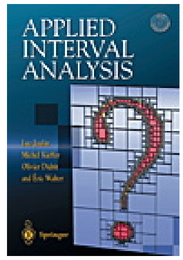
Interval constraint propagation; applications to control, estimation and robotics



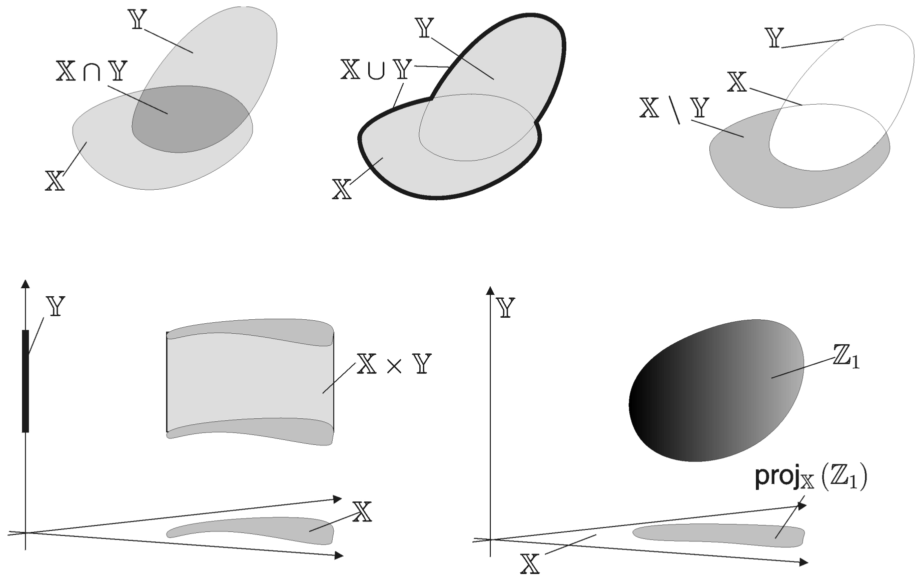
Luc Jaulin,
ENSIETA, Brest, France

1 Set computation

1.1 Basic notions on set theory

We define

$$
\begin{aligned}
\mathbb{X} \cap \mathbb{Y} &\overset{\text{def}}{=} \{x \mid x \in \mathbb{X} \text{ and } x \in \mathbb{Y}\} \\
\mathbb{X} \cup \mathbb{Y} &\overset{\text{def}}{=} \{x \mid x \in \mathbb{X} \text{ or } x \in \mathbb{Y}\} \\
\mathbb{X} \setminus \mathbb{Y} &\overset{\text{def}}{=} \{x \mid x \in \mathbb{X} \text{ and } x \notin \mathbb{Y}\} \\
\mathbb{X} \times \mathbb{Y} &\overset{\text{def}}{=} \{(x, y) \mid x \in \mathbb{X} \text{ and } y \in \mathbb{Y}\} \\
\mathrm{proj}_{\mathbb{X}}(\mathbb{Z}) &\overset{\text{def}}{=} \{x \in \mathbb{X} \mid \exists y \in \mathbb{Y},\ (x, y) \in \mathbb{Z}\}.
\end{aligned}
$$



𝕐
𝕏 ∩ 𝕐
𝕏
𝕐
𝕏 ∪ 𝕐
𝕏
𝕐
𝕏
𝕏 \ 𝕐
𝕐
𝕏 × 𝕐
𝕏
𝕐
ℤ1
projℤ(ℤ1)
𝕏

Exercise: If $\mathbb{X} = \{a, b, c, d\}$ and $\mathbb{Y} = \{b, c, x, y\}$, then

$$\begin{aligned}
\mathbb{X} \cap \mathbb{Y} &= ? \\
\mathbb{X} \cup \mathbb{Y} &= ? \\
\mathbb{X} \setminus \mathbb{Y} &= ? \\
\mathbb{X} \times \mathbb{Y} &= ?
\end{aligned}$$

Exercise: If $\mathbb{X} = \{a, b, c, d\}$ and $\mathbb{Y} = \{b, c, x, y\}$, then

$$\begin{aligned}
\mathbb{X} \cap \mathbb{Y} &= \{b, c\} \\
\mathbb{X} \cup \mathbb{Y} &= \{a, b, c, d, x, y\} \\
\mathbb{X} \setminus \mathbb{Y} &= \{a, d\} \\
\mathbb{X} \times \mathbb{Y} &= \{(a, b), (a, c), (a, x), (a, y), \\
& \quad \ldots, (d, b), (d, c), (d, x), (d, y)\}
\end{aligned}$$

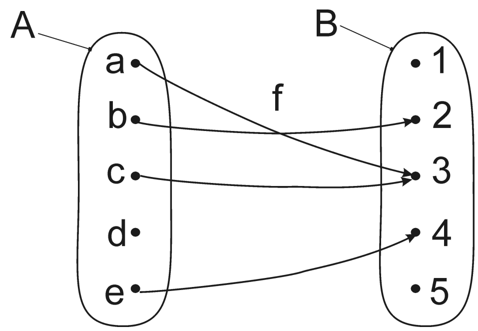
The *direct image* of $\mathbb{X}$ by f is

$$f(\mathbb{X}) \triangleq \{f(x) \mid x \in \mathbb{X}\}.$$

The *reciprocal image* of $\mathbb{Y}$ by f is

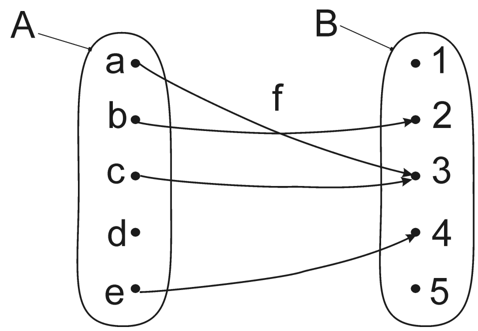
$$f^{-1}(\mathbb{Y}) \triangleq \{x \in \mathbb{X} \mid f(x) \in \mathbb{Y}\}.$$

Exercise: If f is defined as follows



$$
\begin{aligned}
f(A) &= ?. \\
f^{-1}(B) &= ?. \\
f^{-1}(f(A)) &= ? \\
f^{-1}(f(\{b,c\})) &= ?.
\end{aligned}
$$

Exercise: If f is defined as follows



$$\begin{aligned}
f(A) &= \{2,3,4\} = \mathsf{Im}(f). \\
f^{-1}(B) &= \{a,b,c,e\} = \mathsf{dom}(f). \\
f^{-1}(f(A)) &= \{a,b,c,e\} \subset A \\
f^{-1}(f(\{b,c\})) &= \{a,b,c\}.
\end{aligned}$$

Exercise: If $f(x) = x^2$, then

$$\begin{aligned} f([2,3]) &= \ ? \\ f^{-1}([4,9]) &= \ ?. \end{aligned}$$

Exercise: If $f(x) = x^2$, then

$$\begin{aligned} f([2,3]) &= [4,9] \\ f^{-1}([4,9]) &= [-3,-2] \cup [2,3]. \end{aligned}$$

This is consistent with the property

$$f\left(f^{-1}(\mathbb{Y})\right) \subset \mathbb{Y}.$$

1.2 Interval arithmetic

If $\diamond \in \{+, -, ., /, \max, \min\}$

$$[x] \diamond [y] = [\{x \diamond y \mid x \in [x], y \in [y]\}] \,.$$

For instance,

$$\begin{aligned}
[-1,3] + [2,5] &= [?,?], \\
[-1,3].[2,5] &= [?,?], \\
[-1,3]/[2,5] &= [?,?], \\
[-1,3] \vee [2,5] &= [?,?].
\end{aligned}$$

If $\diamond \in \{+, -, ., /, \max, \min\}$

$$[x] \diamond [y] = [\{x \diamond y \mid x \in [x], y \in [y]\}] \,.$$

For instance,

$$\begin{array}{rl} [-1,3]+[2,5] & = [1,8], \\ [-1,3].[2,5] & = [-5,15], \\ [-1,3]/[2,5] & = [-\frac{1}{2}, \frac{3}{2}], \\ [-1,3] \vee [2,5] & = [2,5]. \end{array}$$

$$
\begin{aligned}
[x^-, x^+] + [y^-, y^+] = \quad & [x^- + y^-, x^+ + y^+], \\
[x^-, x^+].[y^-, y^+] = \quad & [x^-y^- \wedge x^+y^- \wedge x^-y^+ \wedge x^+y^+, \\
& x^-y^- \vee x^+y^- \vee x^-y^+ \vee x^+y^+], \\
[x^-, x^+] \vee [y^-, y^+] = \quad & [\vee(x^-, y^-), \vee(x^+, y^+)].
\end{aligned}
$$

If $f \in \{\text{cos, sin,sqr, sqrt, log, exp, } \ldots\}$

$$f([x]) = [\{f(x) \mid x \in [x]\}].$$

For instance,

$$\begin{aligned}
\mathsf{sin}([0,\pi]) &= ?,\\
\mathsf{sqr}([-1,3]) &= [-1,3]^2 = ?,\\
\mathsf{abs}([-7,1]) &= ?,\\
\mathsf{sqrt}([-10,4]) &= \sqrt{[-10,4]} = ?,\\
\mathsf{log}([-2,-1]) &= ?.
\end{aligned}$$

If $f \in \{\text{cos, sin,sqr, sqrt, log, exp, } \dots\}$

$$f\left([x]\right) = \left[\{f(x) \mid x \in [x]\}\right].$$

For instance,

$$\begin{array}{rcl}
\sin\left([0,\pi]\right) & = & [0,1], \\
\mathrm{sqr}\left([-1,3]\right) & = & [-1,3]^2 = [0,9], \\
\mathrm{abs}\left([-7,1]\right) & = & [0,7], \\
\mathrm{sqrt}\left([-10,4]\right) & = & \sqrt{[-10,4]} = [0,2], \\
\log\left([-2,-1]\right) & = & \emptyset.
\end{array}$$

1.3 Boxes

A *box*, or *interval vector* $[\mathbf{x}]$ of $\mathbb{R}^n$ is

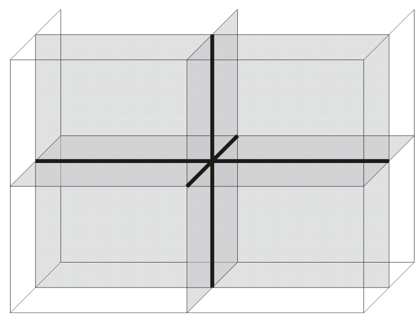
$$[\mathbf{x}] = [x_1^-, x_1^+] \times \cdots \times [x_n^-, x_n^+] = [x_1] \times \cdots \times [x_n].$$

The set of all boxes of $\mathbb{R}^n$ will be denoted by $\mathbb{IR}^n$.

The *width* $w([\mathbf{x}])$ of a box $[\mathbf{x}]$ is the length of its largest side. For instance

$$w([1,2] \times [-1,3]) = 4$$

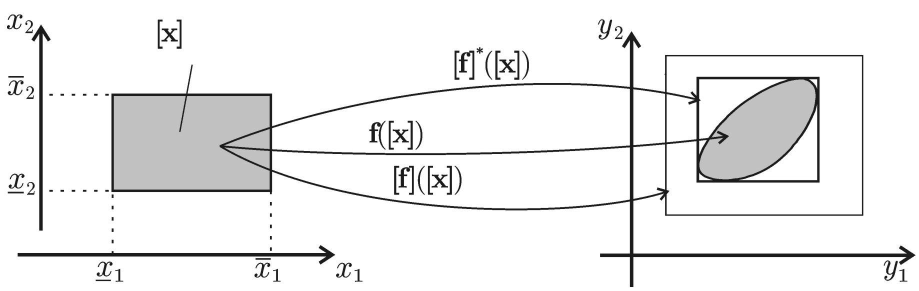
The *principal plane* of $[\mathbf{x}]$ is the symmetric plane $[\mathbf{x}]$ perpendicular to its largest side.



