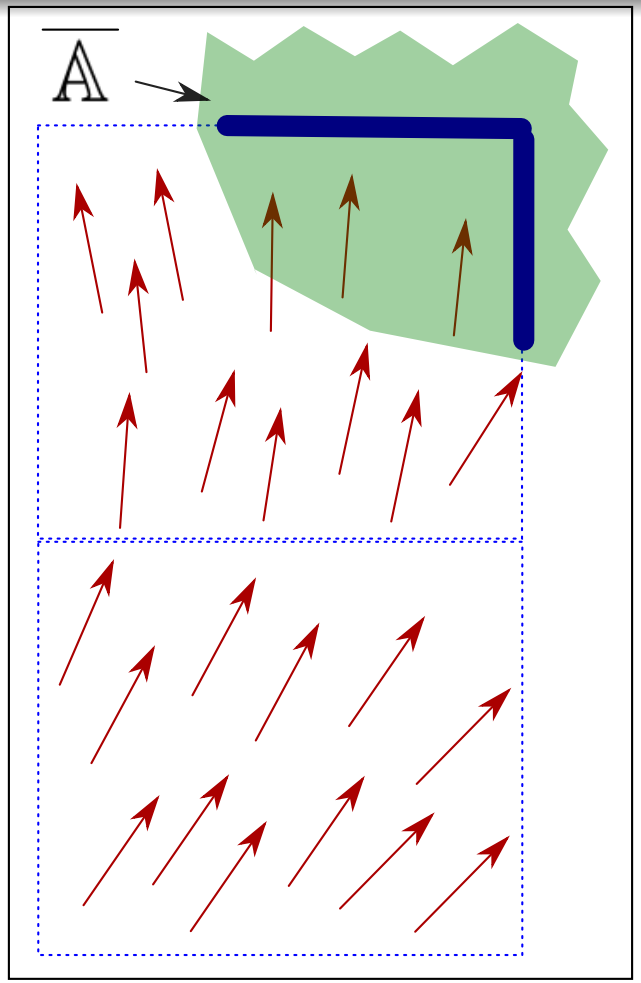


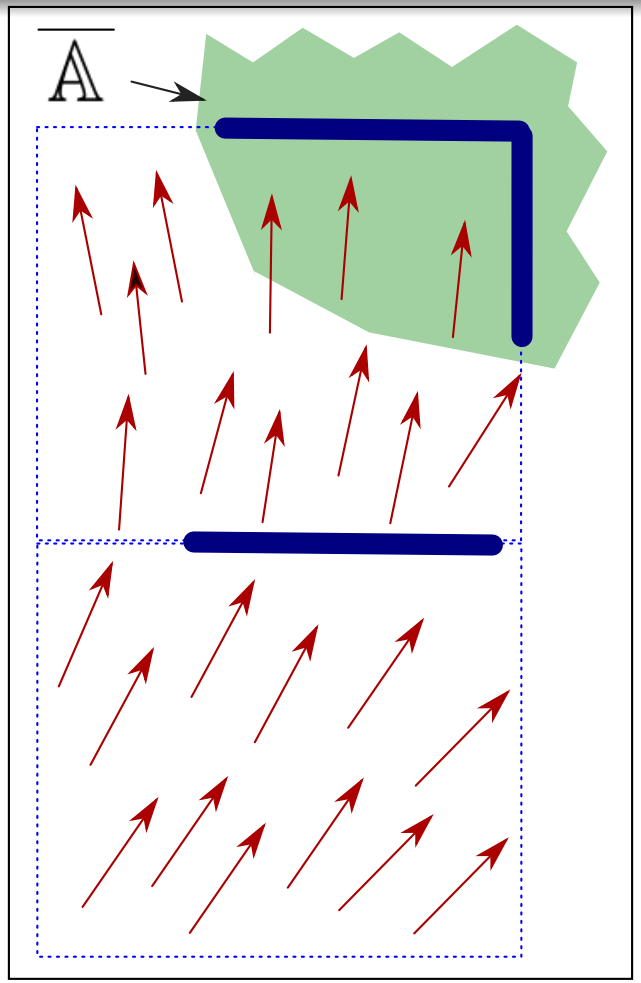


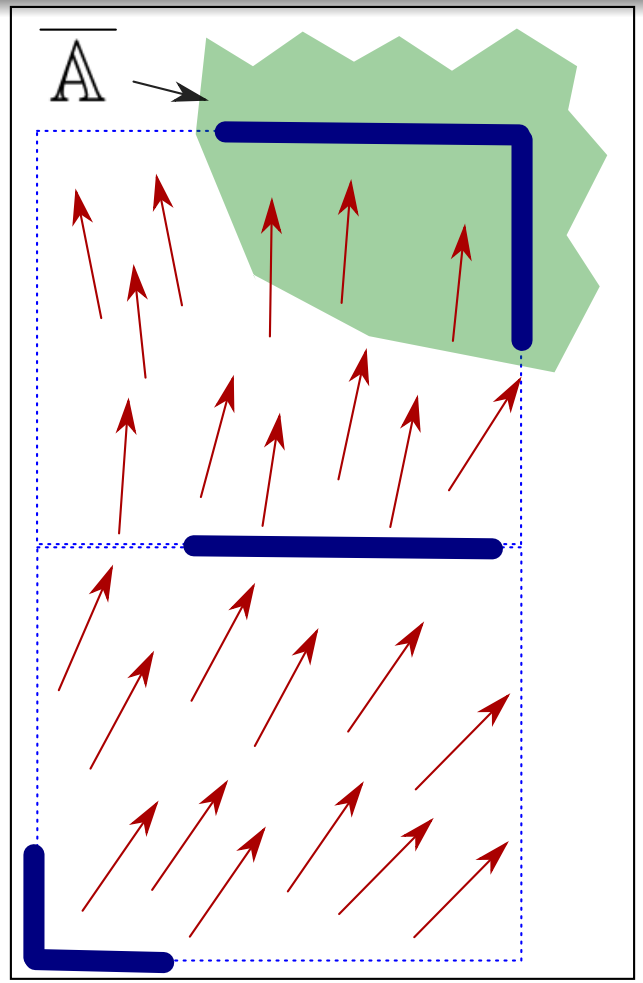


Contract trajectories that certainly go to $\overline{\mathbb{A}}$

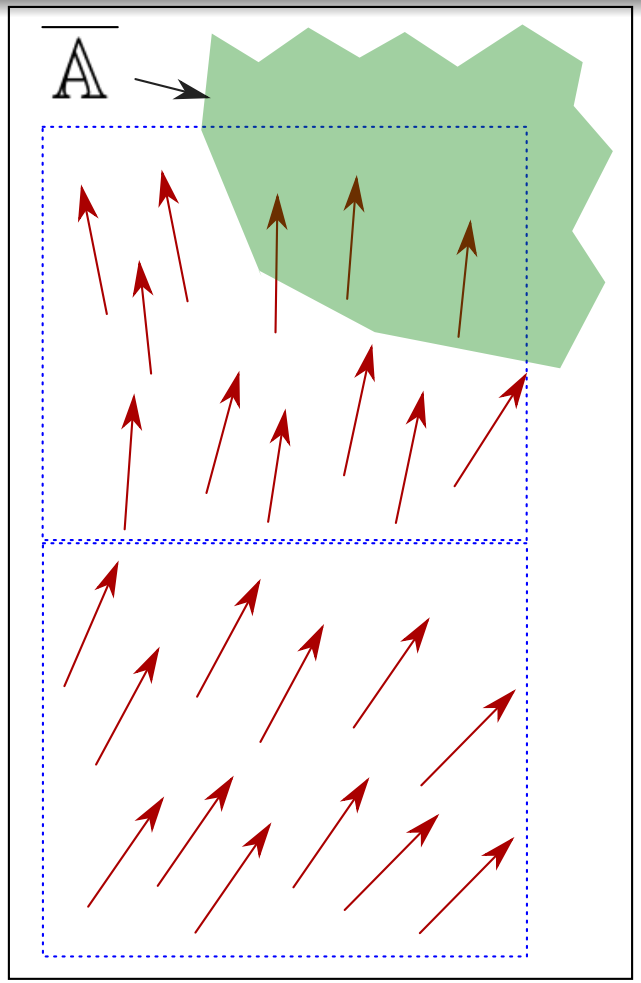

$\overline{\mathbb{A}}$

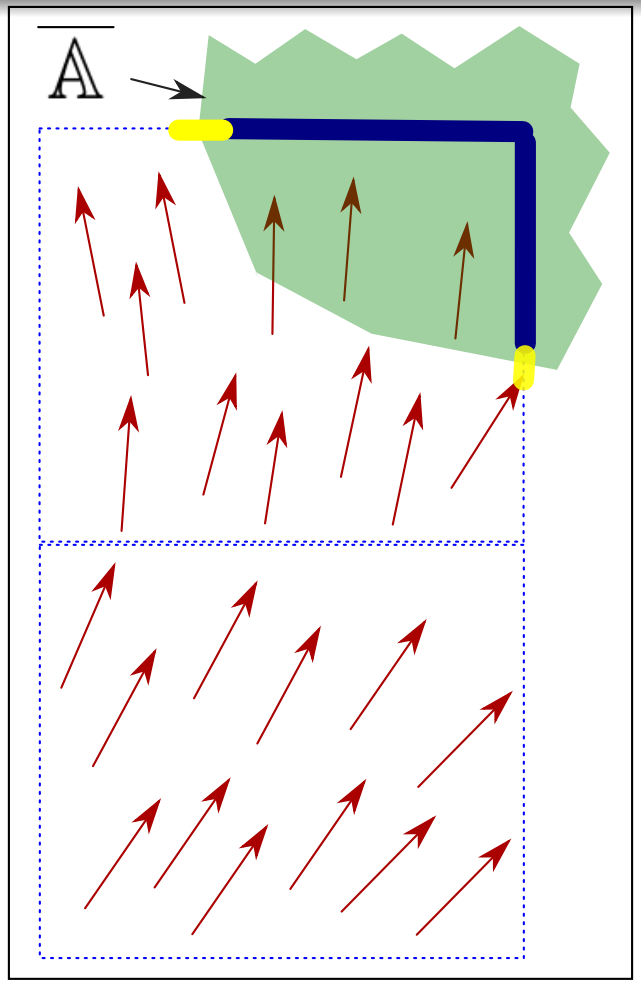

A

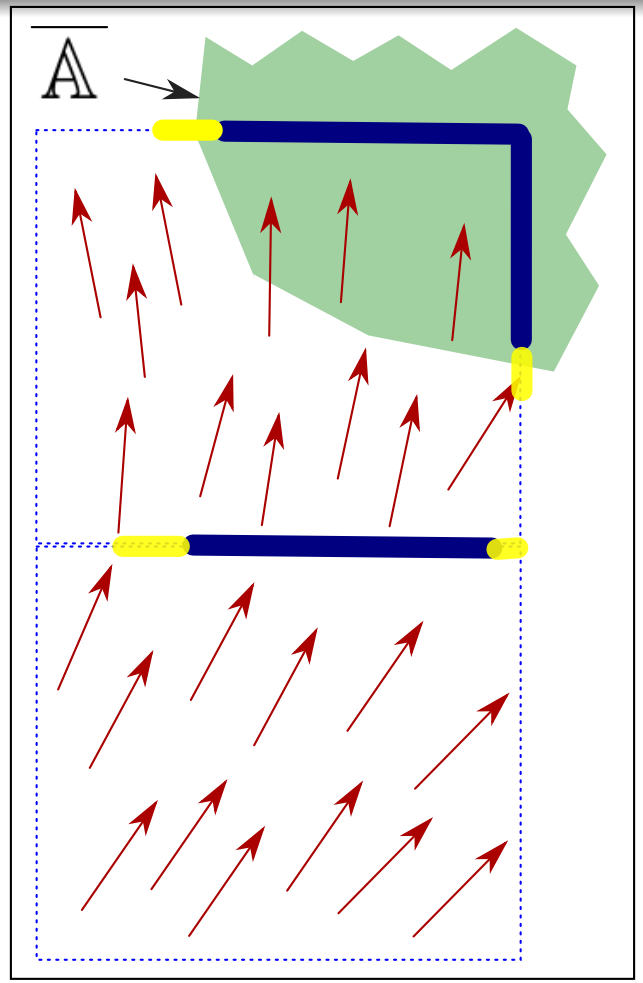

A