1.4 Inclusion function

The interval function $[\mathbf{f}]$ from $\mathbb{IR}^n$ to $\mathbb{IR}^m$, is an *inclusion function* of $\mathbf{f}$ if

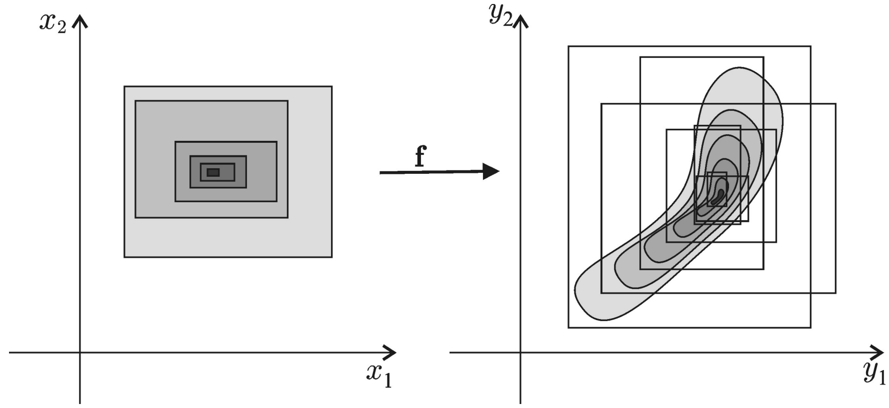
$$\forall\, [\mathbf{x}] \in \mathbb{IR}^n, \quad \mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}]\,([\mathbf{x}]).$$



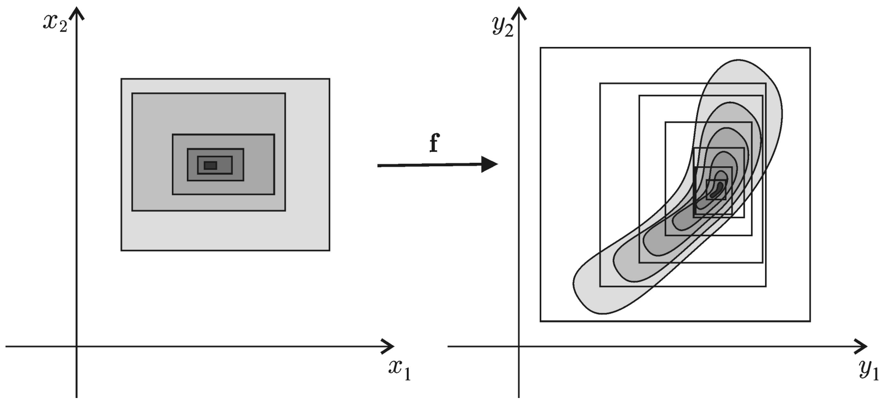
Inclusion functions $[\mathbf{f}]$ and $[\mathbf{f}]^*$; here, $[\mathbf{f}]^*$ is minimal.

The inclusion function $[\mathbf{f}]$ is

monotonic	if	$([\mathbf{x}] \subset [\mathbf{y}]) \Rightarrow ([\mathbf{f}]([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{y}]))$
minimal	if	$\forall [\mathbf{x}] \in \mathbb{IR}^n,\ [\mathbf{f}]([\mathbf{x}]) = [\mathbf{f}([\mathbf{x}])]$
thin	if	$w([\mathbf{x}]) = 0 \Rightarrow w([\mathbf{f}]([\mathbf{x}]) = 0$
convergent	if	$w([\mathbf{x}]) \to 0 \Rightarrow w([\mathbf{f}]([\mathbf{x}]) \to 0.$



Convergent but non-monotonic inclusion function



Convergent and monotonic inclusion function

The natural inclusion function for $f(x) = x^2 + 2x + 4$ is

$$[f]([x]) = [x]^2 + 2[x] + 4.$$

If $[x] = [-3, 4]$, we have

$$\begin{aligned} [f]([-3,4]) &= [-3,4]^2 + 2[-3,4] + 4 \\ &= [0,16] + [-6,8] + 4 \\ &= [-2,28]. \end{aligned}$$

Note that $f([-3,4]) = [3,28] \subset [f]([-3,4]) = [-2,28]$.

A minimal function for

$$\mathbf{f}: \begin{array}{lcl} \mathbb{R}^2 & \longrightarrow & \mathbb{R}^3 \\ (x_1, x_2) & \mapsto & \left(x_1 x_2, x_1^2, x_1 - x_2\right). \end{array}$$

is

$$[\mathbf{f}]: \begin{array}{lcl} \mathbb{IR}^2 & \longrightarrow & \mathbb{IR}^3 \\ ([x_1], [x_2]) & \longrightarrow & \left([x_1] * [x_2], [x_1]^2, [x_1] - [x_2]\right). \end{array}$$

$\mathbf{f}$ is given by the algorithm

Algorithm $\mathbf{f}$(in: $\mathbf{x} = (x_1, x_2, x_3)$, out: $\mathbf{y} = (y_1, y_2)$)

$z := x_1$;
for $k := 0$ to 100
    $z := x_2(z + kx_3)$;
next;
$y_1 := z$;
$y_2 := \sin(zx_1)$;

Its natural inclusion function is

Algorithm $[\mathbf{f}]$(in: $[\mathbf{x}]$, out: $[\mathbf{y}]$)

$[z] := [x_1]$;
for $k := 0$ to 100
    $[z] := [x_2] * ([z] + k * [x_3])$;
next;
$[y_1] := [z]$;
$[y_2] := \sin([z] * [x_1])$;

Here, $[\mathbf{f}]$ is a convergent, thin and monotonic inclusion function for $\mathbf{f}$.

1.5 Subpavings

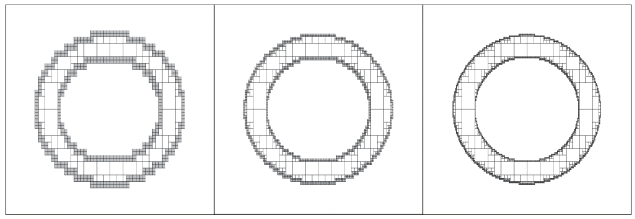
A subpaving of $\mathbb{R}^n$ is a set of non-overlapping boxes of $\mathbb{R}^n$.

Compact sets $\mathbb{X}$ can be bracketed between inner and outer subpavings:

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+.$$

Example.

$$\mathbb{X} = \{(x_1, x_2) \,\big|\, x_1^2 + x_2^2 \in [1, 2]\}.$$



Set operations such as $\mathbb{Z} := \mathbb{X} + \mathbb{Y}$, $\mathbb{X} := \mathbf{f}^{-1}(\mathbb{Y})$, $\mathbb{Z} := \mathbb{X} \cap \mathbb{Y} \dots$ can be approximated by subpaving operations.

1.6 Set inversion

Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ and let $\mathbb{Y}$ be a subset of $\mathbb{R}^m$. Set inversion is the characterization of

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\} = \mathbf{f}^{-1}(\mathbb{Y}).$$

We shall use the following tests.

$$\begin{array}{lll} \text{(i)} & [\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y} & \Rightarrow \; [\mathbf{x}] \subset \mathbb{X} \\ \text{(ii)} & [\mathbf{f}]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset & \Rightarrow \; [\mathbf{x}] \cap \mathbb{X} = \emptyset. \end{array}$$

Boxes for which these tests failed, will be bisected, except if they are too small.

Stack-queue

A *queue* is a list on which two operations are allowed :

- add an element at the end (*push*)

- remove the first element (*pull*).

A *stack* is a list on which two operations are allowed :

- add an element at the beginning of the list (*stack*)
- remove the first element (*pop*).

Example: Let $\mathcal{L}$ be an empty queue.

k	operation	result
0		$\mathcal{L} = \emptyset$
1	push $(\mathcal{L}, a)$	$\mathcal{L} = \{a\}$
2	push $(\mathcal{L}, b)$	$\mathcal{L} = \{a, b\}$
3	$x :=$ pull $(\mathcal{L})$	$x = a, \mathcal{L} = \{b\}$
4	$x :=$ pull $(\mathcal{L})$	$x = b, \mathcal{L} = \emptyset$.

If $\mathcal{L}$ is a stack, the table becomes

k	operation	result
0		$\mathcal{L} = \emptyset$
1	stack $(\mathcal{L}, a)$	$\mathcal{L} = \{a\}$
2	stack $(\mathcal{L}, b)$	$\mathcal{L} = \{a, b\}$
3	$x :=$ pop $(\mathcal{L})$	$x = b, \mathcal{L} = \{a\}$
4	$x :=$ pop $(\mathcal{L})$	$x = a, \mathcal{L} = \emptyset.$

.

Algorithm Sivia(in: $[\mathbf{x}](0), \mathbf{f}, \mathbb{Y}$)	
1	$\mathcal{L} := \{[\mathbf{x}](0)\}$;
2	pull $[\mathbf{x}]$ from $\mathcal{L}$;
3	if $[\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y}$, draw$([\mathbf{x}]$, 'red');
4	elseif $[\mathbf{f}]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset$, draw$([\mathbf{x}]$, 'blue');
5	elseif $w([\mathbf{x}]) < \varepsilon$, {draw ($[\mathbf{x}]$, 'yellow')};
6	else bisect $[\mathbf{x}]$ and push into $\mathcal{L}$;
7	if $\mathcal{L} \neq \emptyset$, go to 2

If $\boldsymbol{\Delta}\mathbb{X}$ denotes the union of yellow boxes and if $\mathbb{X}^-$ is the union of red boxes then :

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^- \cup \boldsymbol{\Delta}\mathbb{X}.$$

1.7 Image evaluation

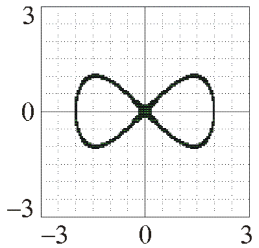
Define

$$\mathbf{f}\left(x_1, x_2\right) = \left(\begin{array}{c} \left(x_1 - 1\right)^2 - 1 + x_2 \\ -x_1^2 + \left(x_2 - 1\right)^2 \end{array}\right),$$

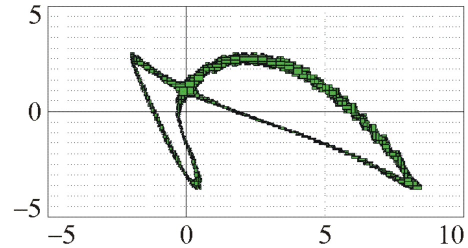
and

$$\mathbb{X}_1 = \left\{\left(x_1, x_2\right) \in \mathbb{R}^2 \,\middle|\, x_1^4 - x_1^2 + 4x_2^2 \in [-0.1, 0.1]\right\}.$$

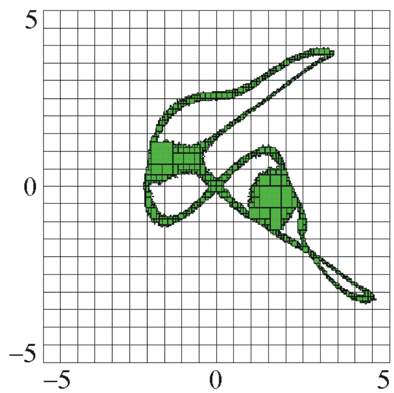
We shall compute $\mathbb{X}_1$, $\mathbf{f}\left(\mathbb{X}_1\right)$ and $\mathbf{f}^{-1} \circ \mathbf{f}\left(\mathbb{X}_1\right)$.



(a): $\mathbb{X}_1$



(b): $\mathbf{f}(\mathbb{X}_1)$



(c): $\mathbf{f}^{-1}(\mathbf{f}(\mathbb{X}_1))$

2 Applications of set computation

2.1 Bounded-error estimation

Model : $\phi(\mathbf{p}, t) = p_1 e^{-p_2 t}$.

Prior feasible box for the parameters : $[\mathbf{p}] \subset \mathbb{R}^2$

Measurement times : $t_1, t_2, \ldots, t_m$

Data bars : $[y_1^-, y_1^+], [y_2^-, y_2^+], \ldots, [y_m^-, y_m^+]$

$\mathbb{S} = \{\mathbf{p} \in [\mathbf{p}], \phi(\mathbf{p}, t_1) \in [y_1^-, y_1^+], \ldots, \phi(\mathbf{p}, t_m) \in [y_m^-, y_m^+]\}$.

If

$$\boldsymbol{\phi}(\mathbf{p}) = \left(\begin{array}{c} \phi(\mathbf{p}, t_1) \\ \\ \phi(\mathbf{p}, t_m) \end{array} \right)$$

and

$$[\mathbf{y}] = [y_1^-, y_1^+] \times \cdots \times [y_m^-, y_m^+]$$

then

$$\mathbb{S} = [\mathbf{p}] \cap \boldsymbol{\phi}^{-1}([\mathbf{y}]).$$

Show Setdemo (Guillaume Baffet), available at

`http://www.istia.univ-angers.fr/~jaulin/demo.html`

2.2 Robustification against outliers

Define a *relaxing function* for the box $[\mathbf{y}] = [y_1] \times \cdots \times [y_n]$

$$\lambda(\mathbf{y}) = \pi_{[y_1]}(y_1) + \cdots + \pi_{[y_n]}(y_n)$$

where

$$\pi_{[a,b]}(x) \begin{cases} = 1 & \text{if} \quad x \in [a, b] \\ = 0 & \text{if} \quad x \notin [a, b]. \end{cases}$$

Allow up to q of the n output variables y_i to escape their prior feasible intervals. The posterior feasible set becomes

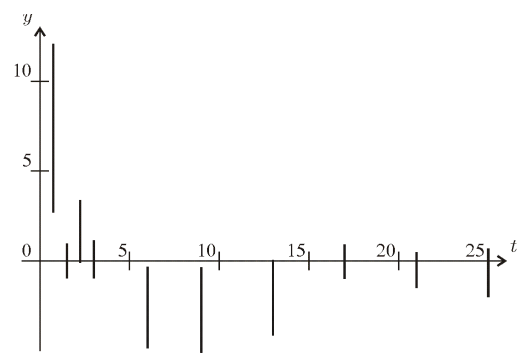
$$\hat{\mathbb{P}}_q = \{\mathbf{p} \in [\mathbf{p}] \mid \pi_{[y_1]}(\phi_{\mathbf{1}}(\mathbf{p})) + \cdots + \pi_{[y_n]}(\phi_n(\mathbf{p})) \geqslant n - q\}.$$

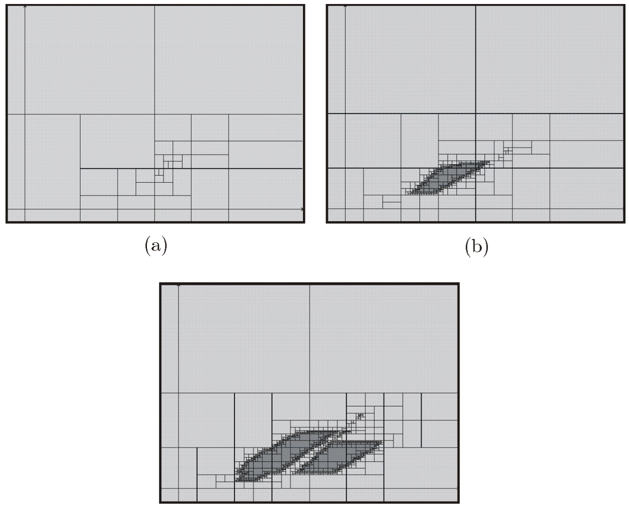
This is a set inversion problem. The set $\hat{\mathbb{P}}_q$ can thus be characterized by Sivia.

As an illustration, consider the model

$$\phi(\mathbf{p}, t) = 20\,\exp(-p_1 t) - 8\,\exp(-p_2 t)$$

with the data bars represented on the figure below





(a) (b) (c)

(a) no outlier assumed; (b) one outlier assumed; (c) two outliers assumed;

2.3 Robust stability

The *stability domain* $\mathbb{S}_p$ of the polynomial

$$P(s, \mathbf{p}) = s^n + a_{n-1}(\mathbf{p})s^{n-1} + \ldots + a_1(\mathbf{p})s + a_0(\mathbf{p})$$

is the set of all $\mathbf{p}$ such that $P(s, \mathbf{p})$ is stable.

If $P(s, \mathbf{p})$ is given by

$$s^3+(p_1+p_2+2)s^2+(p_1+p_2+2)s+2p_1p_2+6p_1+6p_2+2.25,$$

Its Routh table is given by

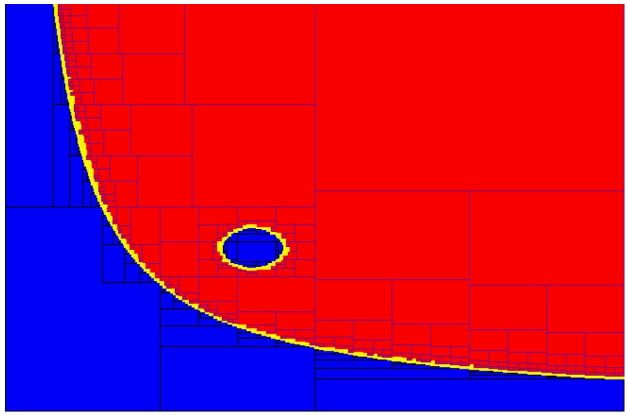
1	$p_1 + p_2 + 2$
$p_1 + p_2 + 2$	$2p_1p_2 + 6p_1 + 6p_2 + 2.25$
$\frac{(p_1-1)^2+(p_2-1)^2-0.25}{p_1+p_2+2}$	0
$2(p_1 + 3)(p_2 + 3) - 15.75$	0

Its stability domain is thus defined by

$$\mathbb{S}_{\mathsf{p}} \triangleq \{\mathbf{p} \in \mathbb{R}^n \mid \mathbf{r}(\mathbf{p}) > \mathbf{0}\} = \mathbf{r}^{-1}\left(]0, +\infty[^{\times n}\right).$$

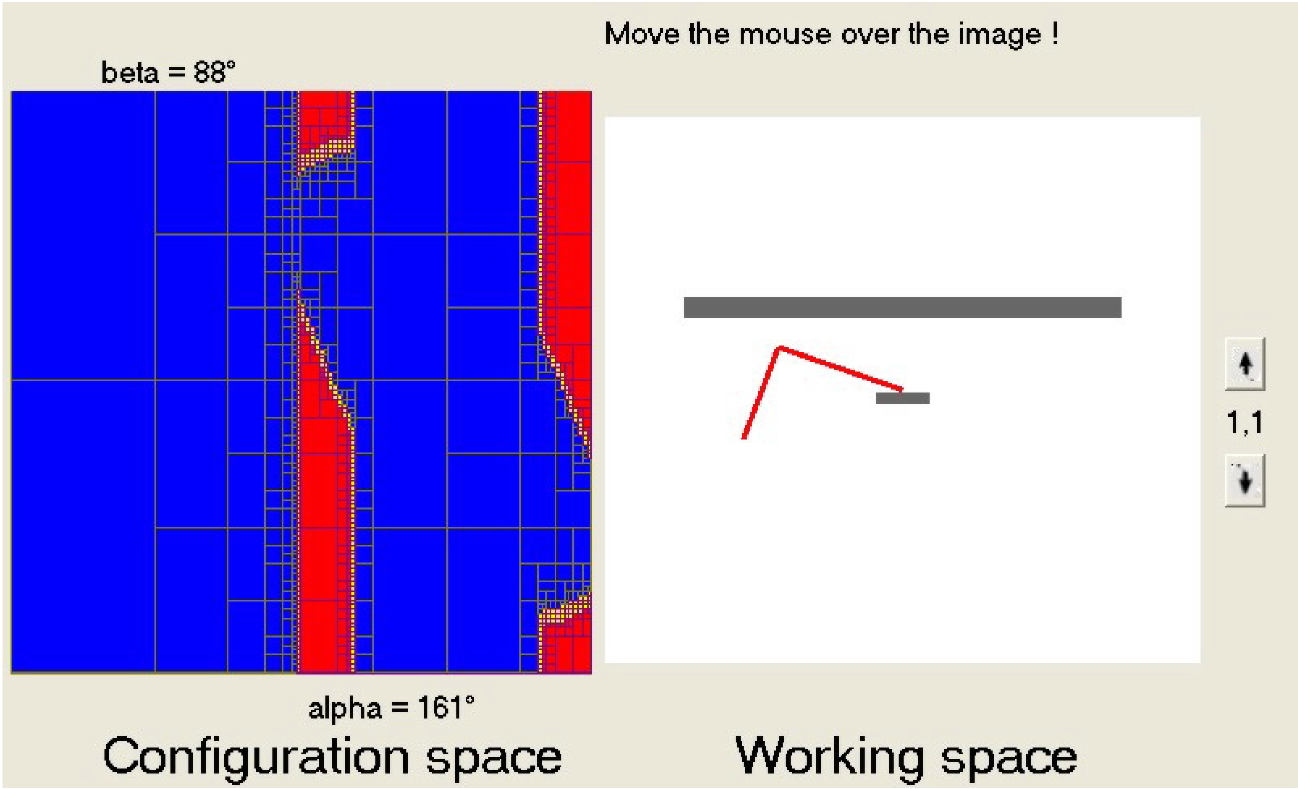
where

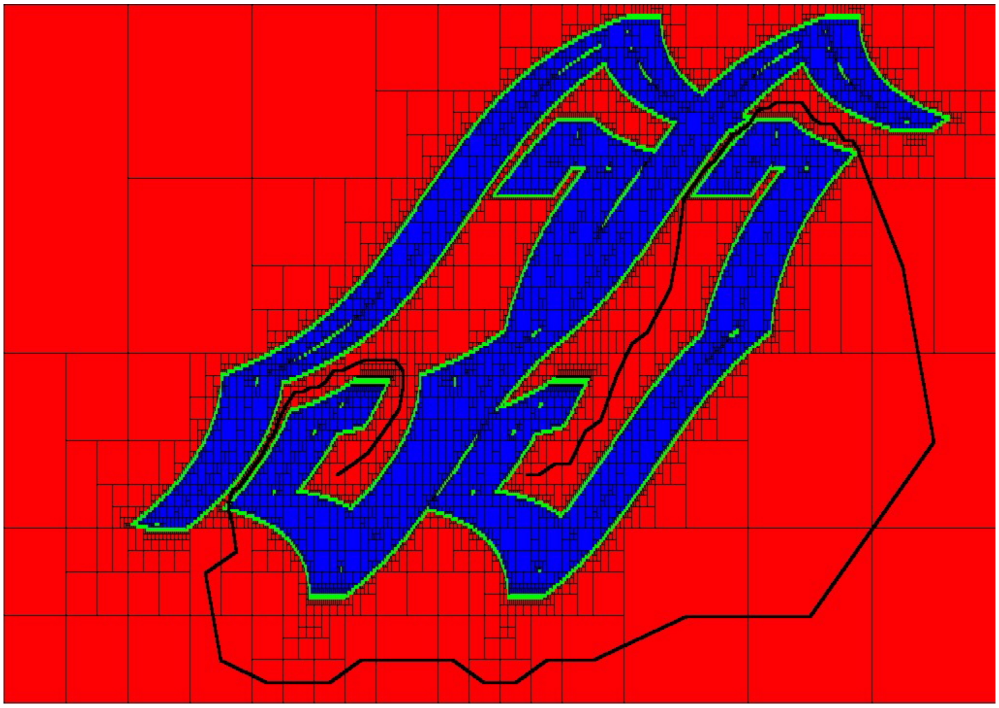
$$\mathbf{r}(\mathbf{p}) = \begin{pmatrix} p_1 + p_2 + 2 \\ (p_1 - 1)^2 + (p_2 - 1)^2 - 0.25 \\ 2(p_1 + 3)(p_2 + 3) - 15.75 \end{pmatrix}.$$