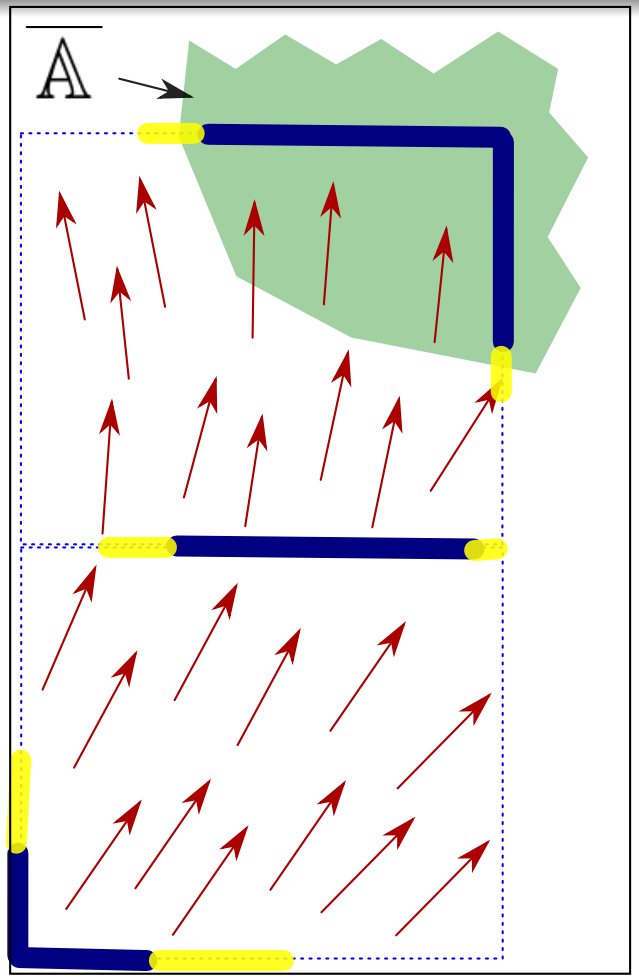

A

Contract trajectories that possibly go to $\overline{\mathbb{A}}$

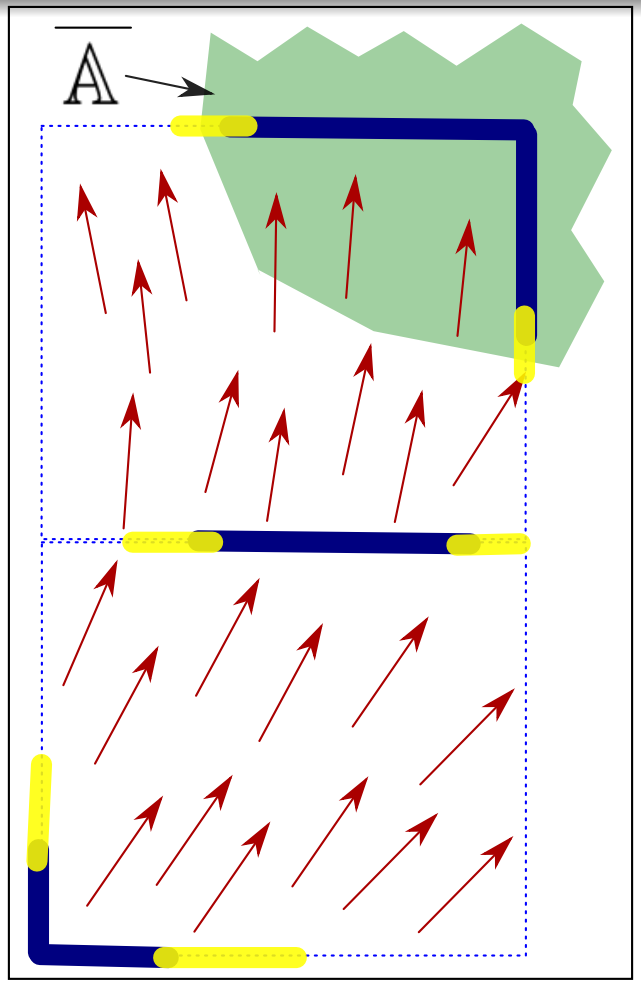

$\overline{\mathbb{A}}$

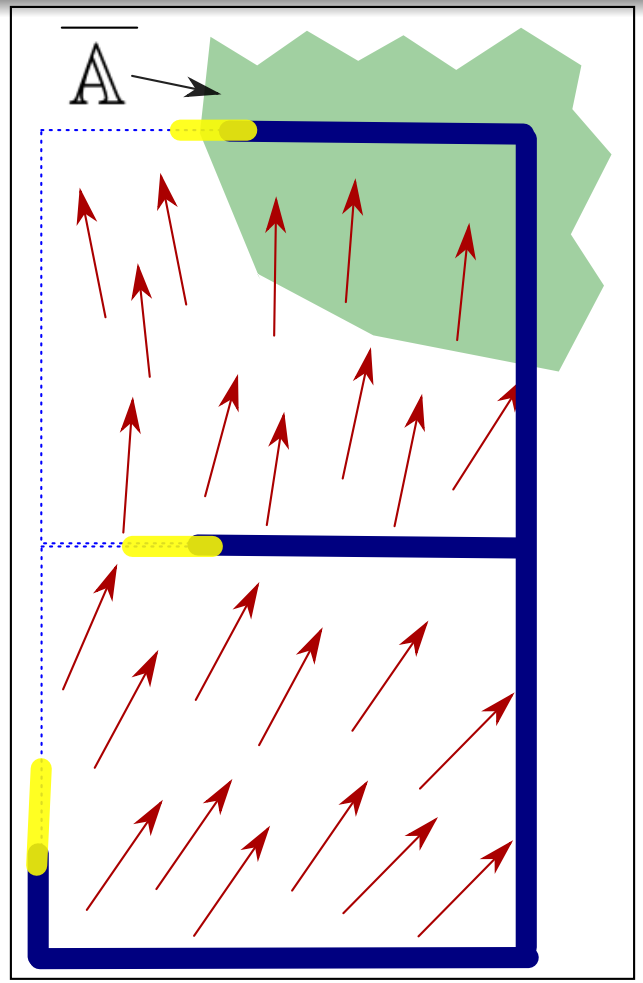

A

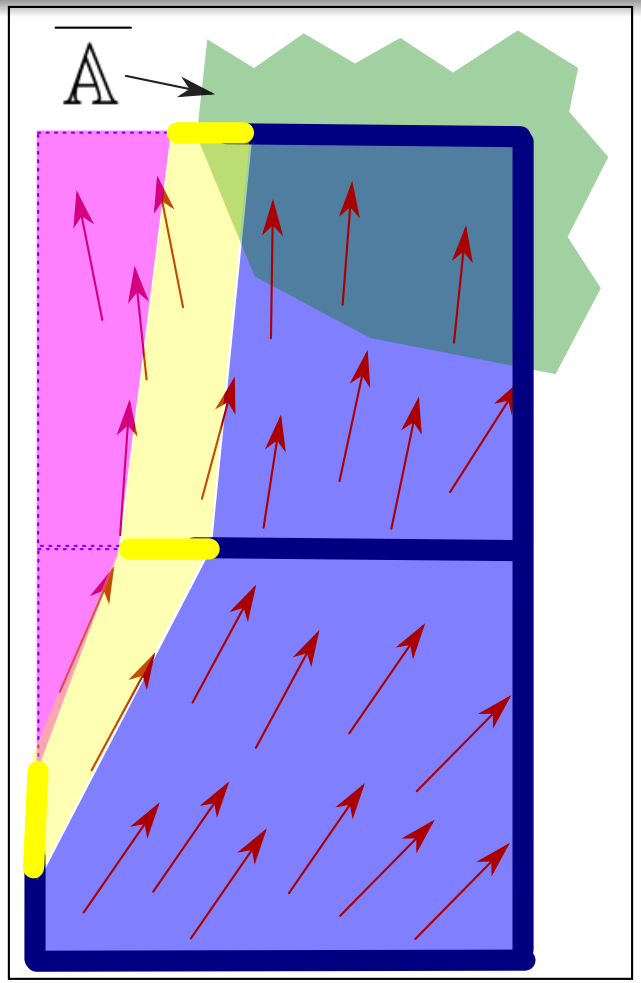

A


A

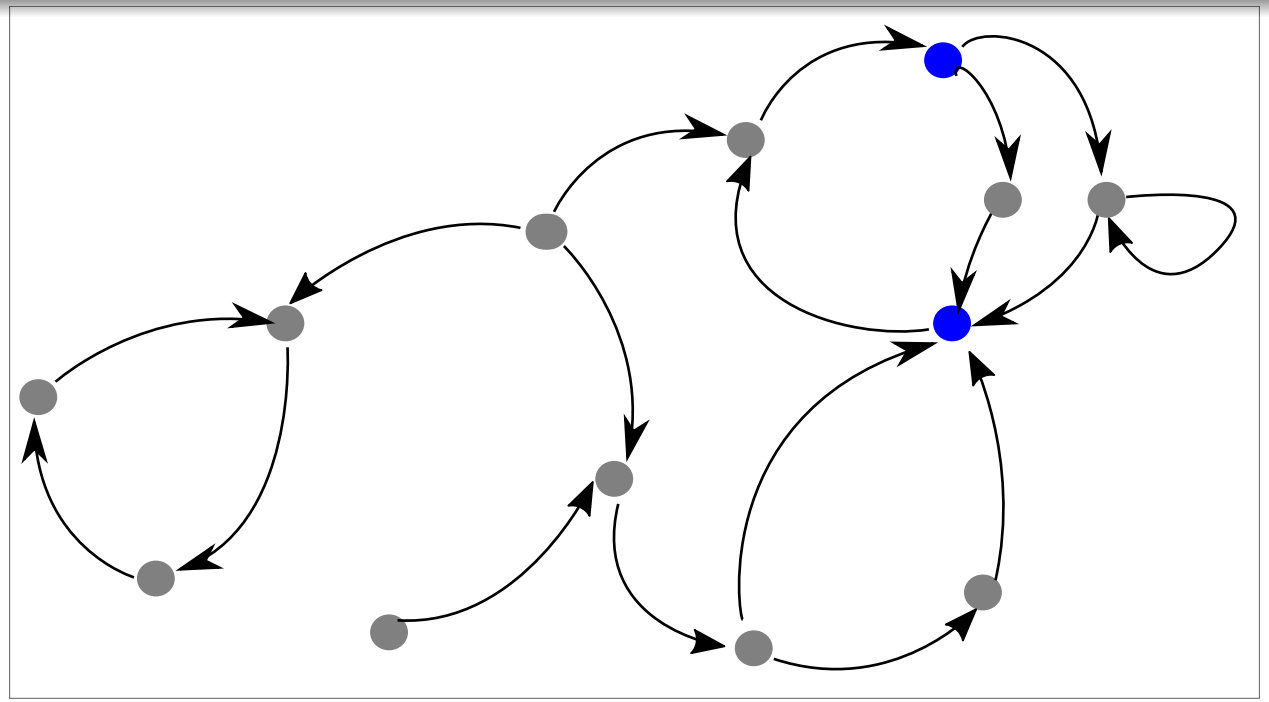
Getting the largest positive invariant set


A

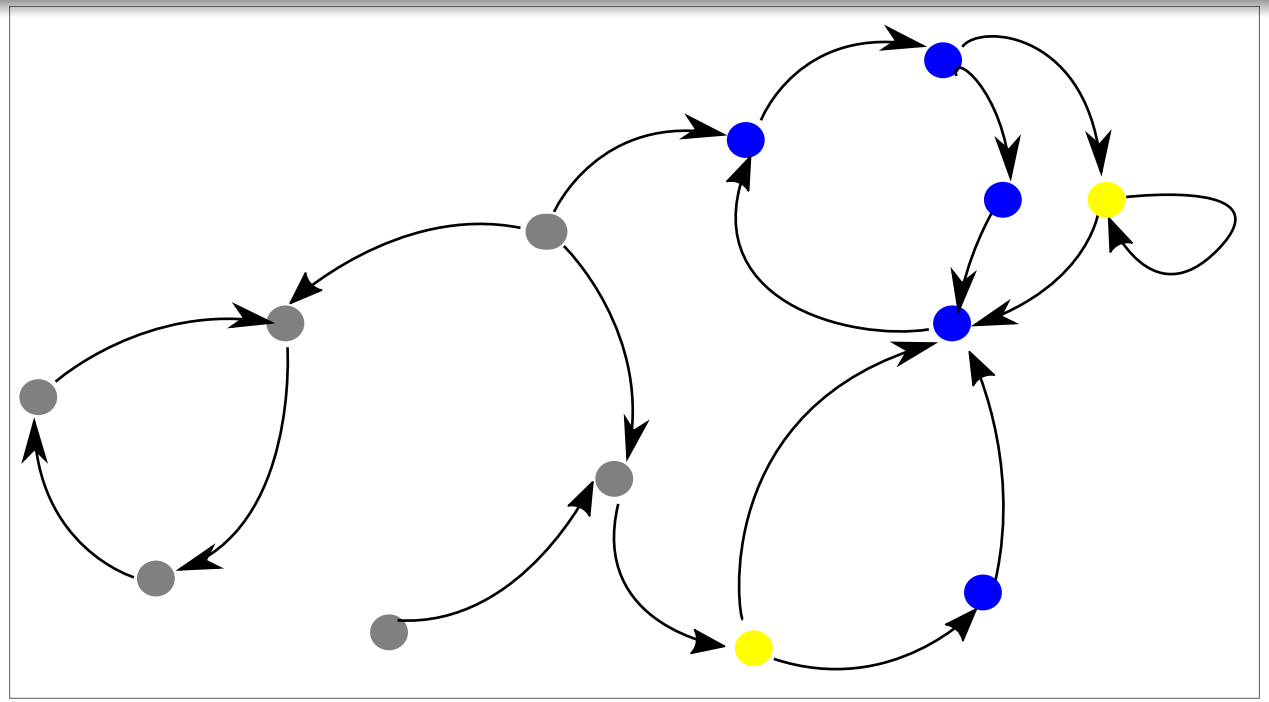

A

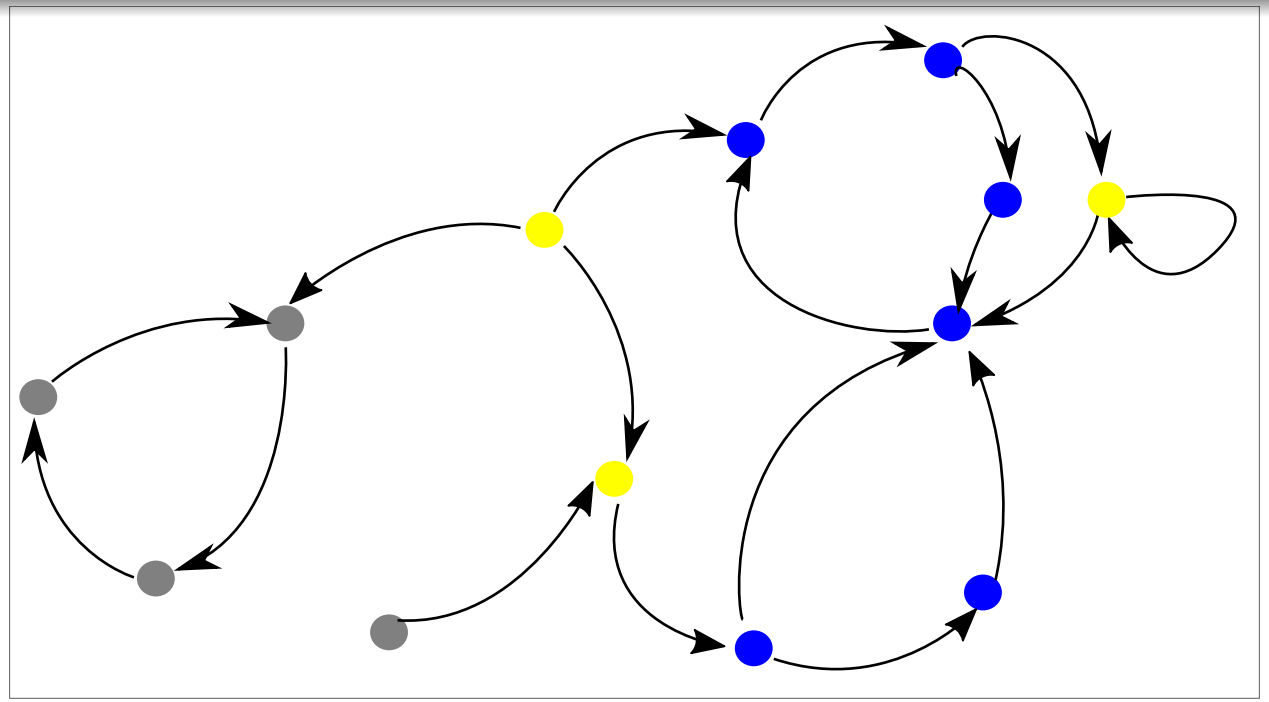

A

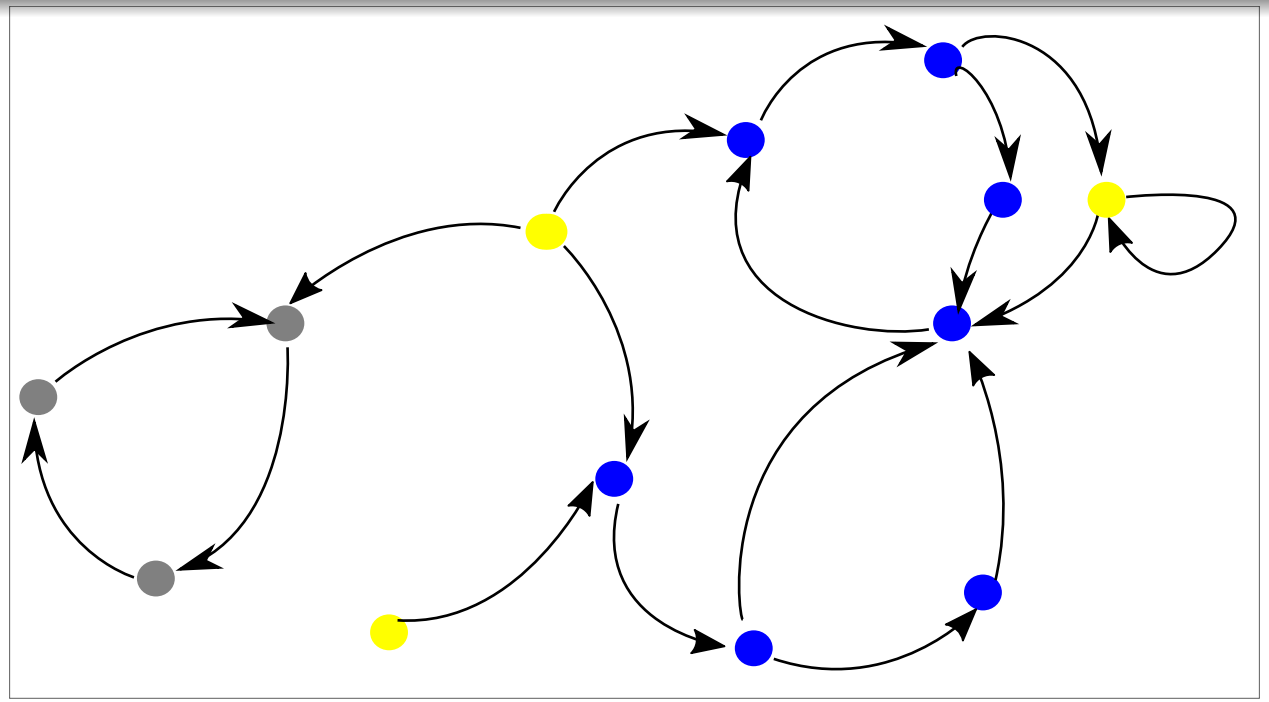
Discrete space

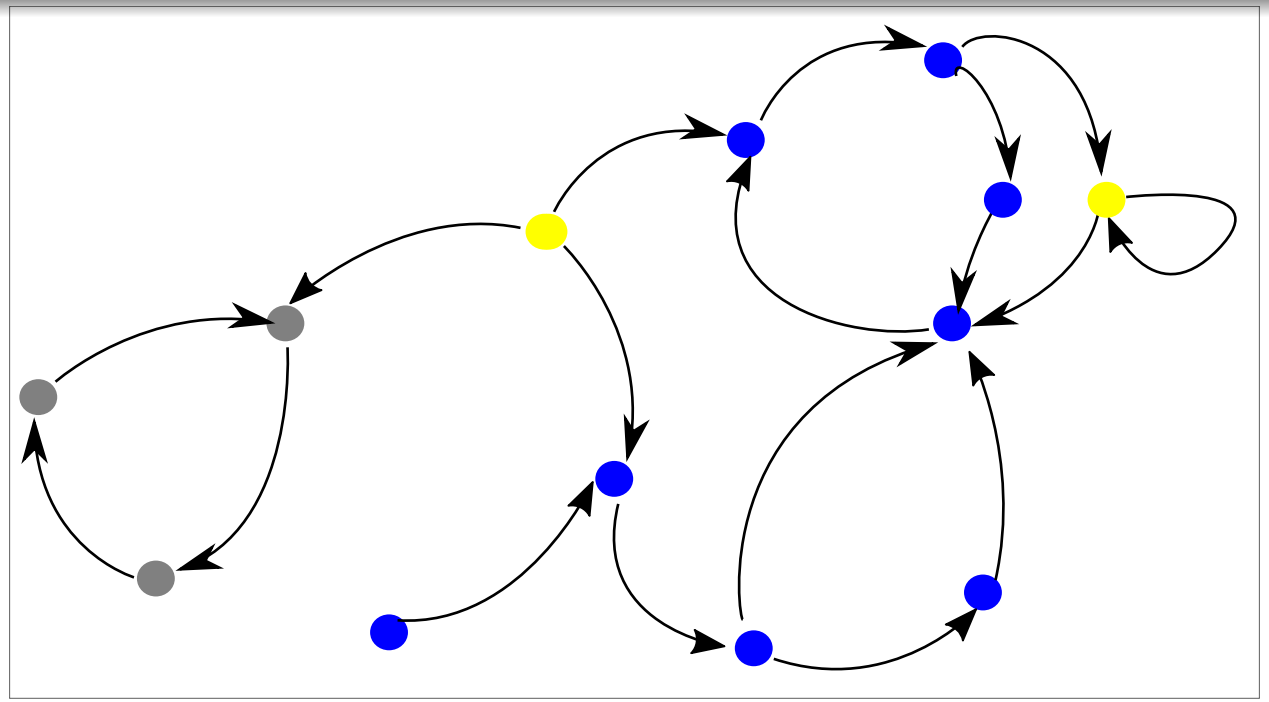


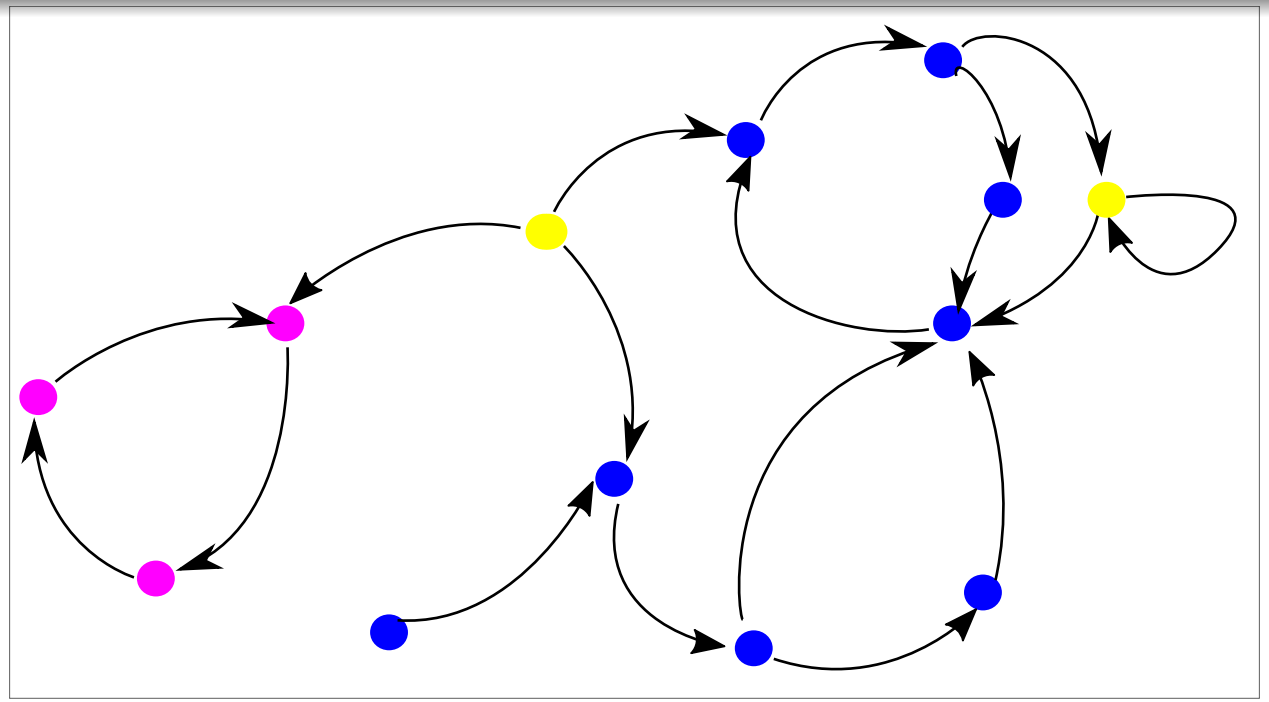
$\mathbb{A}$ is gray











Discrete case

Consider the discrete-time state equation

$$\mathbf{x}_{k+1} = \mathbf{h}(\mathbf{x}_k)$$

We want to compute

$$\mathbb{I}^+(\mathbb{A}) = \left\{\mathbf{x} | \forall k \geq 0, \mathbf{h}^k(\mathbf{x}) \in \mathbb{A}\right\}.$$

Equivalently, $\mathbb{I}^+(\mathbb{A})$ is the largest subset of $\mathbb{A}$ which satisfies $\mathbf{h}(\mathbb{X}) \subset \mathbb{X}$

Backward method

1. Set $\mathbb{X}_0 = \mathbb{A}$.
2. Repeat $\mathbb{X}_{k+1} = \mathbf{h}^{-1}(\mathbb{X}_k) \cap \mathbb{X}_k$, until $\mathbb{X}_{k+1} = \mathbb{X}_k$.

The principle is to remove all $\mathbf{x}$ that leave $\mathbb{X}_k$ until no more point can be removed.
If the algorithm terminates, we have $\mathbb{X}_k = \mathbb{I}^+(\mathbb{A})$.

In practice, we have two sets $\mathbb{A}^-, \mathbb{A}^+$ such that $\mathbb{A}^- \subset \mathbb{A} \subset \mathbb{A}^+$ and an outer apporximation $\left[\mathbf{h}^{-1}\right]$ for $\mathbf{h}^{-1}$ (i.e., $\mathbf{h}^{-1}(\mathbb{X}) \subset \left[\mathbf{h}^{-1}\right](\mathbb{X})$).

Outer approximation

1. Set $\mathbb{X}_0^+ = \mathbb{A}^+$.
2. Repeat $\mathbb{X}_{k+1}^+ = \left[\mathbf{h}^{-1}\right](\mathbb{X}_k^+) \cap \mathbb{X}_k^+$ several times.

This means that we remove the $\mathbf{x}$ that certainly leave $\mathbb{X}_k^+$.
At each step $\mathbb{I}^+(\mathbb{A}) \subset \mathbb{X}_k^+$.

Inner approximation

1. Set $\mathbb{X}_0^- = \mathbb{A}^-$.
2. Repeat $\mathbb{X}_{k+1}^- = \overline{[\mathbf{h}^{-1}](\overline{\mathbb{X}_k^-})} \cap \mathbb{X}_k^-$ several times.