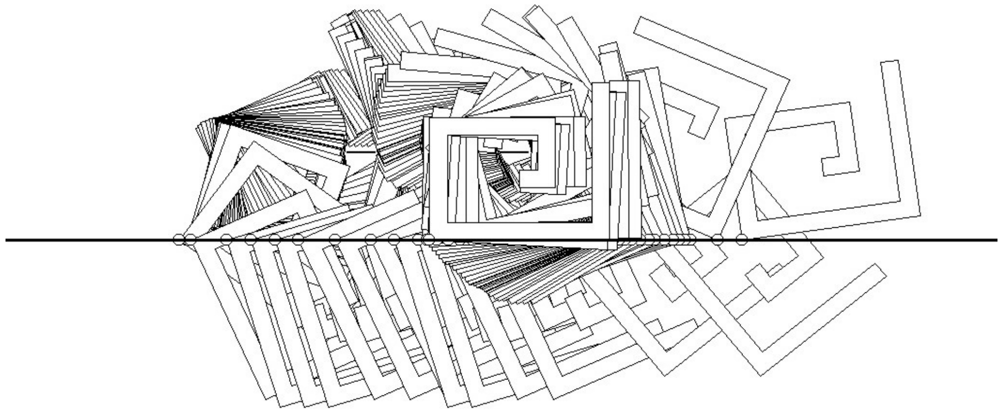


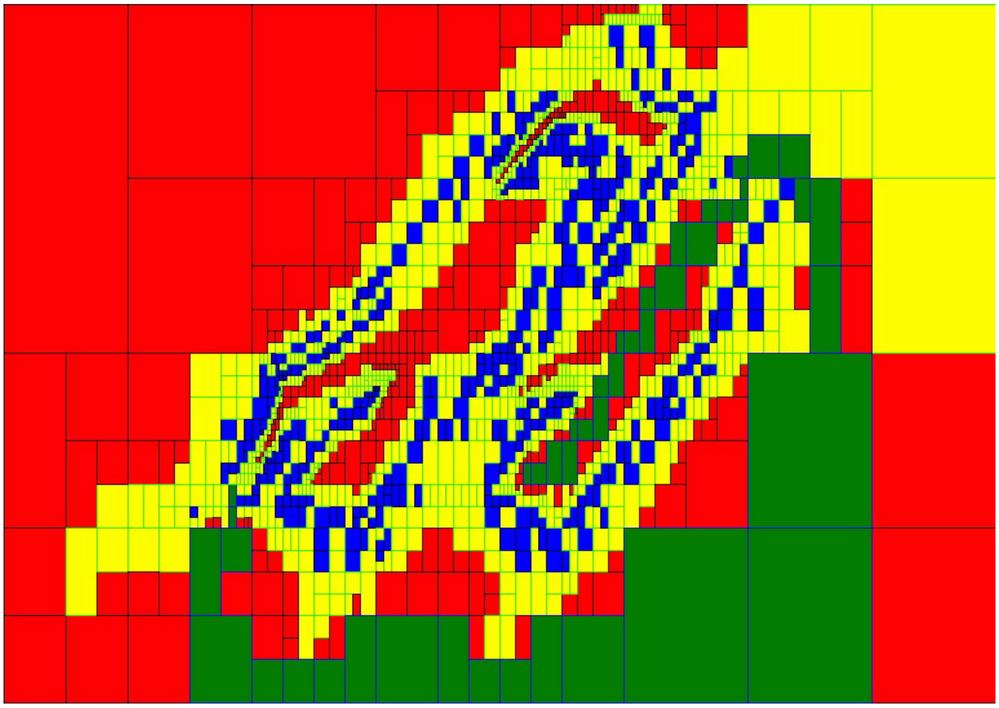
Stability domain $\mathbb{S}_{\mathrm{p}}$ generated by Proj2d

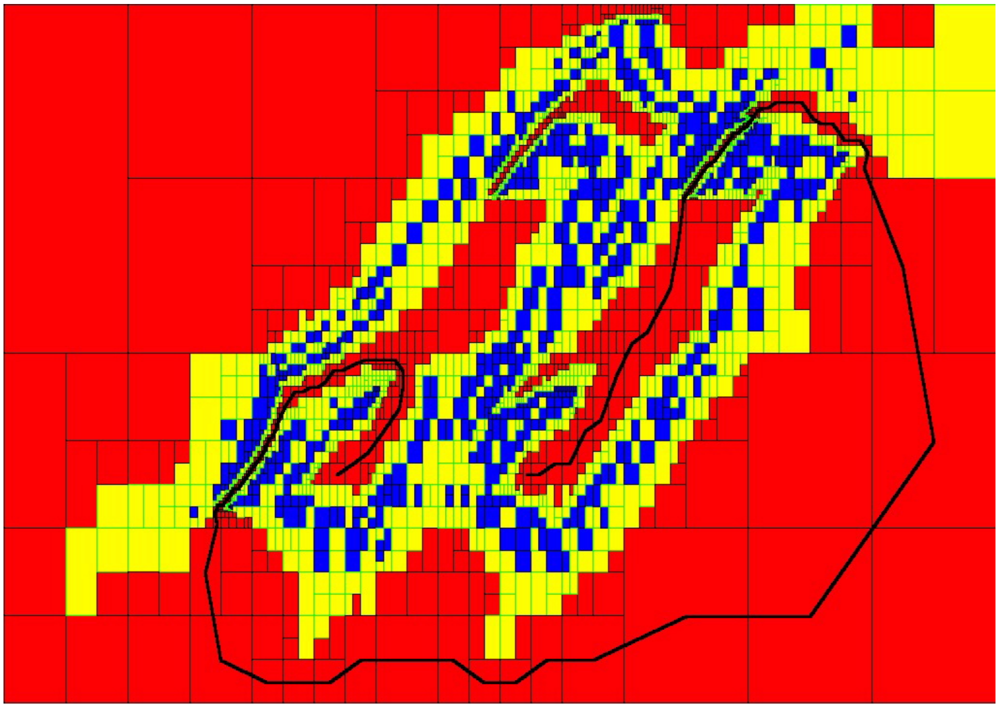
2.4 Path planning


Move the mouse over the image !
beta = 88°
1,1
alpha = 161°
Configuration space
Working space









2.5 Sailboat

State equations

$$\begin{cases} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta - 1 \\ \dot{\theta} &= \omega \\ \dot{\delta}_s &= u_1 \\ \dot{\delta}_r &= u_2 \\ \dot{v} &= f_s \sin\delta_s - f_r \sin\delta_r - v \\ \dot{\omega} &= (1-\cos\delta_s) f_s - \cos\delta_r . f_r - \omega \\ f_s &= \cos(\theta+\delta_s) - v\sin\delta_s \\ f_r &= v\sin\delta_r . \end{cases}$$

In a cruising phase

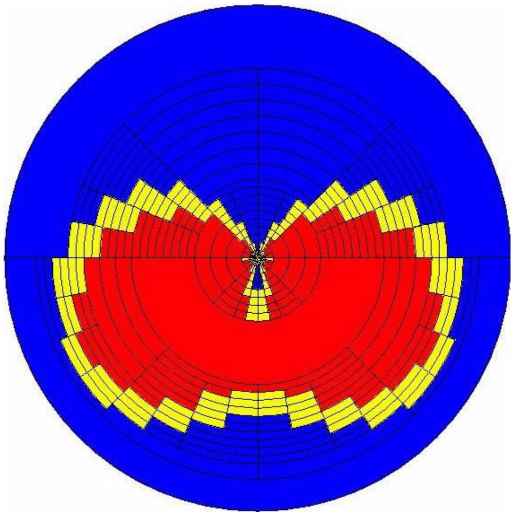
$$\dot{\theta} = 0, \dot{\delta}_s = 0, \dot{\delta}_r = 0, \dot{v} = 0, \dot{\omega} = 0.$$

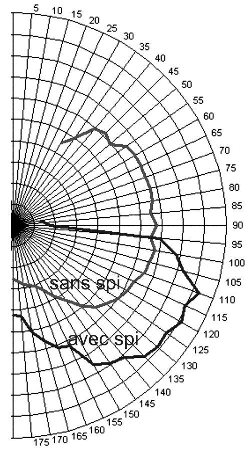
i.e.,

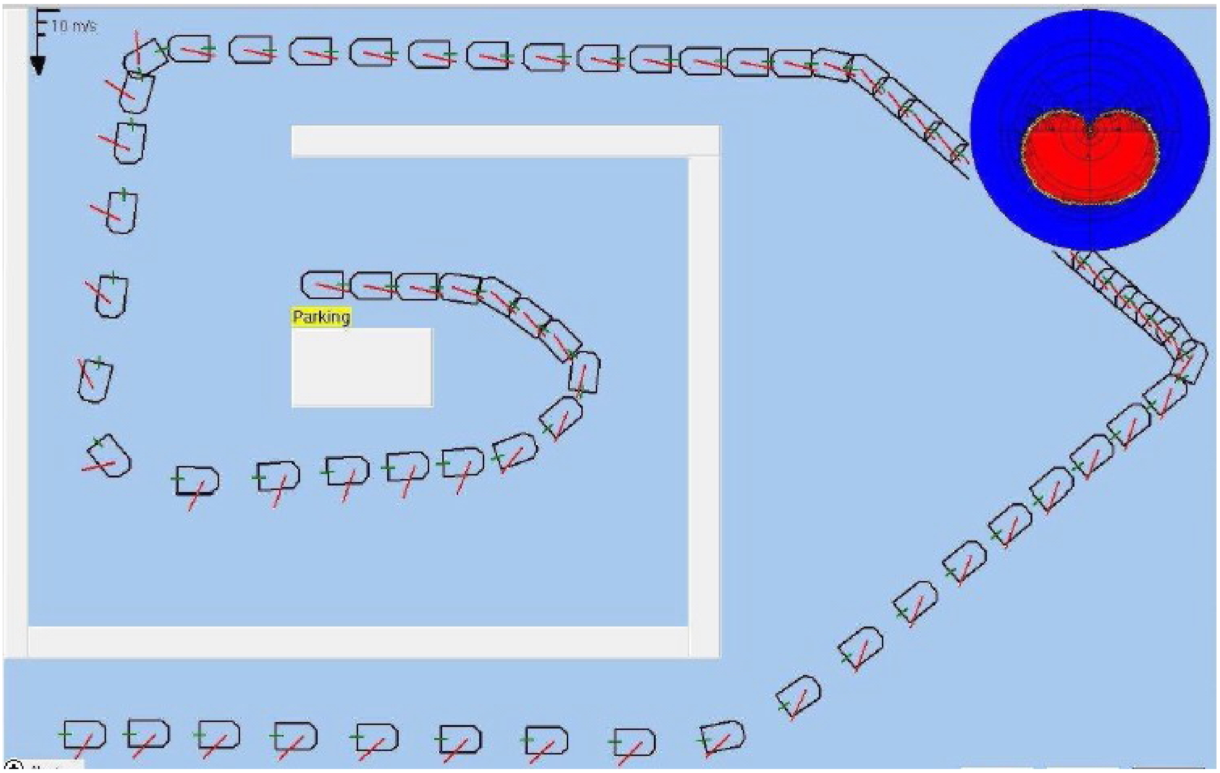
$$\left\{ \begin{array}{rcl} 0 & = & \omega \\ 0 & = & u_1 \\ 0 & = & u_2 \\ 0 & = & f_s \sin \delta_s - f_r \sin \delta_r - v \\ 0 & = & (1 - \cos \delta_s) f_s - \cos \delta_r . f_r - \omega \\ f_s & = & \cos(\theta + \delta_s) - v \sin \delta_s \\ f_r & = & v \sin \delta_r . \end{array} \right.$$

The polar diagram is

$$\begin{aligned}\mathbb{S}_y = \; \{(\theta, v) \quad & \mid \exists f_s, \delta_s, f_r, \delta_r, v, \\ & f_s \sin\delta_s - f_r \sin\delta_r - v = 0 \\ & (1 - \cos\delta_s)\, f_s - \cos\delta_r f_r = 0 \\ & f_s = \cos(\theta + \delta_s) - v \sin\delta_s \\ & f_r = v \sin\delta_r \;\}\end{aligned}$$




5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
sans spi
avec spi


10 m/s
Parking

3 Contractors

To characterize $\mathbb{X} \subset \mathbb{R}^n$, bisection algorithms bisect all boxes in all directions and become inefficient. Interval methods can still be useful if

- the solution set $\mathbb{X}$ is small (optimization problem, solving equations),

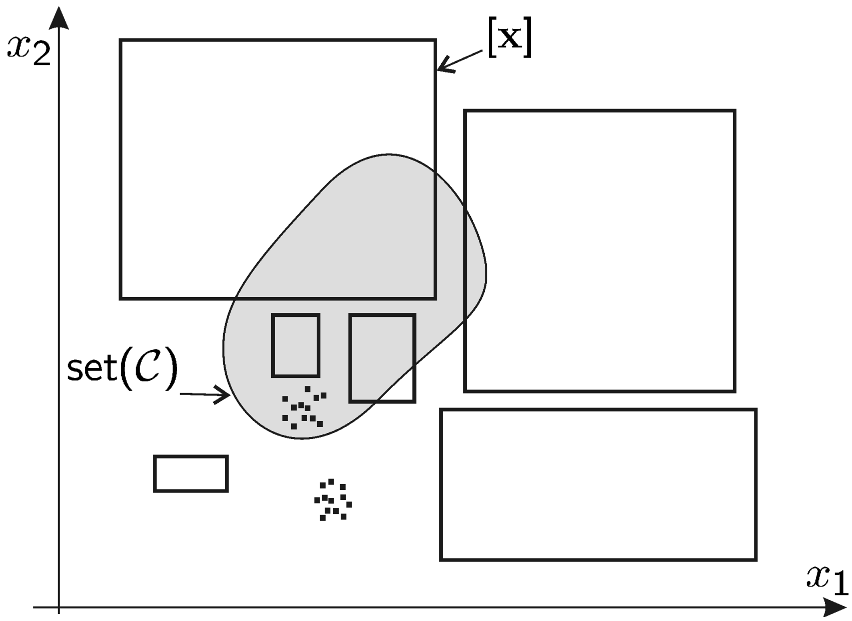
- contraction procedures are used as much as possible,

- bisections are used only as a last resort.