The principle is to remove the $\mathbf{x}$ that may possibly leave $\mathbb{X}_k^-$ until no more point can be removed.
If the algorithm terminates, $\mathbb{X}_k^- \subset \mathbb{I}^+(\mathbb{A})$.

Constraint network

A Constraint Network is composed of

1) a set of variables $\mathscr{V} = \{x_1 \in \mathbb{X}_1, \ldots, x_n \in \mathbb{X}_n\}$,
2) a set of constraints $\mathscr{C} = \{c_1, \ldots, c_m\}$ and
3) a set of domains $\{[x_1], \ldots, [x_n]\}$.

Example

1) a set of variables $\mathscr{V} = \{x \in \mathbb{R}, y \in \mathbb{R}\}$,
2) a set of constraints $\mathscr{C} = \{y = x^2, y = \sqrt{x}\}$ and
3) a set of domains $\{[-1,2],[-1,2]\}$.

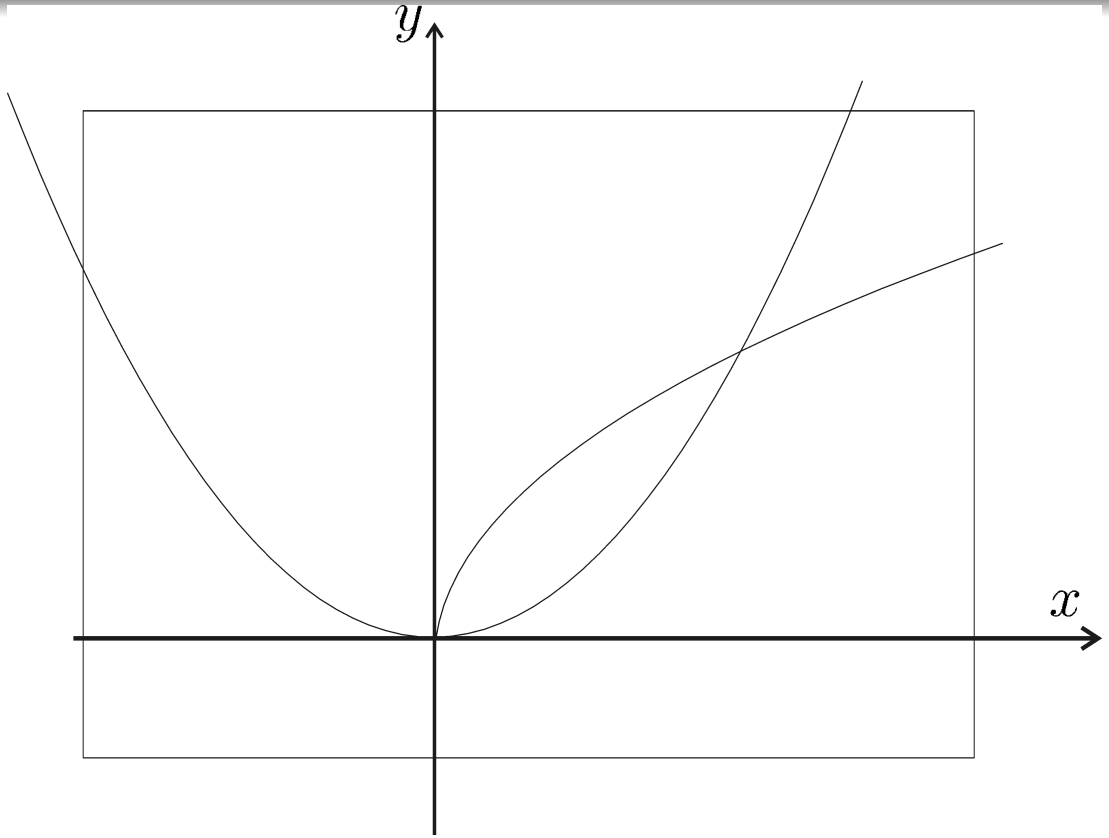
We have a system of two equations.

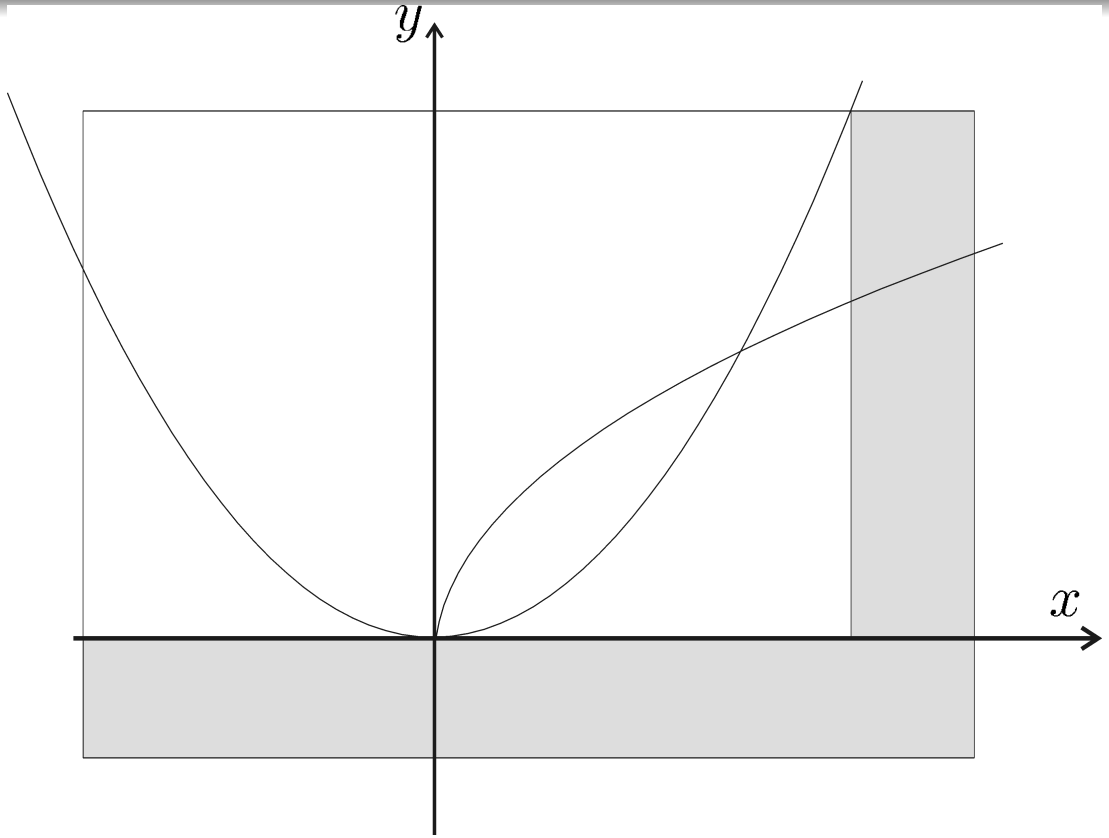
$$\begin{aligned} y &= x^2 \\ y &= \sqrt{x}. \end{aligned}$$

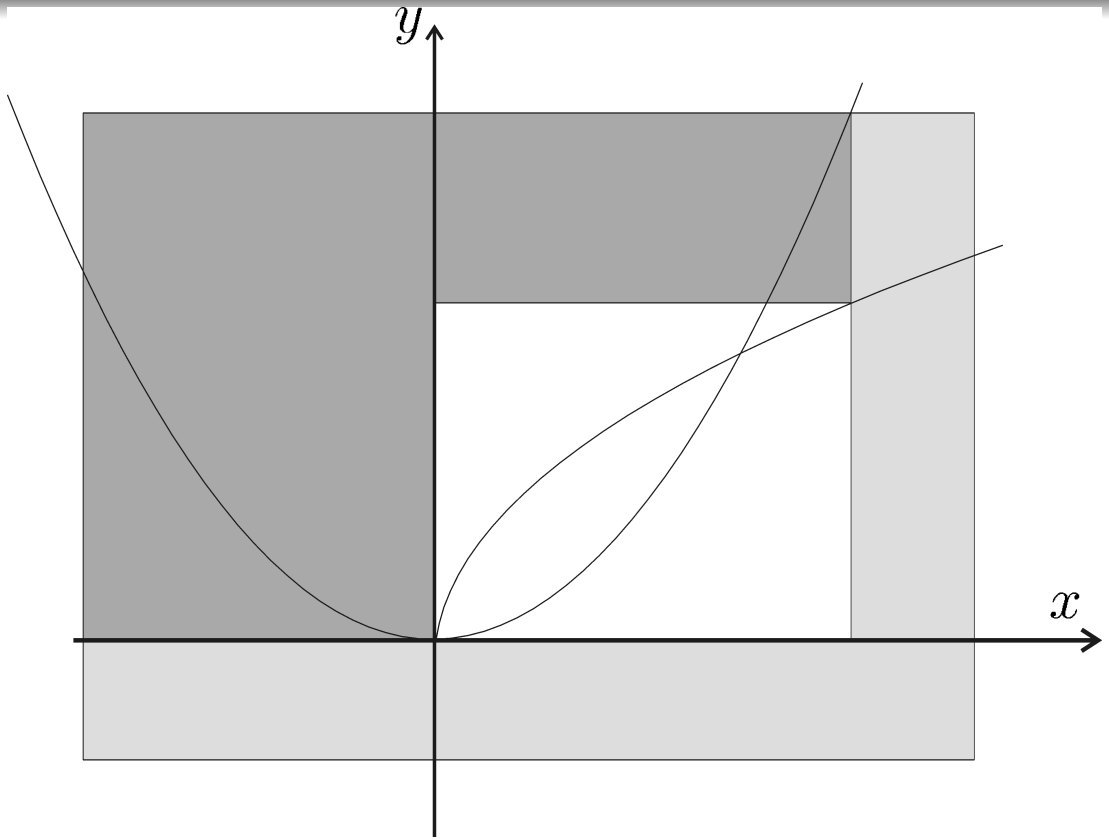
We can build two contractors

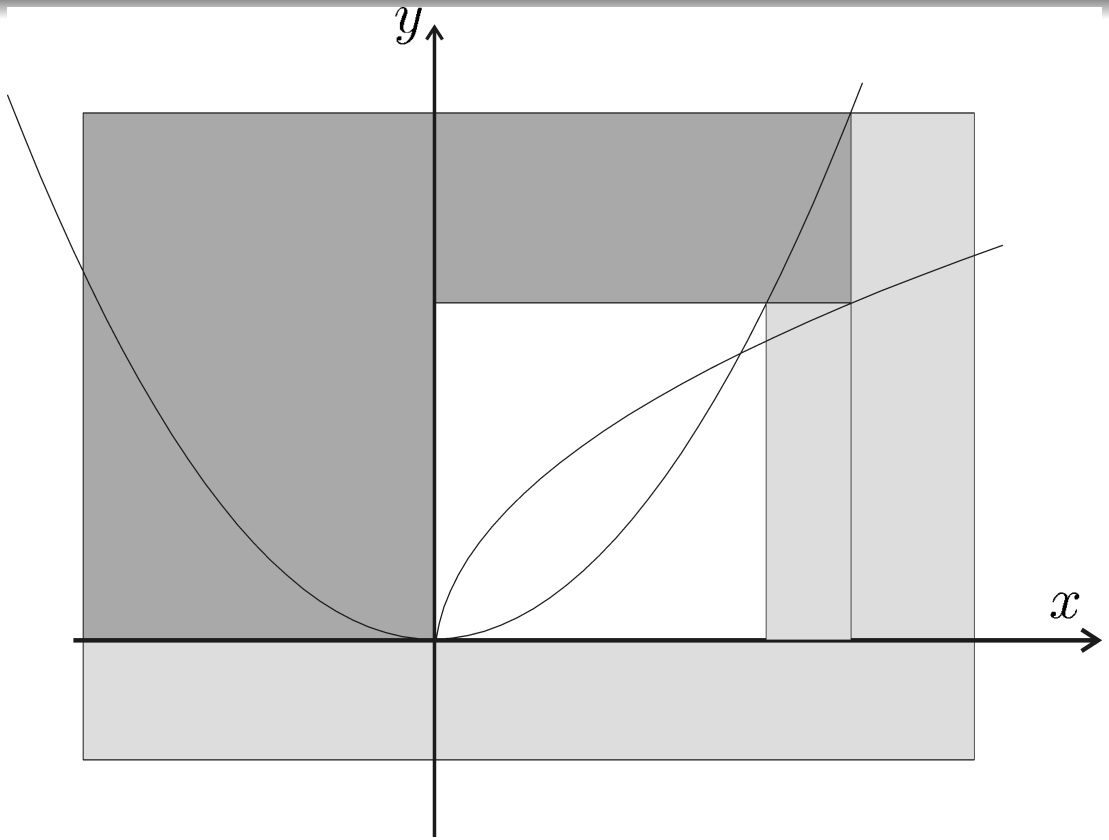
$$\mathscr{C}_1 : \begin{cases} [y] = [y] \cap [x]^2 \\ [x] = [x] \cap \sqrt{[y]} \end{cases} \quad \text{associated to } y = x^2$$