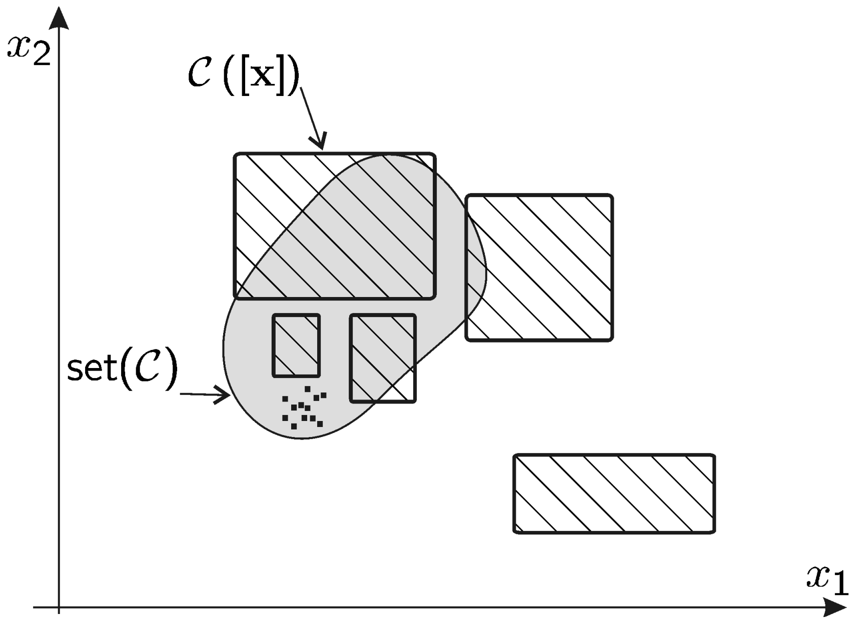
3.1 Definition

The operator $\mathcal{C}_{\mathbb{X}} : \mathbb{IR}^n \rightarrow \mathbb{IR}^n$ is a *contractor* for $\mathbb{X} \subset \mathbb{R}^n$ if

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \left\{ \begin{array}{ll} \mathcal{C}_{\mathbb{X}}([\mathbf{x}]) \subset [\mathbf{x}] & \text{(contractance),} \\ \mathcal{C}_{\mathbb{X}}([\mathbf{x}]) \cap \mathbb{X} = [\mathbf{x}] \cap \mathbb{X} & \text{(completeness).} \end{array} \right.$$



x_2
$[\mathbf{x}]$
set($\mathcal{C}$)
x_1



x_2
$\mathcal{C}([\mathbf{x}])$
$\mathsf{set}(\mathcal{C})$
x_1

$\mathcal{C}_{\mathbb{X}}$ is *monotonic* if	$[\mathbf{x}] \subset [\mathbf{y}] \Rightarrow \mathcal{C}_{\mathbb{X}}([\mathbf{x}]) \subset \mathcal{C}_{\mathbb{X}}([\mathbf{y}])$
$\mathcal{C}_{\mathbb{X}}$ is *minimal* if	$\forall[\mathbf{x}] \in \mathbb{IR}^n,\ \mathcal{C}_{\mathbb{X}}([\mathbf{x}]) = [[\mathbf{x}] \cap \mathbb{X}]$
$\mathcal{C}_{\mathbb{X}}$ is *thin* if	$\forall\mathbf{x} \in \mathbb{R}^n,\ \mathcal{C}_{\mathbb{X}}(\{\mathbf{x}\}) = \{\mathbf{x}\} \cap \mathbb{X}$
$\mathcal{C}_{\mathbb{X}}$ is *idempotent* if	$\forall[\mathbf{x}] \in \mathbb{IR}^n, \mathcal{C}_{\mathbb{X}}\left(\mathcal{C}_{\mathbb{X}}([\mathbf{x}])\right) = \mathcal{C}_{\mathbb{X}}([\mathbf{x}])$.

$\mathcal{C}_{\mathbb{X}}$ is said to be *convergent* if

$$[\mathbf{x}](k) \to \mathbf{x} \quad \Rightarrow \quad \mathcal{C}_{\mathbb{X}}([\mathbf{x}]\,(k)) \to \{\mathbf{x}\} \cap \mathbb{X}.$$

3.2 Projection of constraints

Let x, y, z be 3 variables such that

$$\begin{aligned} x &\in [-\infty, 5], \\ y &\in [-\infty, 4], \\ z &\in [6, \infty], \\ z &= x + y. \end{aligned}$$

The values < 2 for x, < 1 for y and > 9 for z are inconsistent.

To *project* a constraint (here, $z = x + y$), is to compute the smallest intervals which contains all consistent values.

For our example, this amounts to project onto x, y and z the set

$$\mathbb{S} = \{(x, y, z) \in [1, 5] \times [2, 4] \times [6, 10] \mid z = x + y\} .$$

3.3 Numerical method for projection

Since $x \in [-\infty, 5], y \in [-\infty, 4], z \in [6, \infty]$ and $z = x + y$, we have

$$
\begin{aligned}
z = x + y \Rightarrow \quad z \in \quad & [6, \infty] \cap ([-\infty, 5] + [-\infty, 4]) \\
& = [6, \infty] \cap [-\infty, 9] = [6, 9]. \\
x = z - y \Rightarrow \quad x \in \quad & [-\infty, 5] \cap ([6, \infty] - [-\infty, 4]) \\
& = [-\infty, 5] \cap [2, \infty] = [2, 5]. \\
y = z - x \Rightarrow \quad y \in \quad & [-\infty, 4] \cap ([6, \infty] - [-\infty, 5]) \\
& = [-\infty, 4] \cap [1, \infty] = [1, 4].
\end{aligned}
$$

The contractor associated with $z = x + y$ is.

Algorithm pplus(inout: $[z], [x], [y]$)
$[z] := [z] \cap ([x] + [y])\,;$
$[x] := [x] \cap ([z] - [y])\,;$
$[y] := [y] \cap ([z] - [x])\,.$

The projection procedure developed for plus can be extended to other ternary constraints such as mult: $z = x * y$, or equivalently

$$\text{mult} \triangleq \left\{(x, y, z) \in \mathbb{R}^3 \mid z = x * y\right\}.$$

The resulting projection procedure becomes

Algorithm pmult(inout: $[z], [x], [y]$)
1 $\quad [z] := [z] \cap ([x] * [y])\,;$
2 $\quad [x] := [x] \cap ([z] * 1/[y])\,;$
3 $\quad [y] := [y] \cap ([z] * 1/[x])\,.$

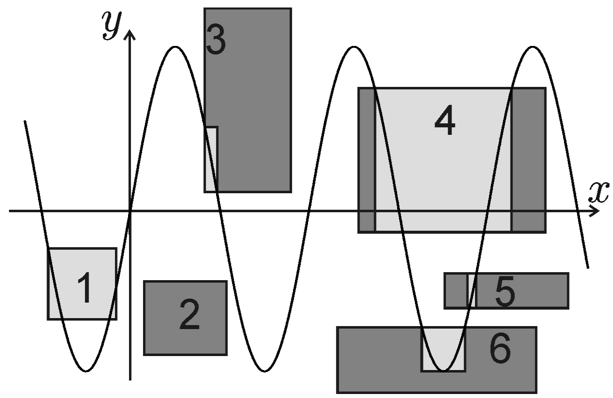
Consider the binary constraint

$$\mathsf{exp} \triangleq \{(x, y) \in \mathbb{R}^n | y = \exp(x)\}.$$

The associated contractor is

Algorithm pexp(inout: $[y], [x]$)
1 $[y] := [y] \cap \exp([x])$;
2 $[x] := [x] \cap \log([y])$.

Any constraint for which such a projection procedure is available will be called a *primitive constraint*.



Projection of the sine constraint

3.4 Constraint propagation

A CSP (Constraint Satisfaction Problem) is composed of

1) a set of variables $\mathcal{V} = \{x_1, \ldots, x_n\}$,
2) a set of constraints $\mathcal{C} = \{c_1, \ldots, c_m\}$ and
3) a set of interval domains $\{[x_1], \ldots, [x_n]\}$.

Principle of propagation techniques: contract $[\mathbf{x}] = [x_1] \times \cdots \times [x_n]$ as follows:

$$((((((([\mathbf{x}] \sqcap c_1) \sqcap c_2) \sqcap \ldots) \sqcap c_m) \sqcap c_1) \sqcap c_2) \ldots,$$

until a steady box is reached.

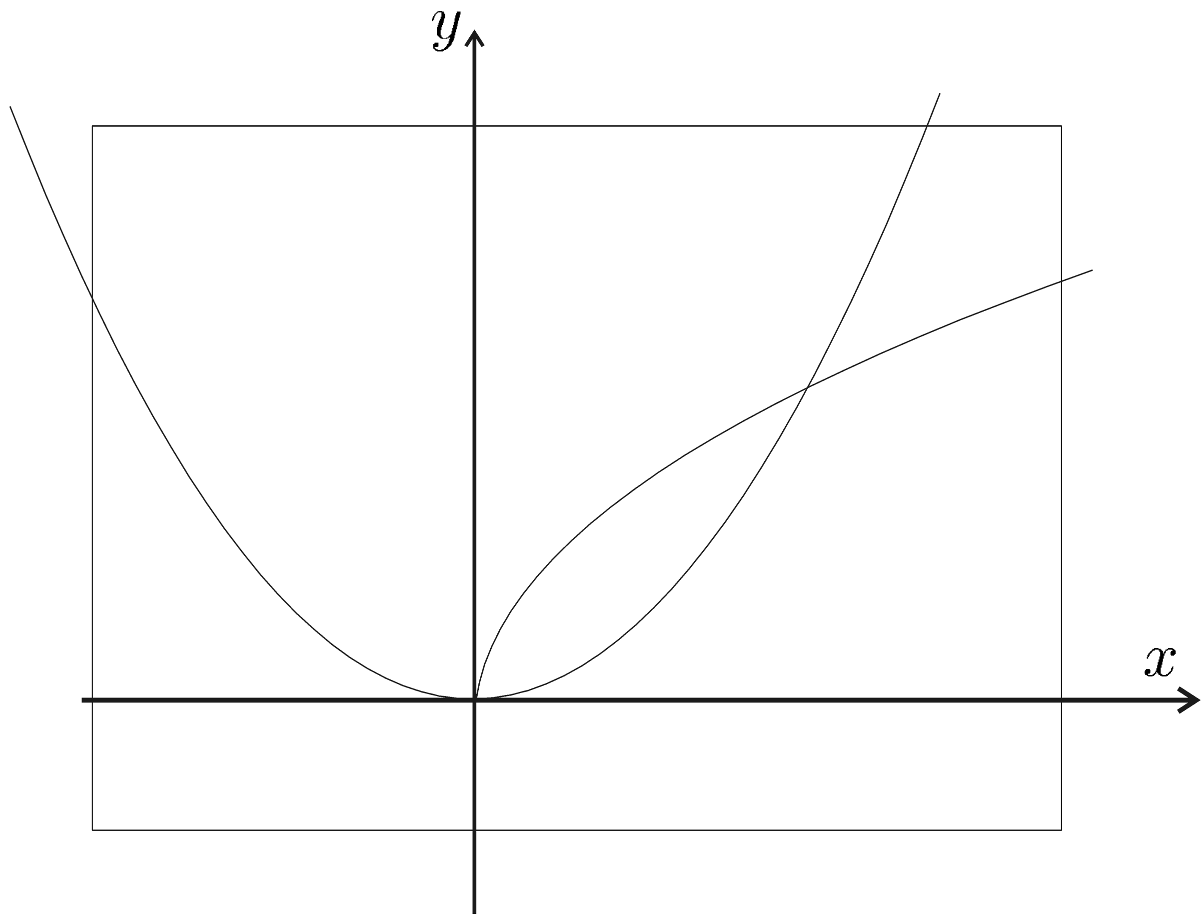
Example. Consider the system of two equations.

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

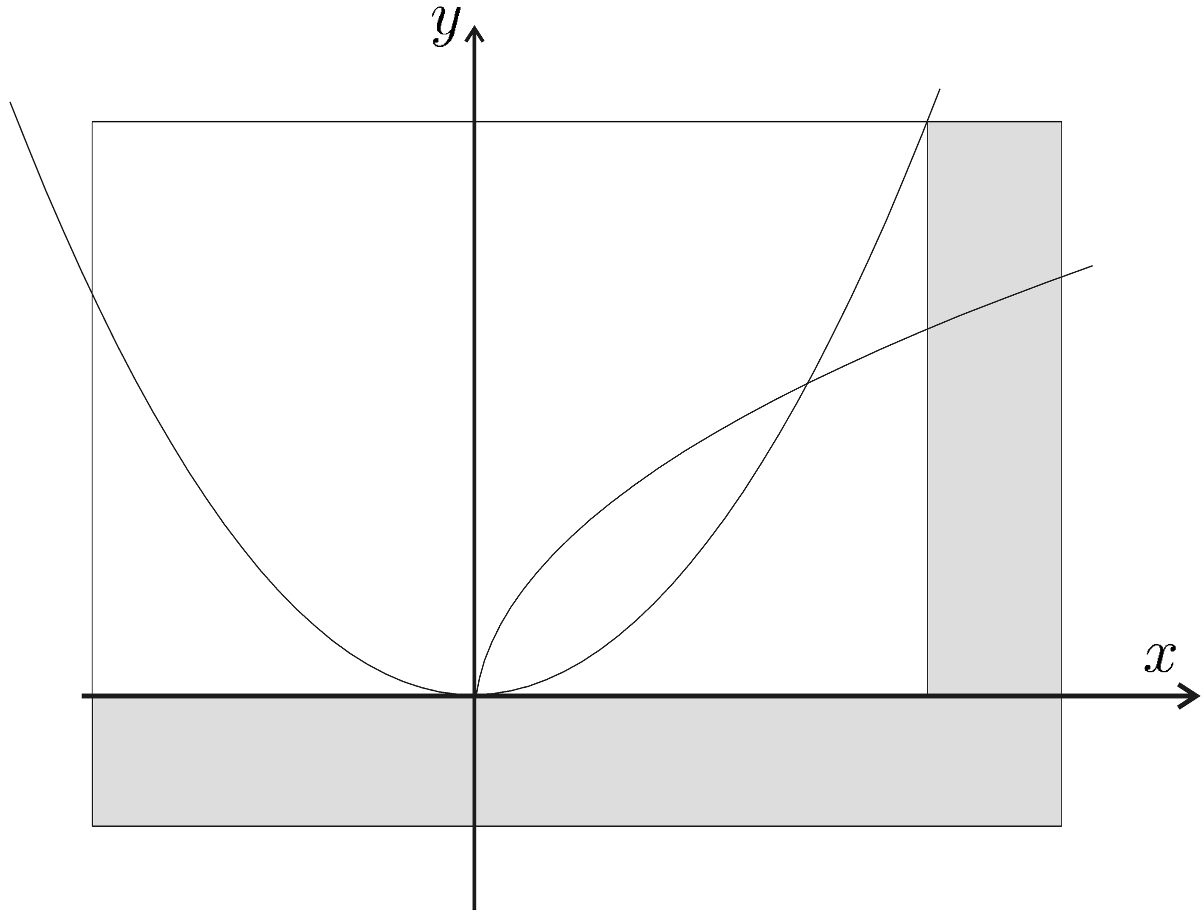
We can build two contractors

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

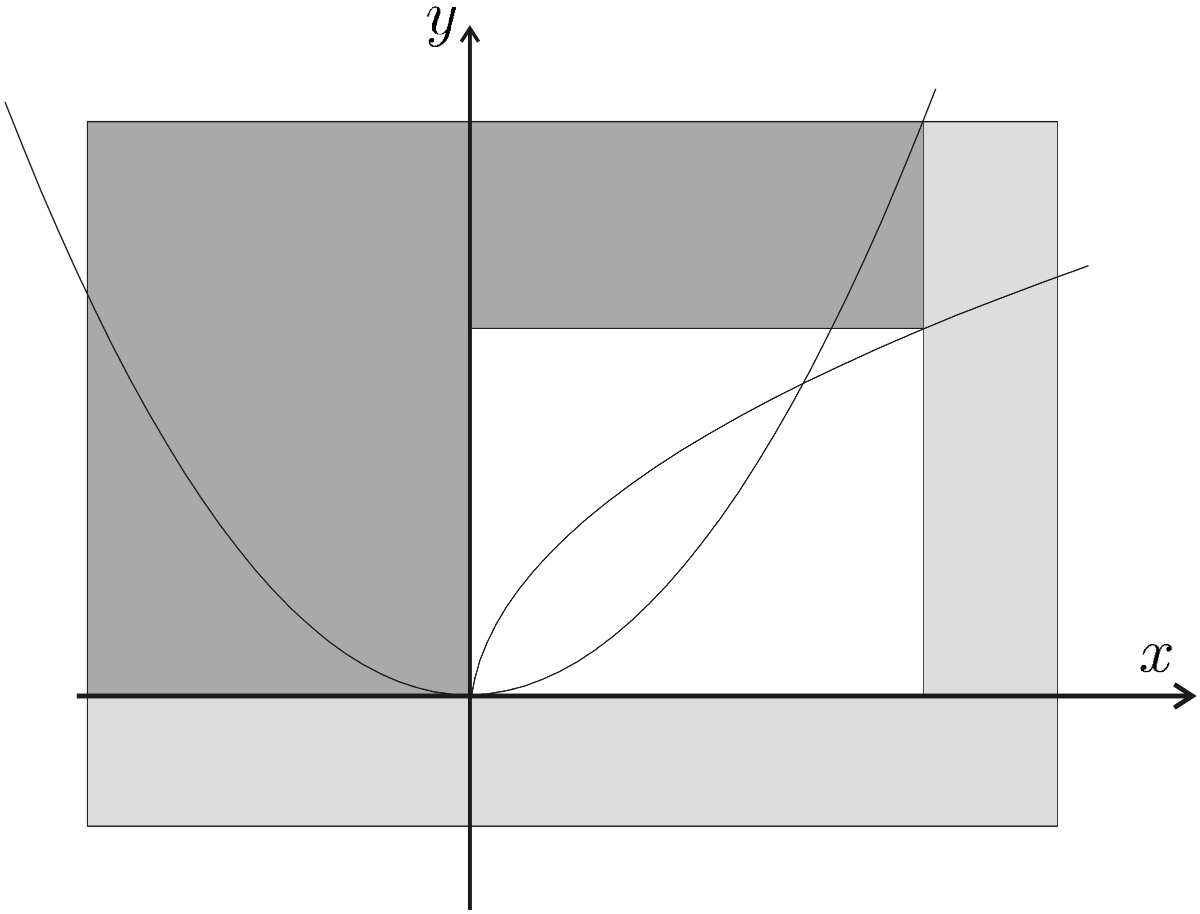
$$\mathcal{C}_2 : \begin{cases} [y] = [y] \cap \sqrt{[x]} \\ [x] = [x] \cap [y]^2 \end{cases} \quad \text{associated to } y = \sqrt{x}$$