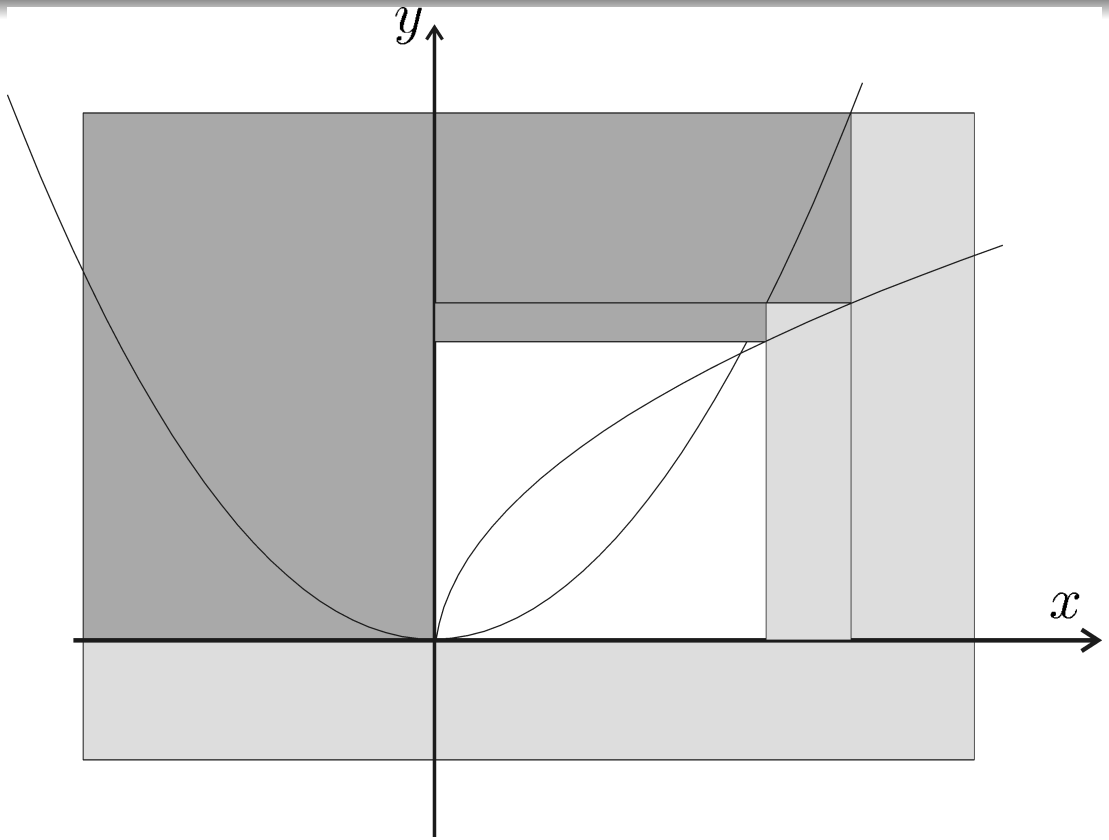
$$\mathscr{C}_2 : \begin{cases} [y] = [y] \cap \sqrt{[x]} \\ [x] = [x] \cap [y]^2 \end{cases} \quad \text{associated to } y = \sqrt{x}$$

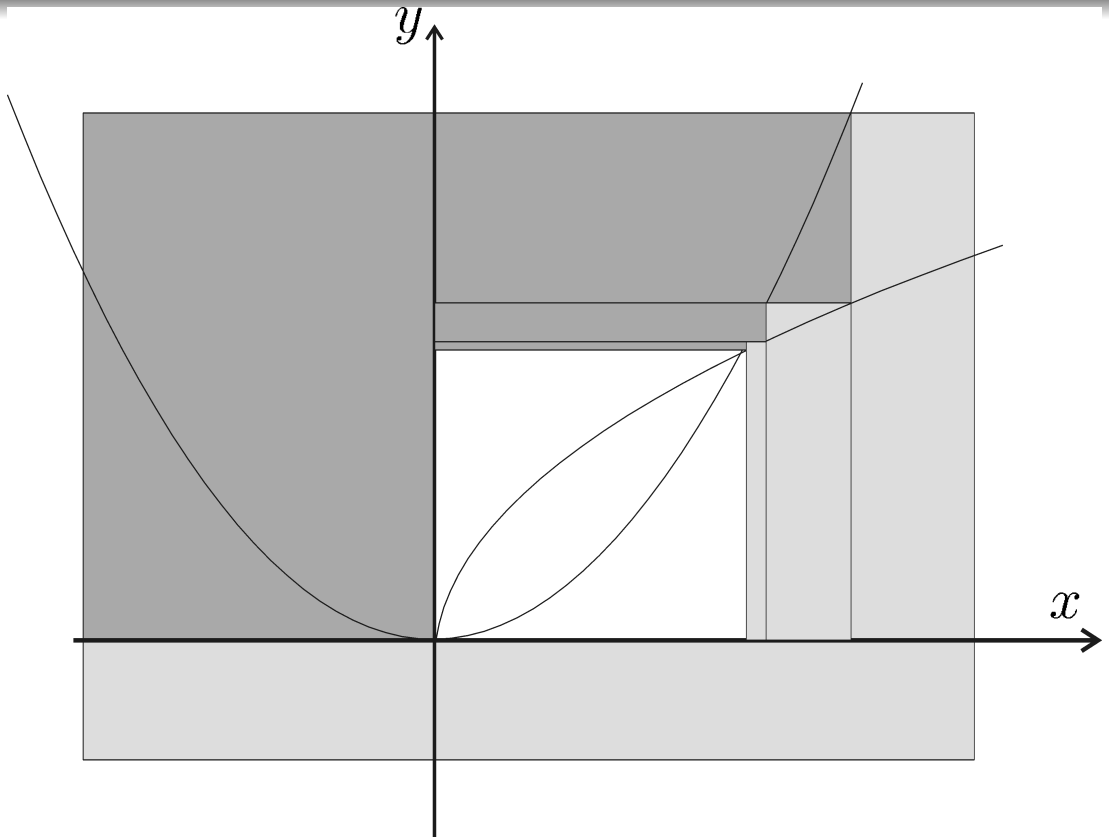

y
x

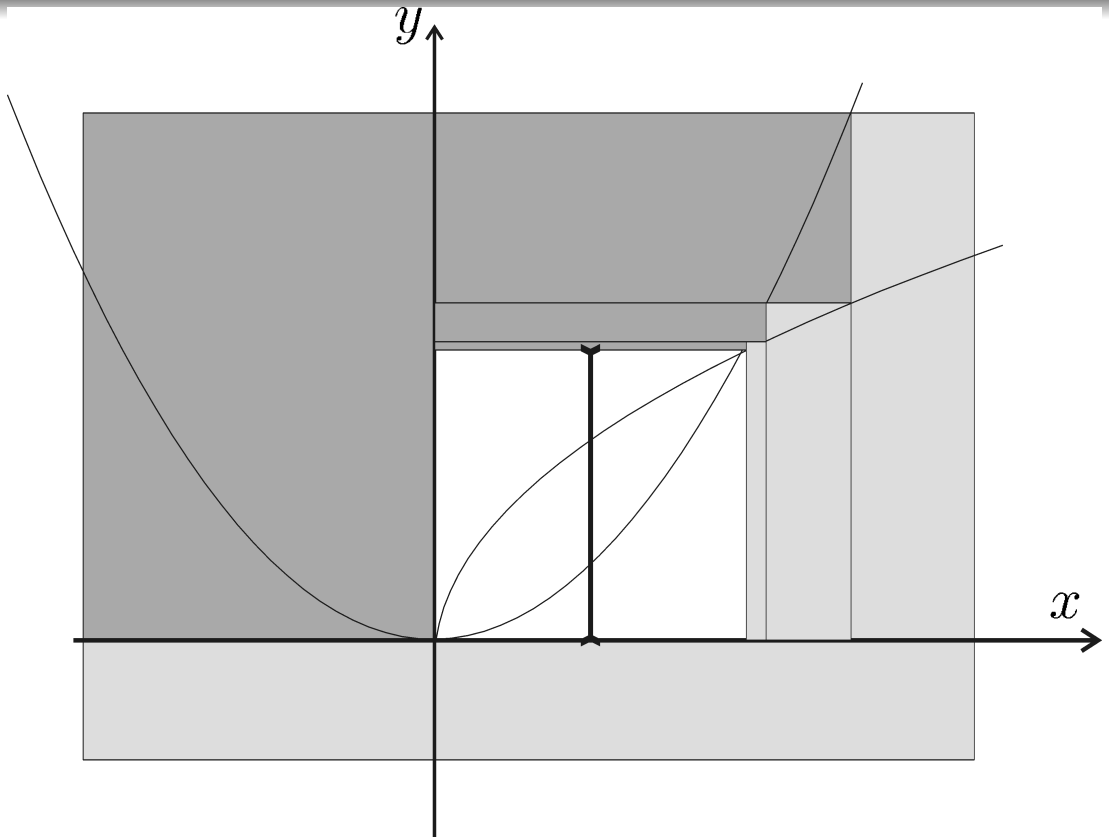

y
x

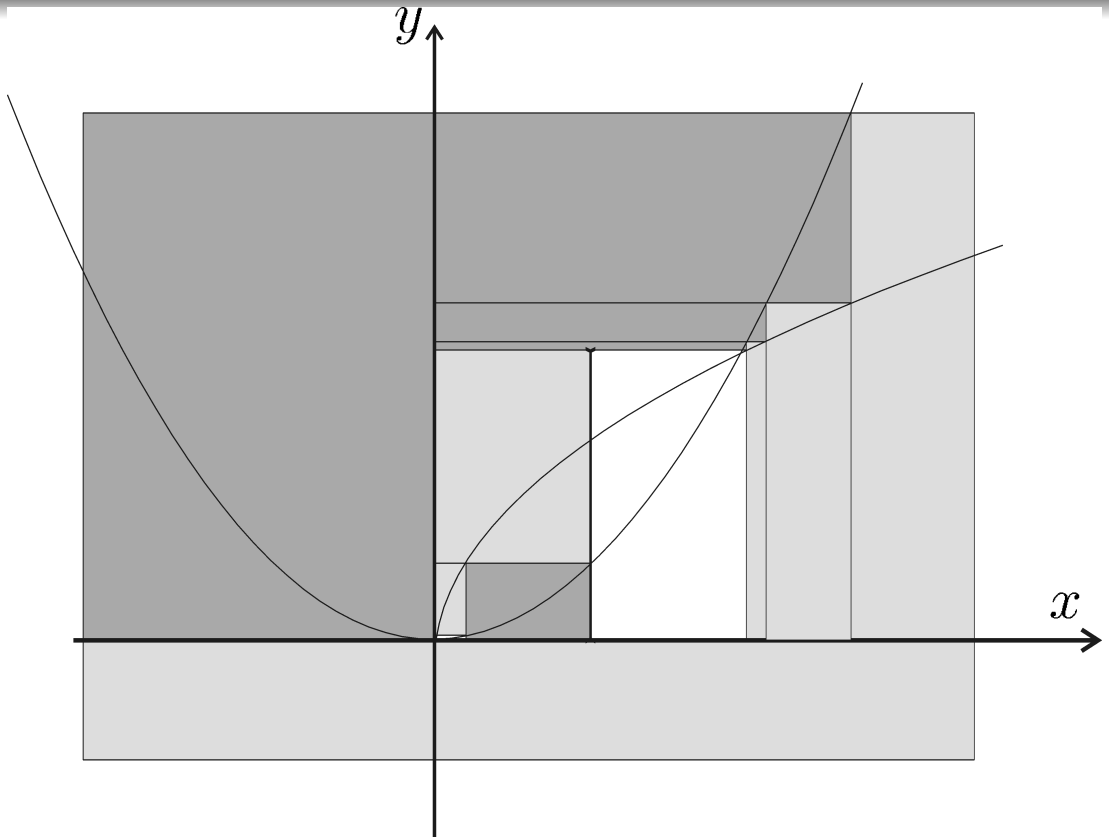

y
x

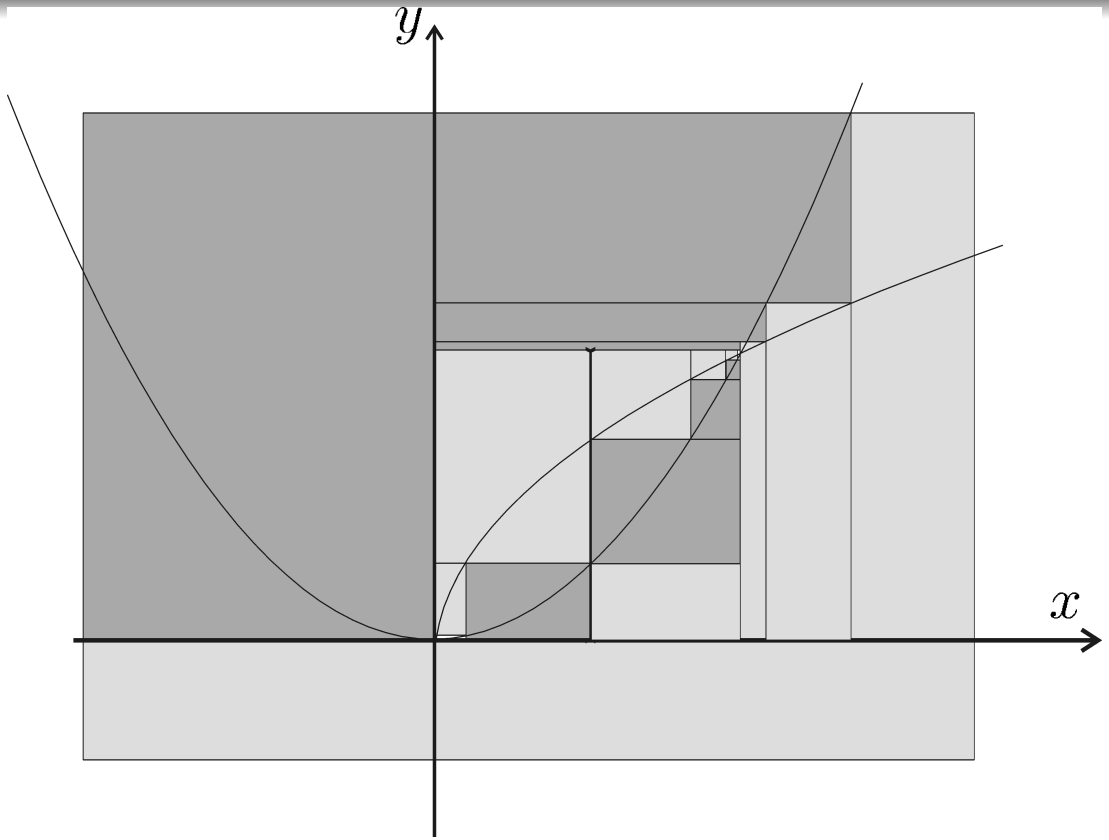

y
x


y
x


y
x


y
x


y
x


y
x

Constraint Network

A Constraint Network is composed of

1) a set of variables $\mathscr{V} = \{x_1 \in \mathbb{X}_1, \ldots, x_n \in \mathbb{X}_n\}$,
2) a set of constraints $\mathscr{C} = \{c_1, \ldots, c_m\}$ and
3) a set of domains $\{[x_1], \ldots, [x_n]\}$.

Classically, the $\mathbb{X}_i$ are lattices, but it is not necessary.

The domains $[x_i]$ should be representable in the machine.
The domains should be a Moore family.

An example with angles

The set of angles $\mathbb{A}$ is not a lattice. Thus, we cannot define intervals of angles.

The family $\mathbb{IA}$ is a Moore family (containing $\mathbb{A}$) if

$$\forall i, [a](i) \in \mathbb{IA} \Rightarrow \bigcap_i [a](i) \in \mathbb{IA}$$

E. H. Moore

Eliakim Hastings Moore

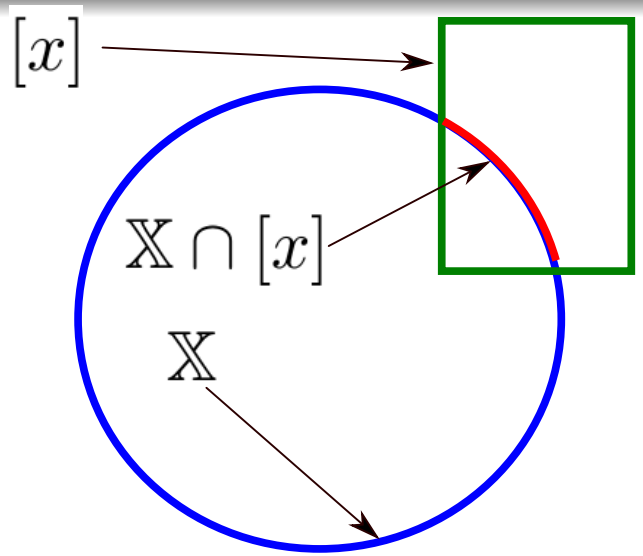
Born	January 26, 1862 Marietta, Ohio, U.S.
Died	December 30, 1932 (aged 70) Chicago, Illinois, U.S.
Nationality	American
Fields	Mathematics
Institutions	University of Chicago 1892–31 Yale University 1887–89

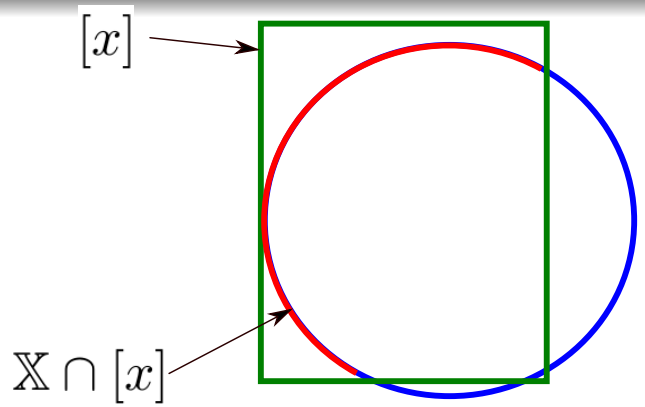
Embedding

Embedding. To have a Moore family, we perform an embedding:

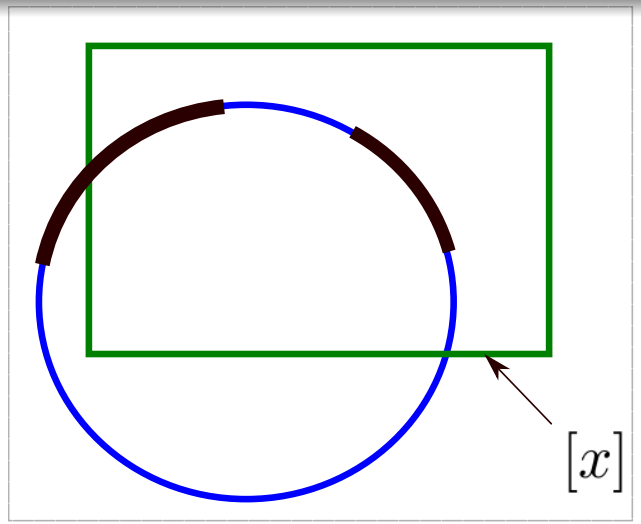
$$\alpha \mapsto \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \in \mathbb{R}^2$$

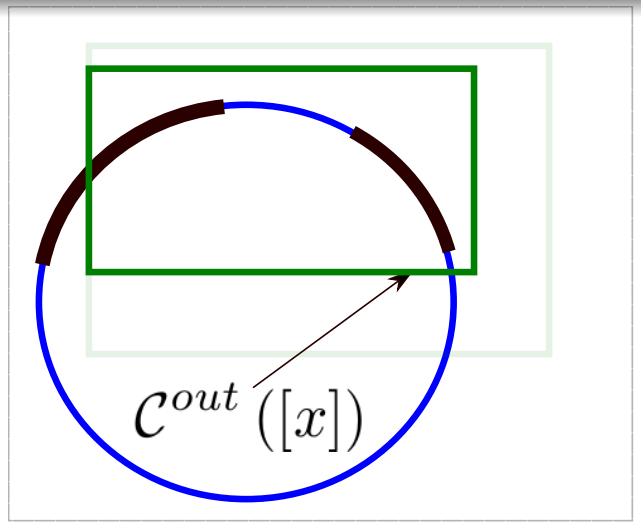
Now, we introduce a pessimism (*Embedding effect*).

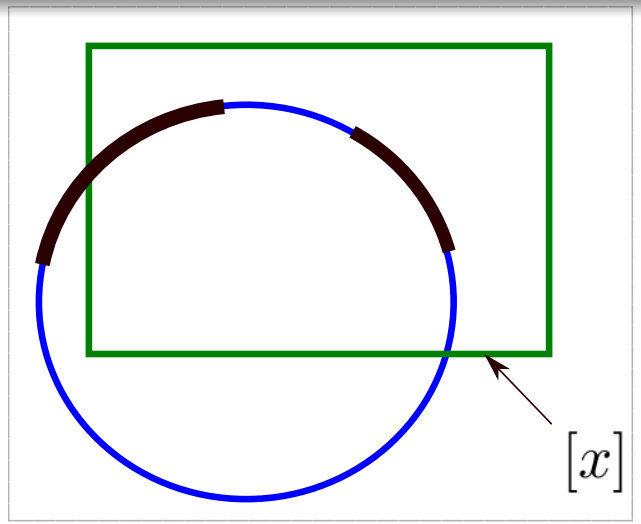

$[x]$
$\mathbb{X} \cap [x]$
$\mathbb{X}$

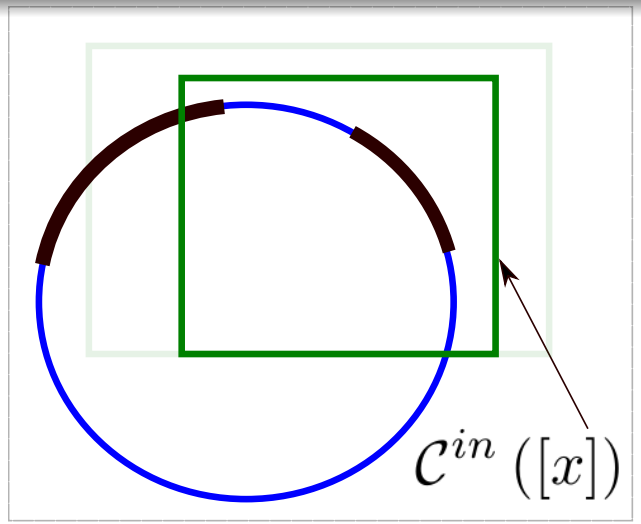

$[x]$
$\mathbb{X} \cap [x]$

Consider a constraint, such as $2\alpha \in [5,6]$.


$[x]$


$\mathcal{C}^{out}([x])$

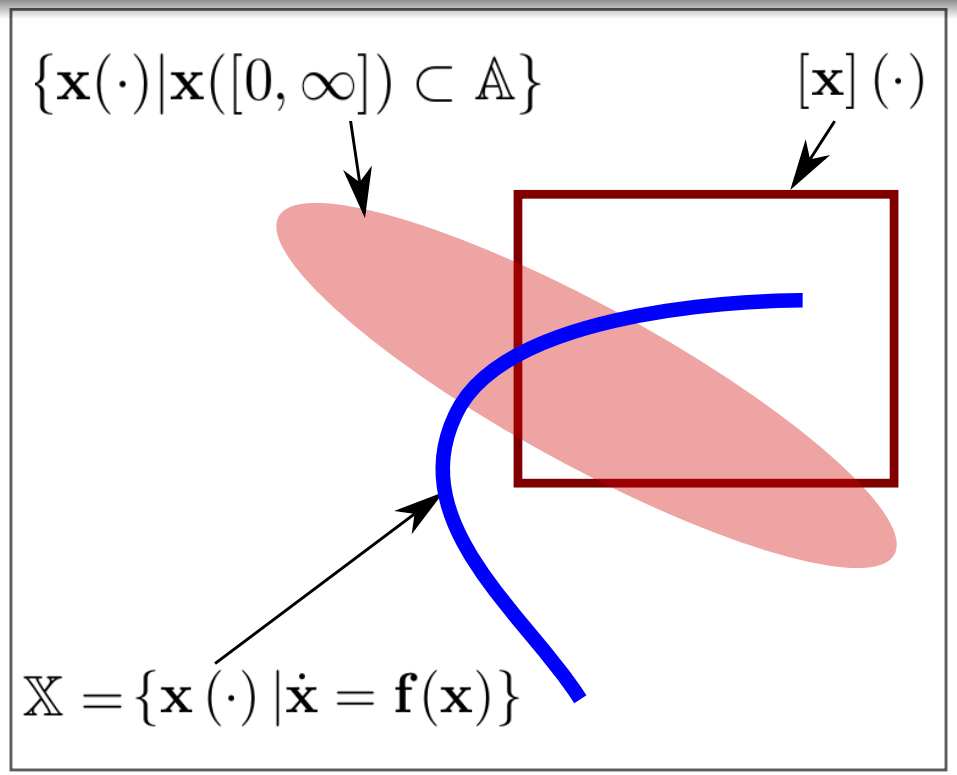

$[x]$

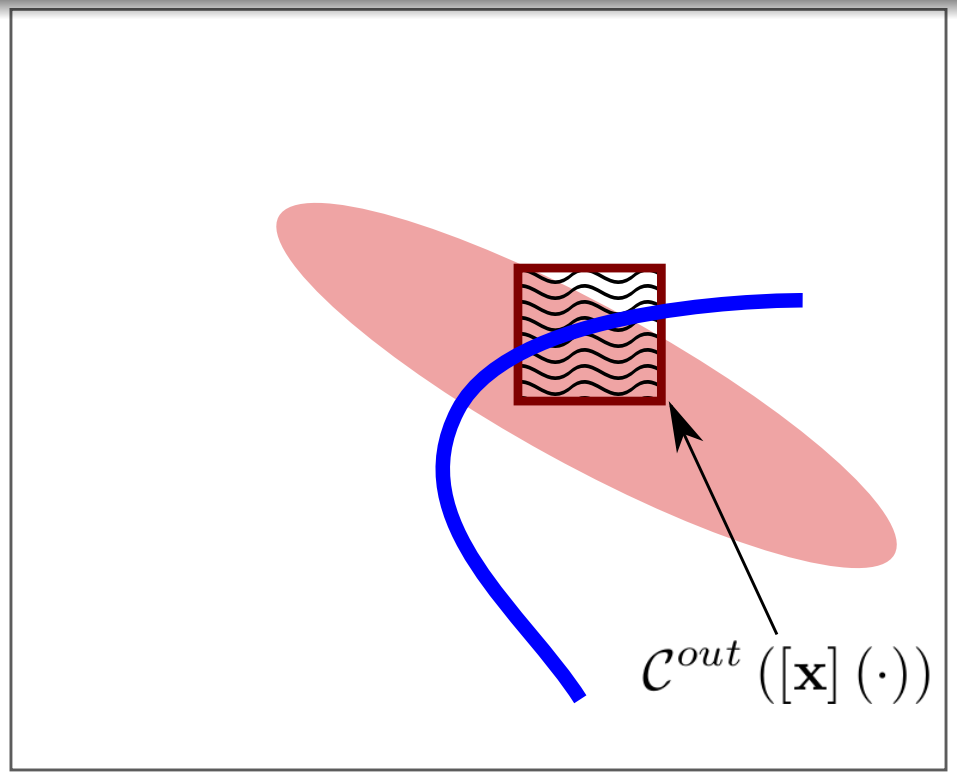

$\mathcal{C}^{in}([x])$

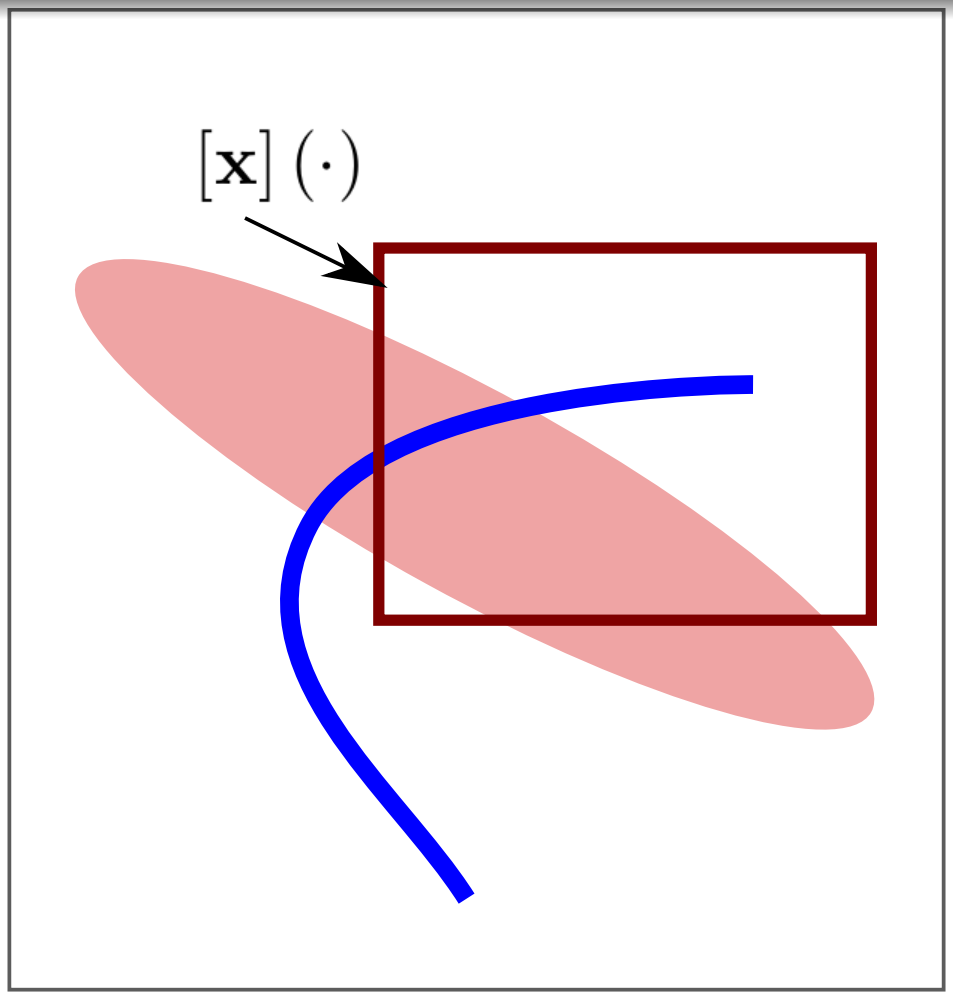
Positive invariance constraint

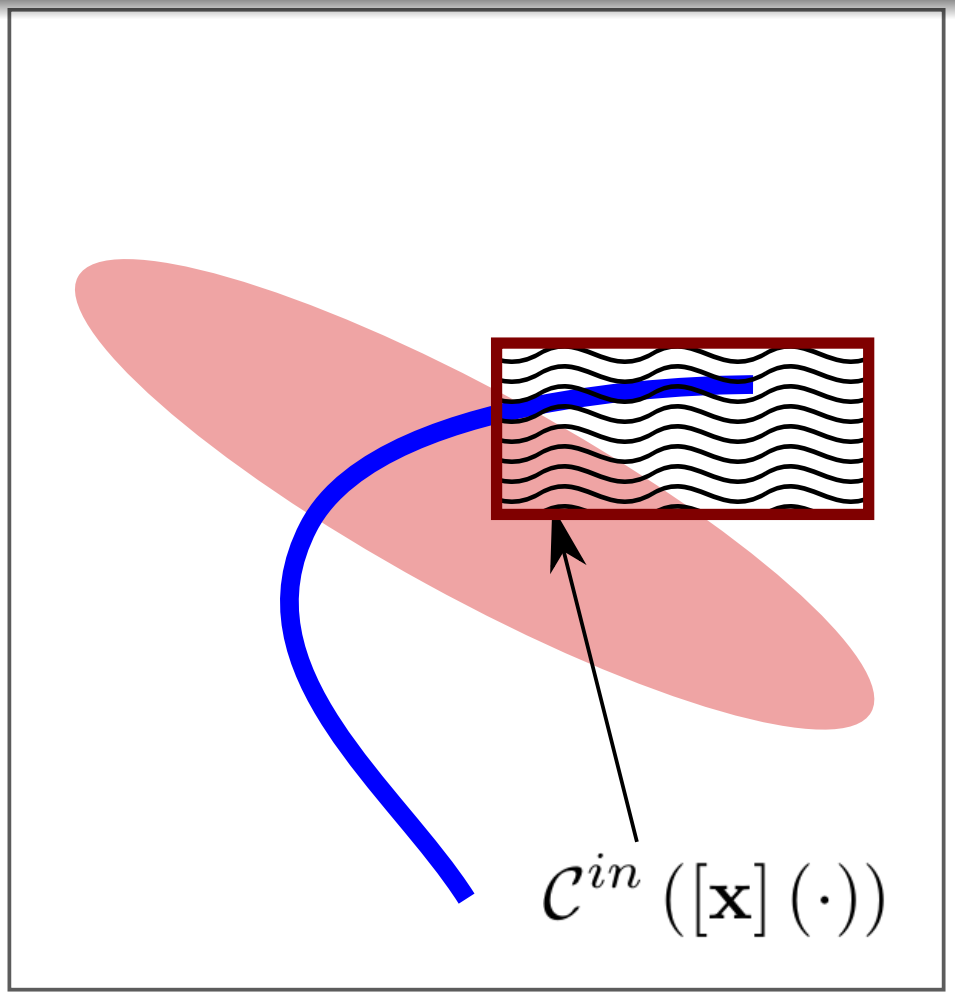
We consider a constraint network composed of

1) One trajectory $\mathscr{V} = \{\mathbf{x}(\cdot) \in \mathbb{X} = \{\mathbf{x}(\cdot) | \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})\}\}$,
2) One constraint $\mathscr{C} = \{\mathbf{x}([0, \infty]) \subset \mathbb{A}\}$
3) One maze $\{[\mathbf{x}]\}$.

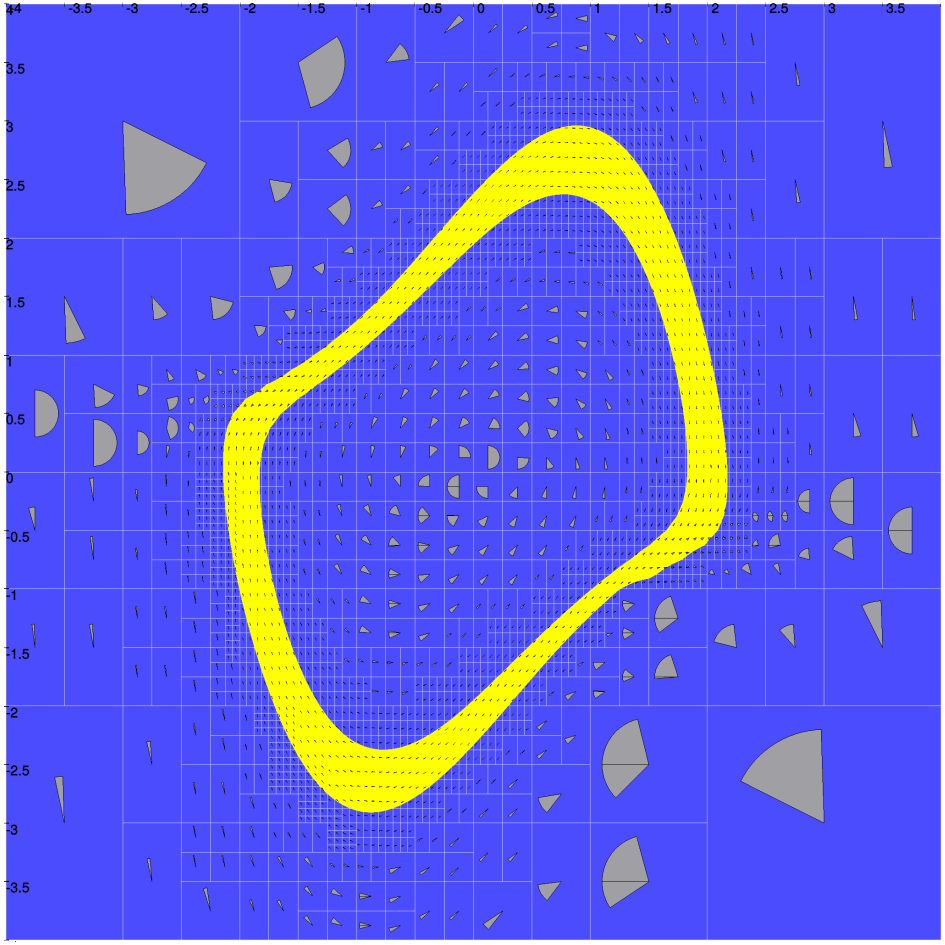

$\{\mathbf{x}(\cdot)|\mathbf{x}([0,\infty]) \subset \mathbb{A}\}$
$[\mathbf{x}](\cdot)$
$\mathbb{X} = \{\mathbf{x}(\cdot)\,|\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})\}$


$\mathcal{C}^{out}\left([\mathbf{x}]\left(\cdot\right)\right)$


[x] (·)


$\mathcal{C}^{in}\left([\mathbf{x}]\left(\cdot\right)\right)$

www.ensta-bretagne.fr/lemezo/pyinvariant/pyinvariant.html



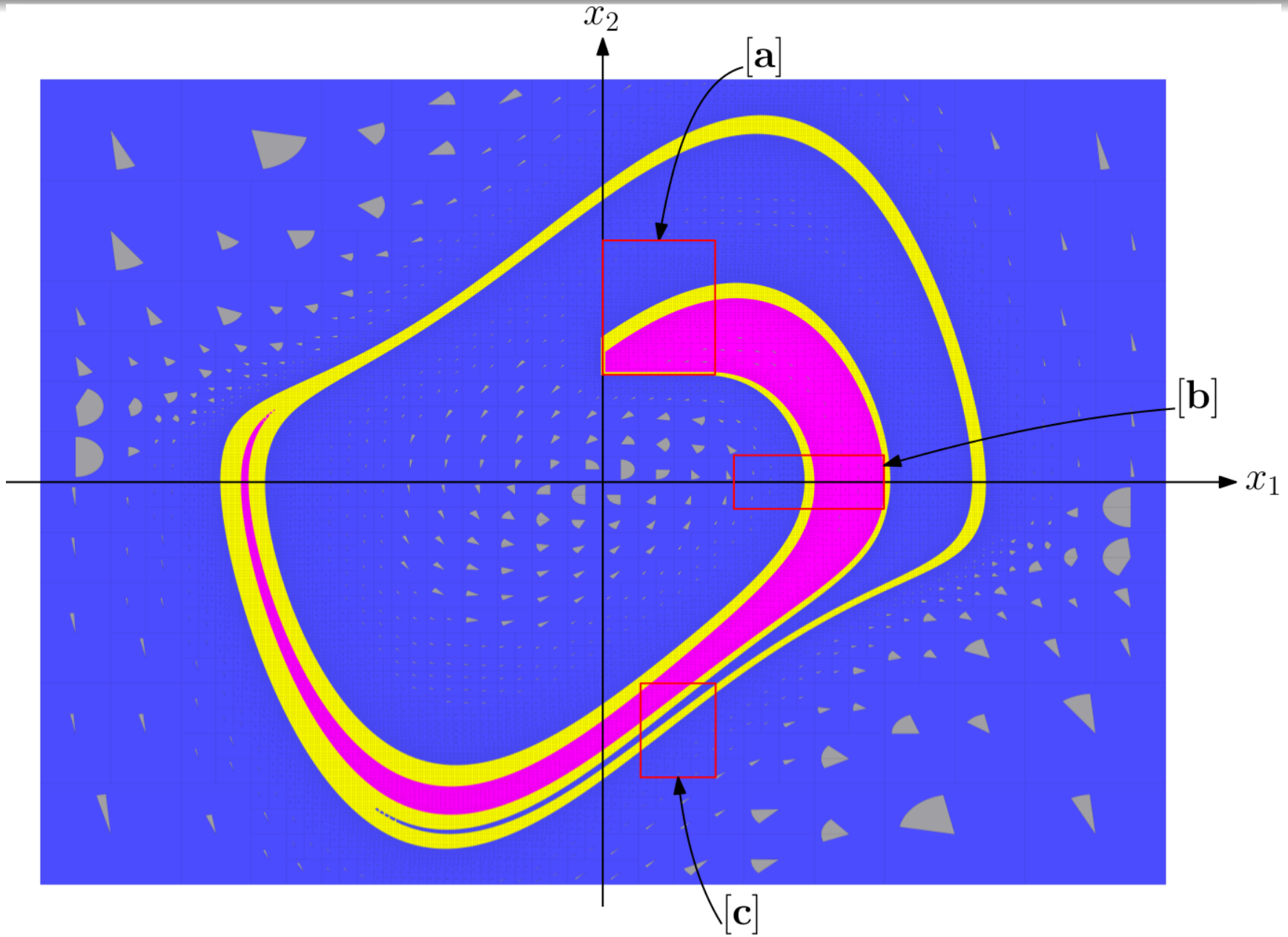
$$\mathbb{X} = [-4,4]^{\times 2} \backslash [-0.1,0.1]^{\times 2}, \mathbb{I}^{+}_{\varphi^{-1}}(\mathbb{X}) \cap \mathbb{I}^{+}_{\varphi}(\mathbb{X})$$

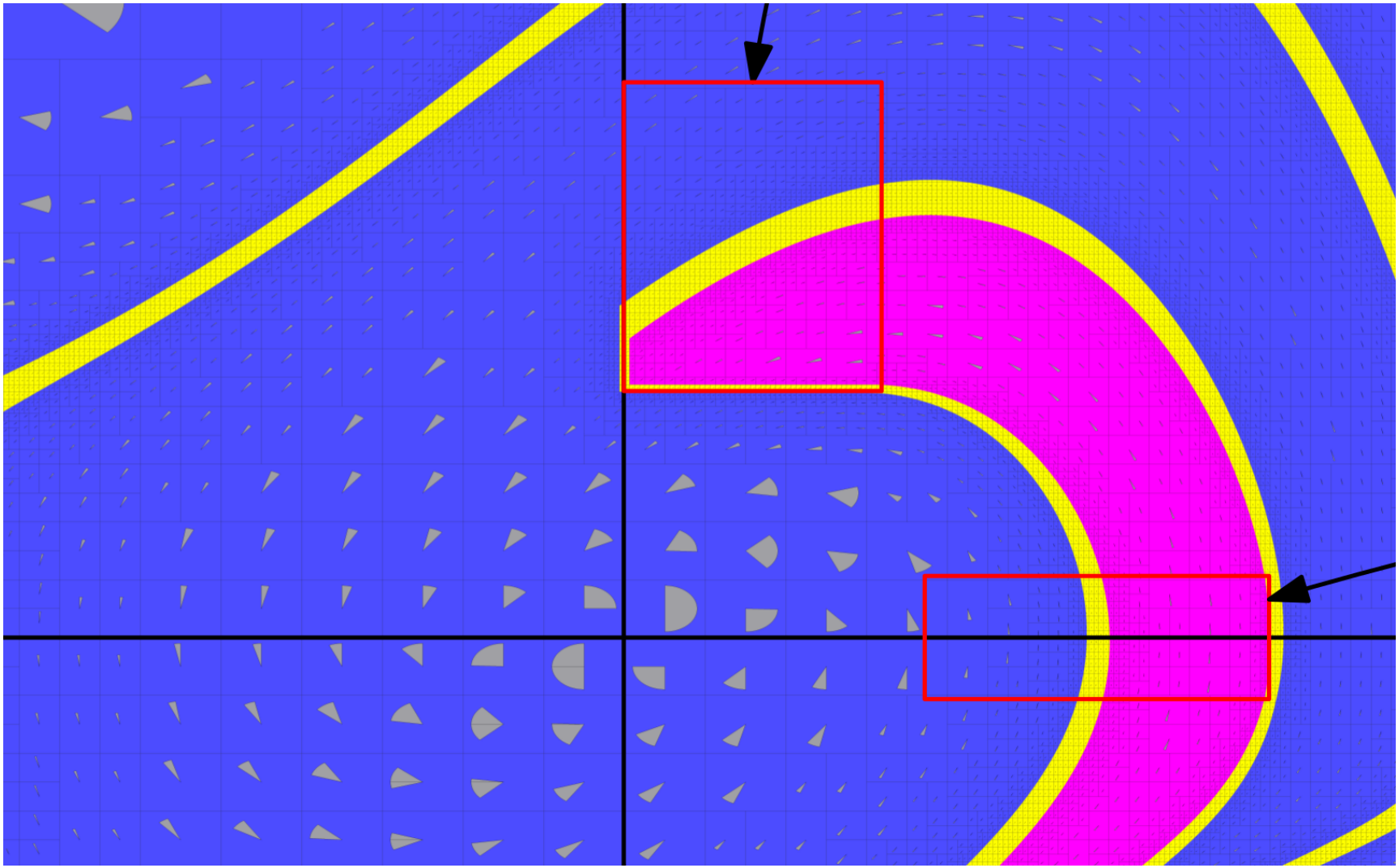
Eulerian smoother

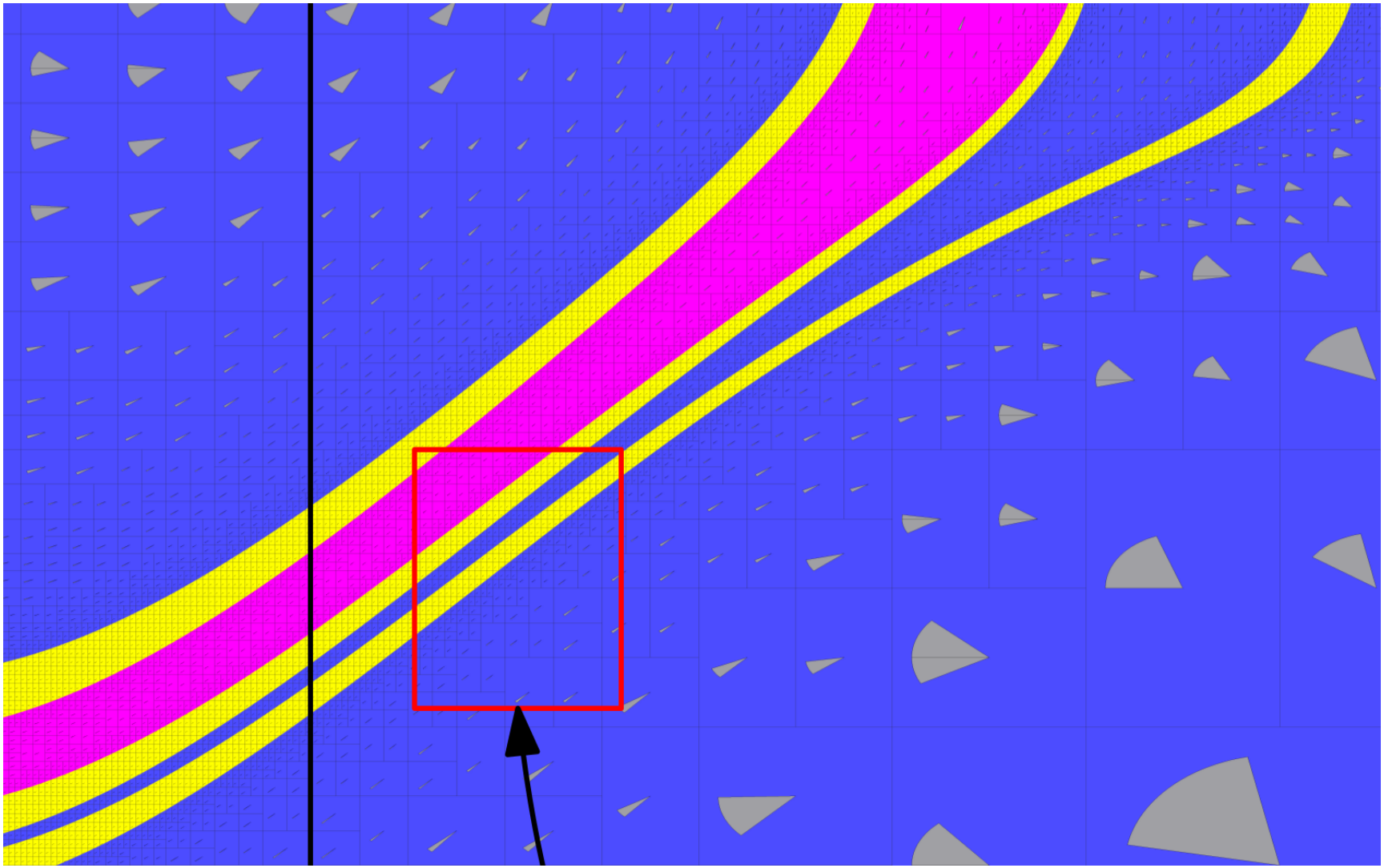
Take the Van der Pol system with

$$\begin{aligned}\mathbb{X}_0 &= [\mathbf{a}] = [0, 0.6] \times [0.8, 1.8] \\ \mathbb{X}_1 &= [\mathbf{b}] = [0.7, 1.5] \times [-0.2, 0.2] \\ \mathbb{X}_2 &= [\mathbf{c}] = [0.2, 0.6] \times [-2.2, -1.5]\end{aligned}$$

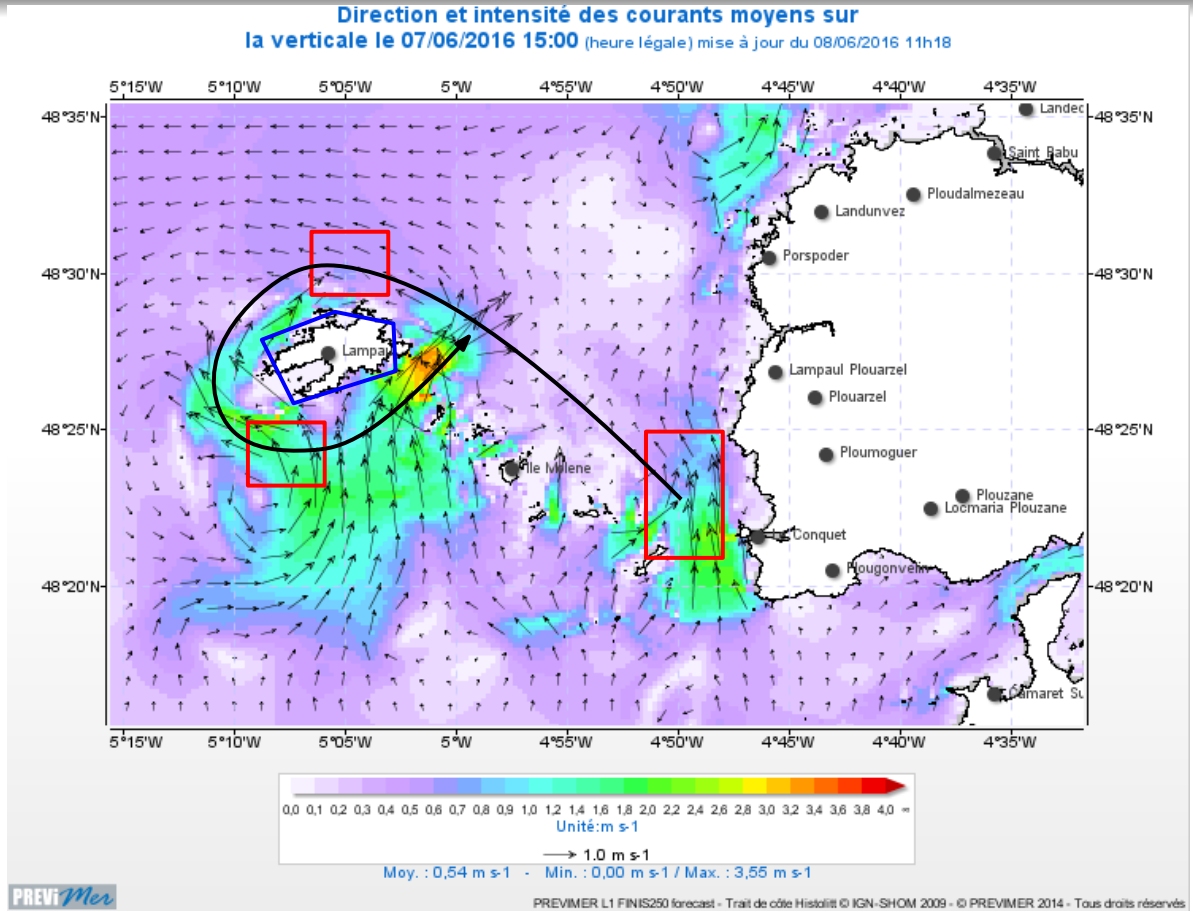
1) One trajectory $\mathscr{V} = \{\mathbf{x}(\cdot) \in \mathbb{X} = \{\mathbf{x}(\cdot) | \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})\}\}$,
2) Three constraints $\mathscr{C} = \{\mathbf{x}(0) \in [\mathbf{a}], \mathbf{x}(\cdot) \cap [\mathbf{b}] \neq \emptyset, , \mathbf{x}(\cdot) \cap [\mathbf{c}] \neq \emptyset\}$
3) One maze $\{[\mathbf{x}]\}$.


x_2
[a]
[b]
x_1
[c]





An application of Eulerian state estimation moving taking advantage of ocean currents.



Visiting the three red boxes using a buoy that follows the currents is an Eulerian state estimation problem


T. Le Mézo L. Jaulin and B. Zerr.
Bracketing the solutions of an ordinary differential equation with uncertain initial conditions.
Applied Mathematics and Computation, 2017.


T. Le Mézo, L. Jaulin, and B. Zerr.
An interval approach to compute invariant sets.
IEEE Transaction on Automatic Control, 2017.