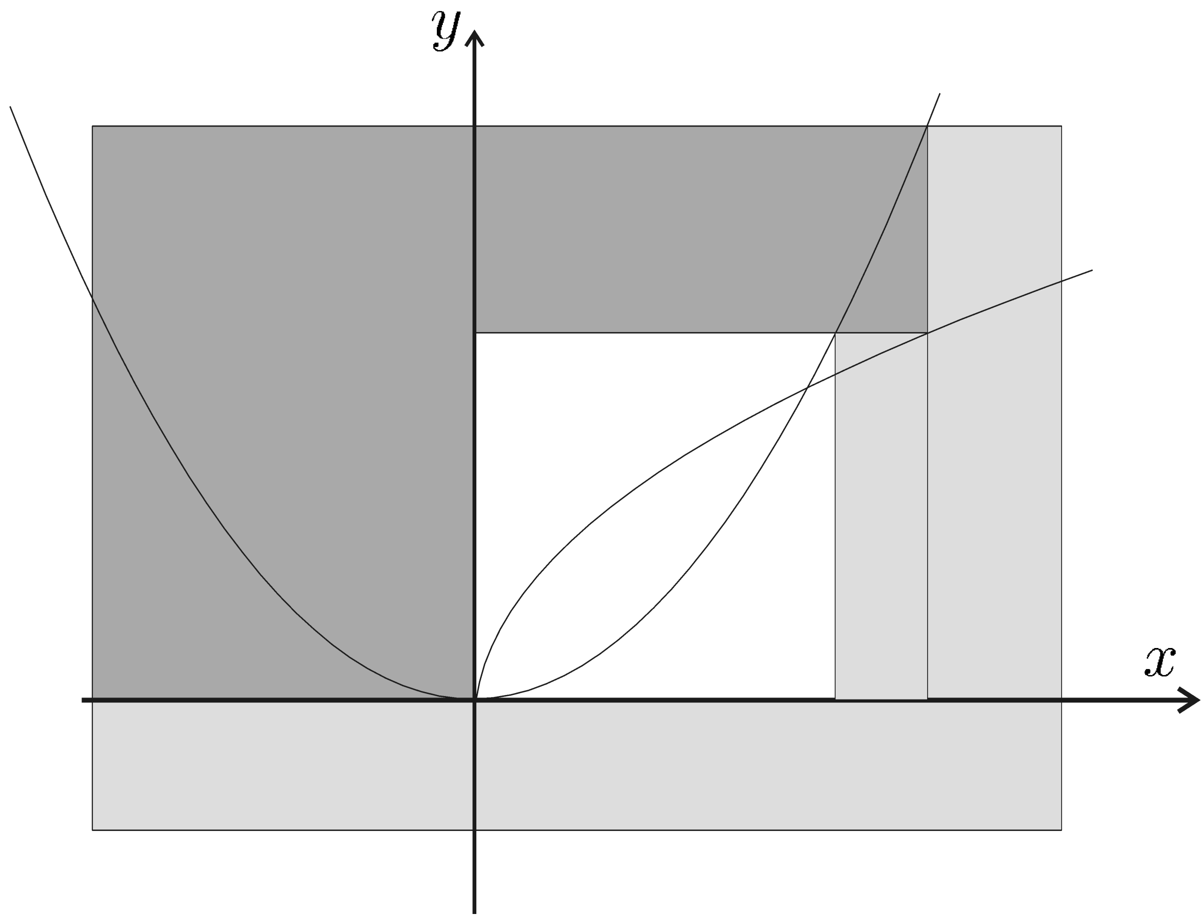
y
x



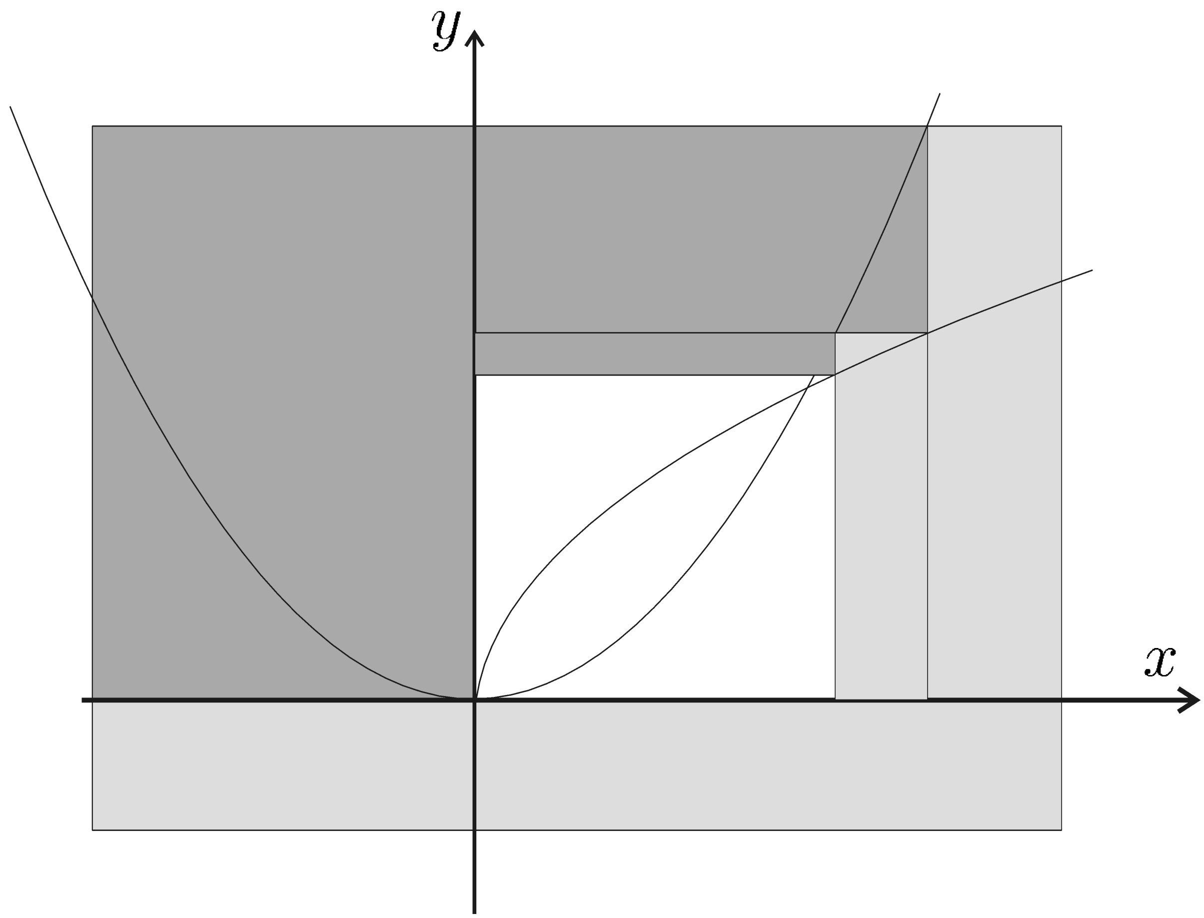
y
x

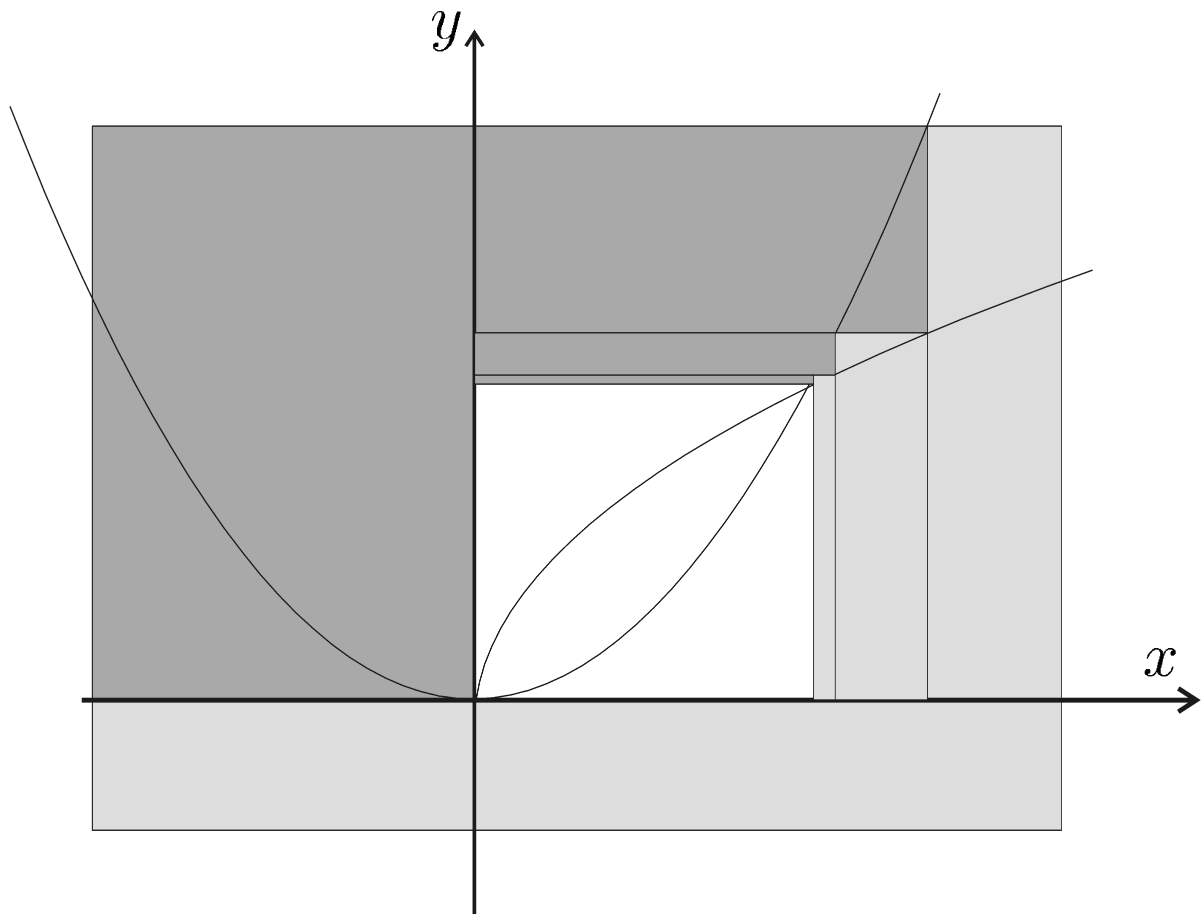


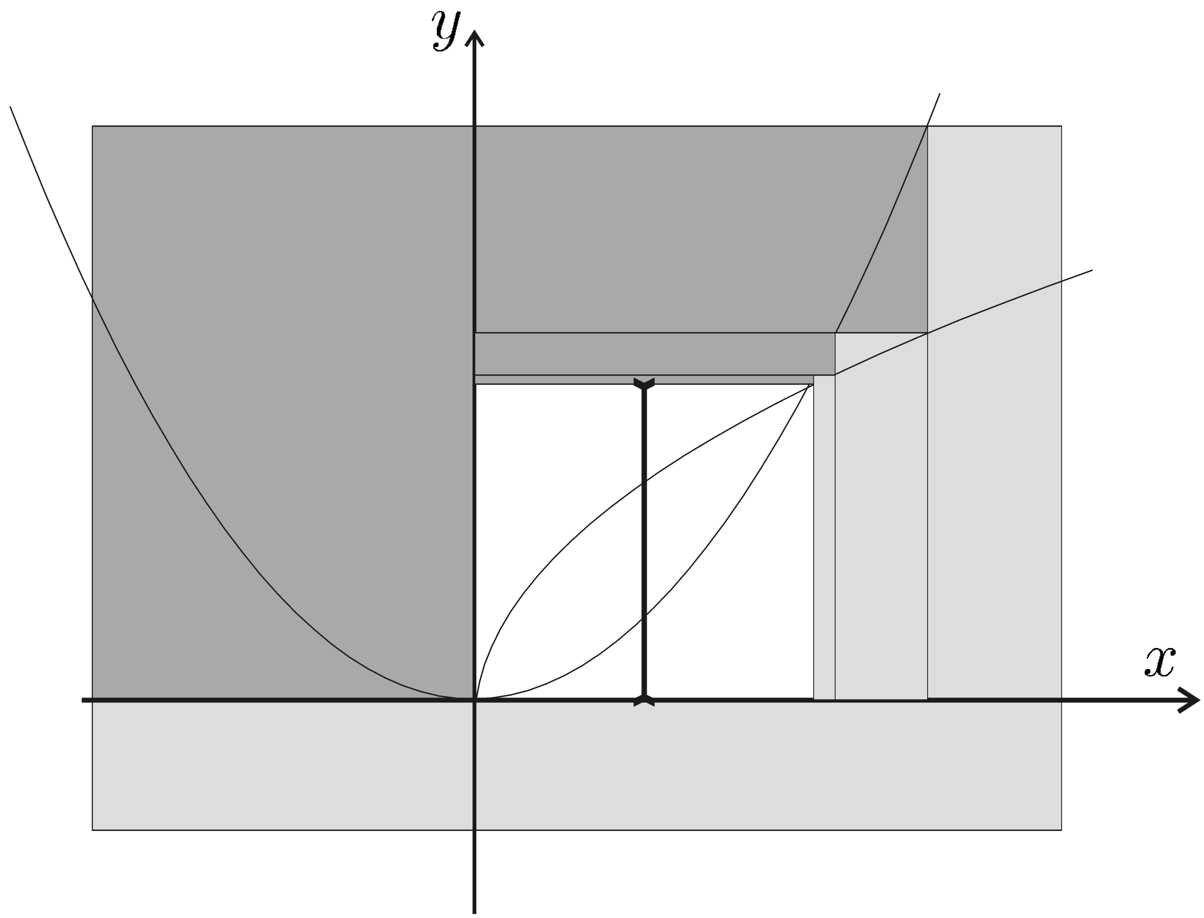
y
x



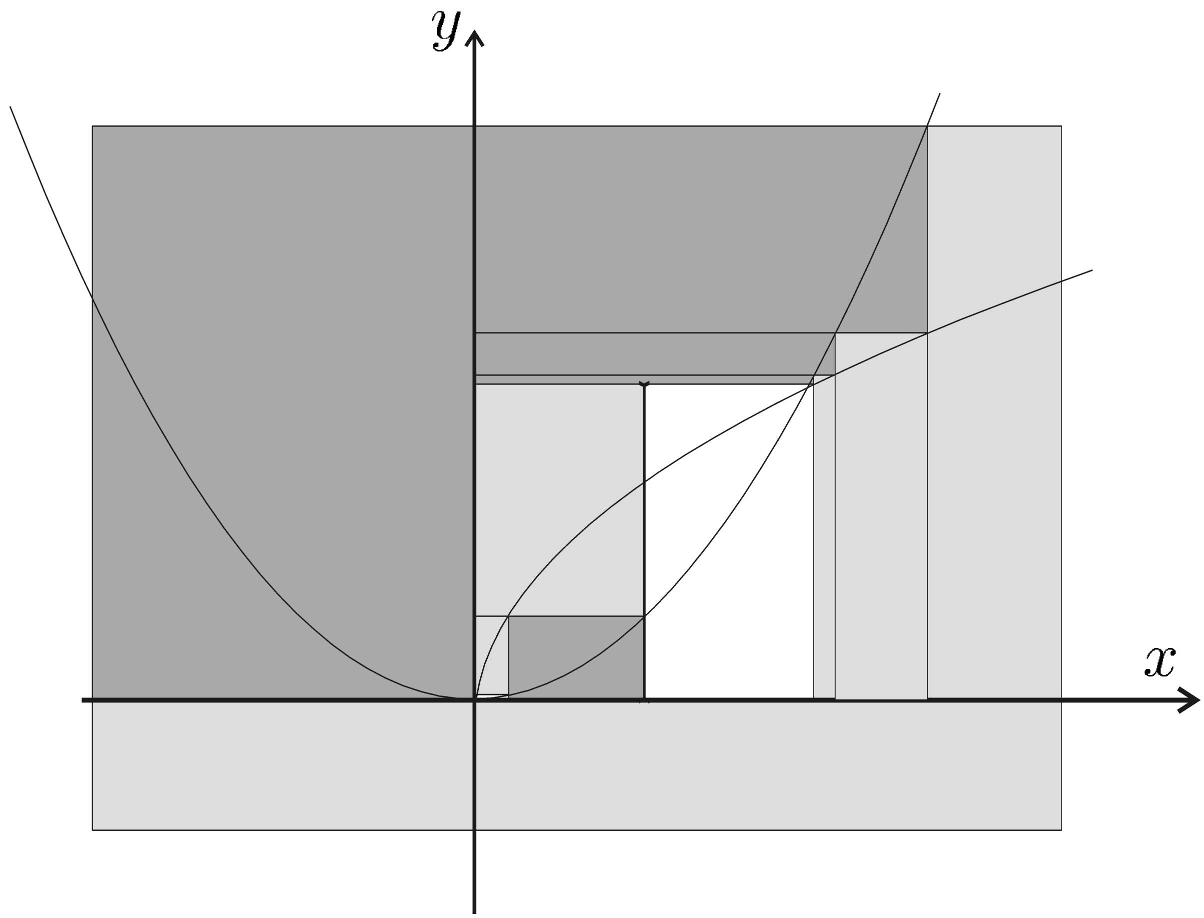
y
x

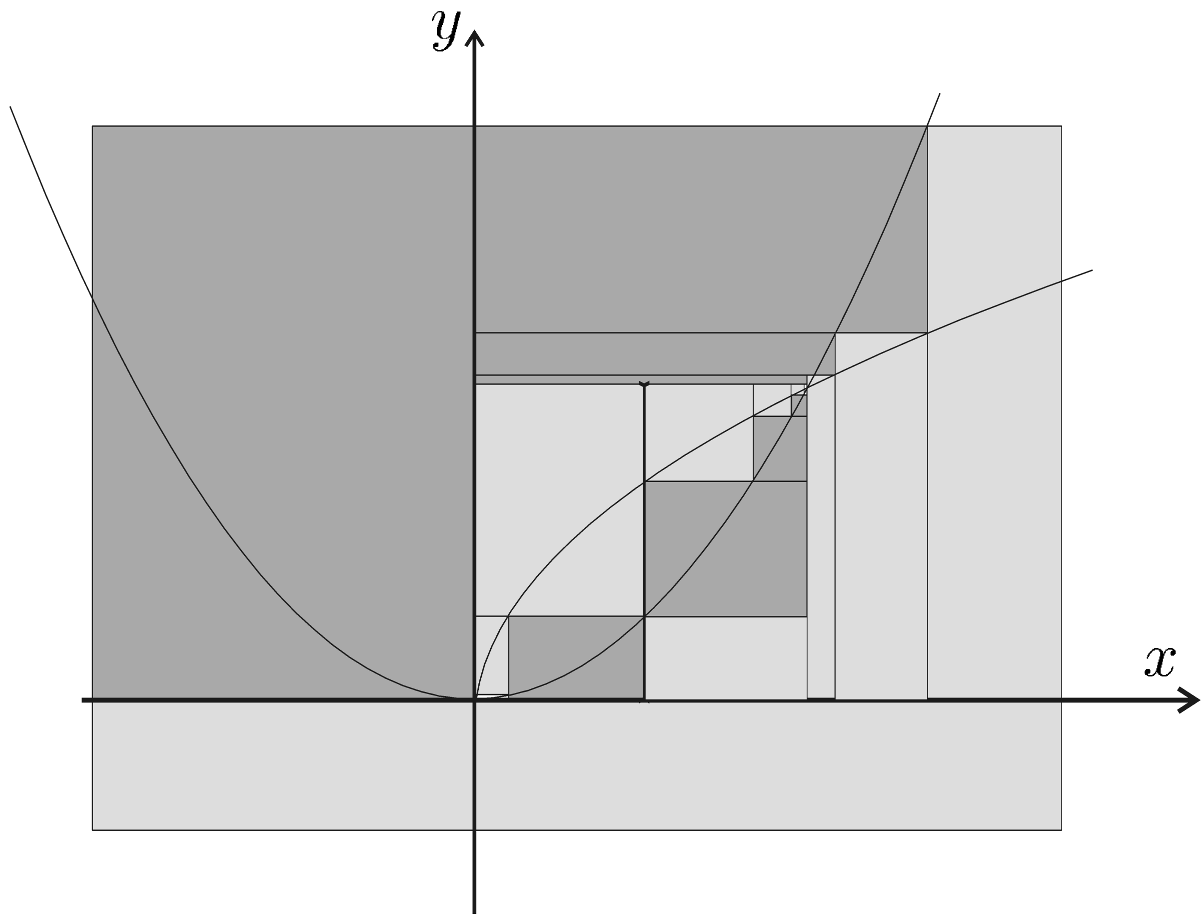

y
x


y
x



y
x


y
x



y
x

3.5 Local consistency

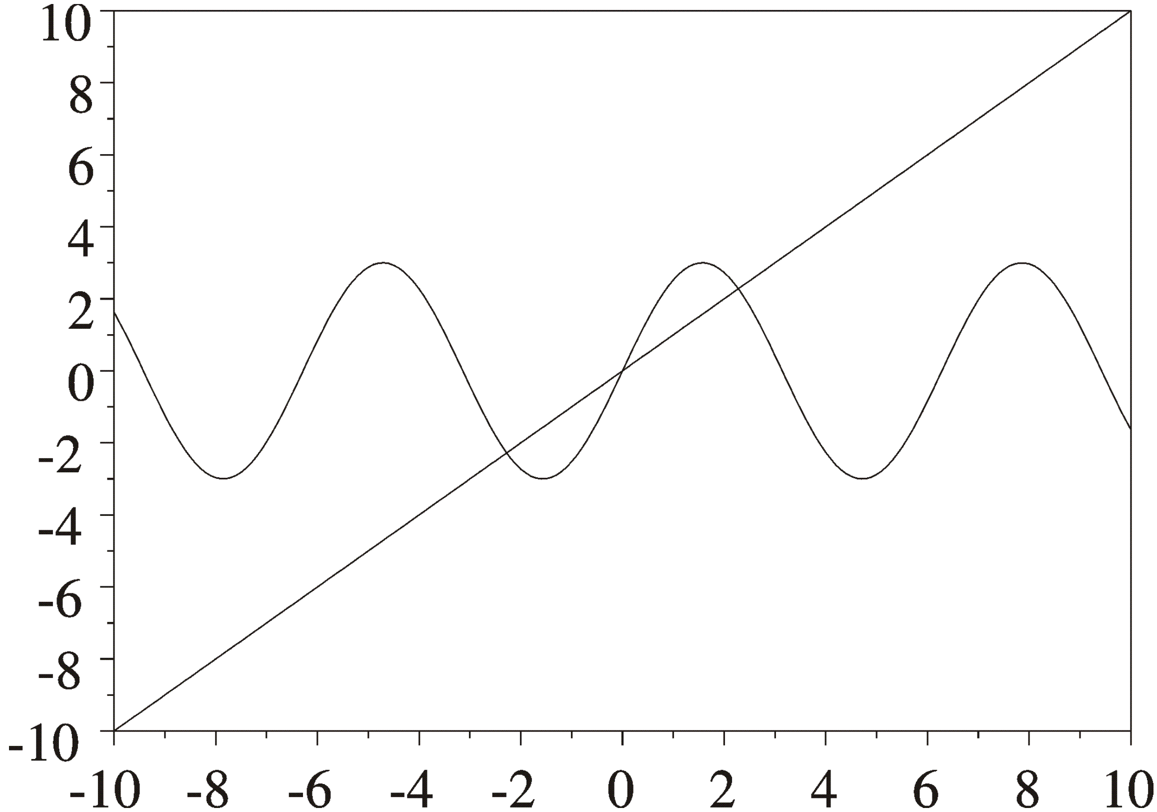
If $\mathcal{C}^*_{\mathbb{S}_1}$ and $\mathcal{C}^*_{\mathbb{S}_2}$ are two minimal contractors for $\mathbb{S}_1$ and $\mathbb{S}_2$ then

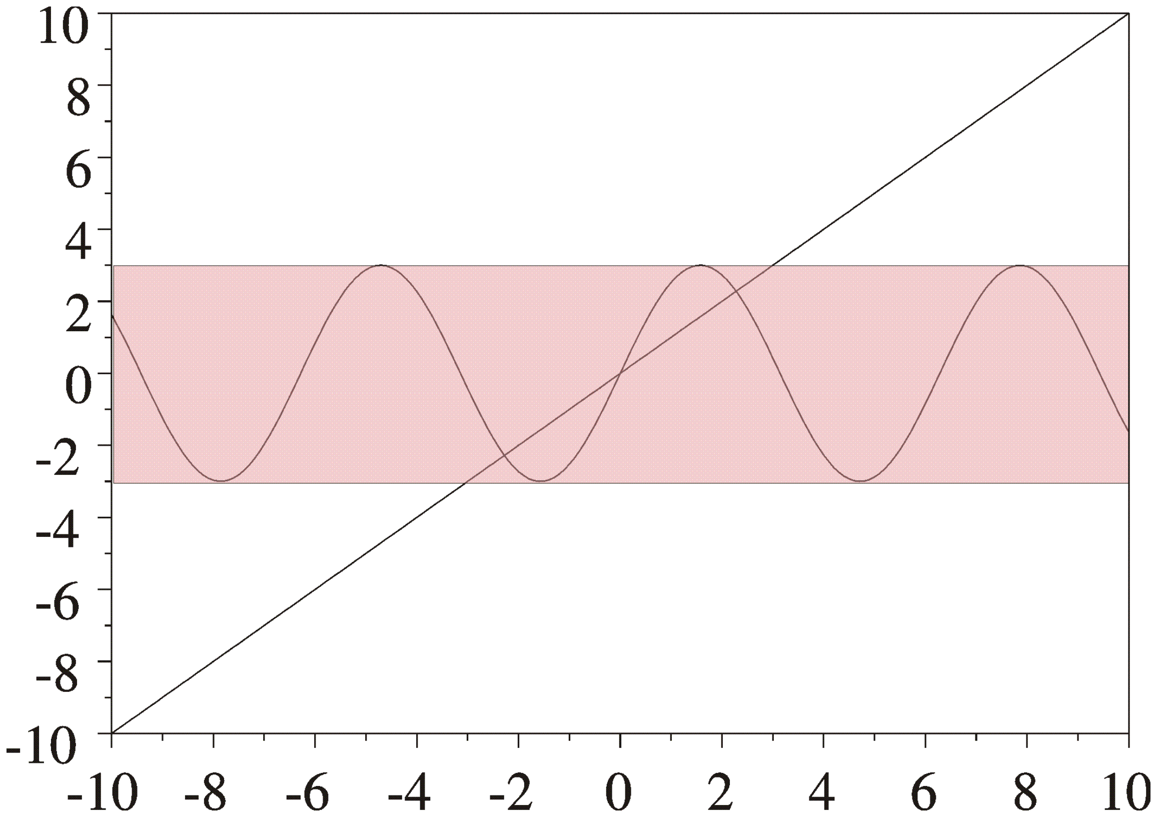
$$\mathcal{C}_{\mathbb{S}} = \mathcal{C}^*_{\mathbb{S}_1} \circ \mathcal{C}^*_{\mathbb{S}_2} \circ \mathcal{C}^*_{\mathbb{S}_1} \circ \mathcal{C}^*_{\mathbb{S}_2} \circ \dots$$

is a contractor for $\mathbb{S} = \mathbb{S}_1 \cap \mathbb{S}_2$, but it is not always optimal. This is the *local consistency effect.*

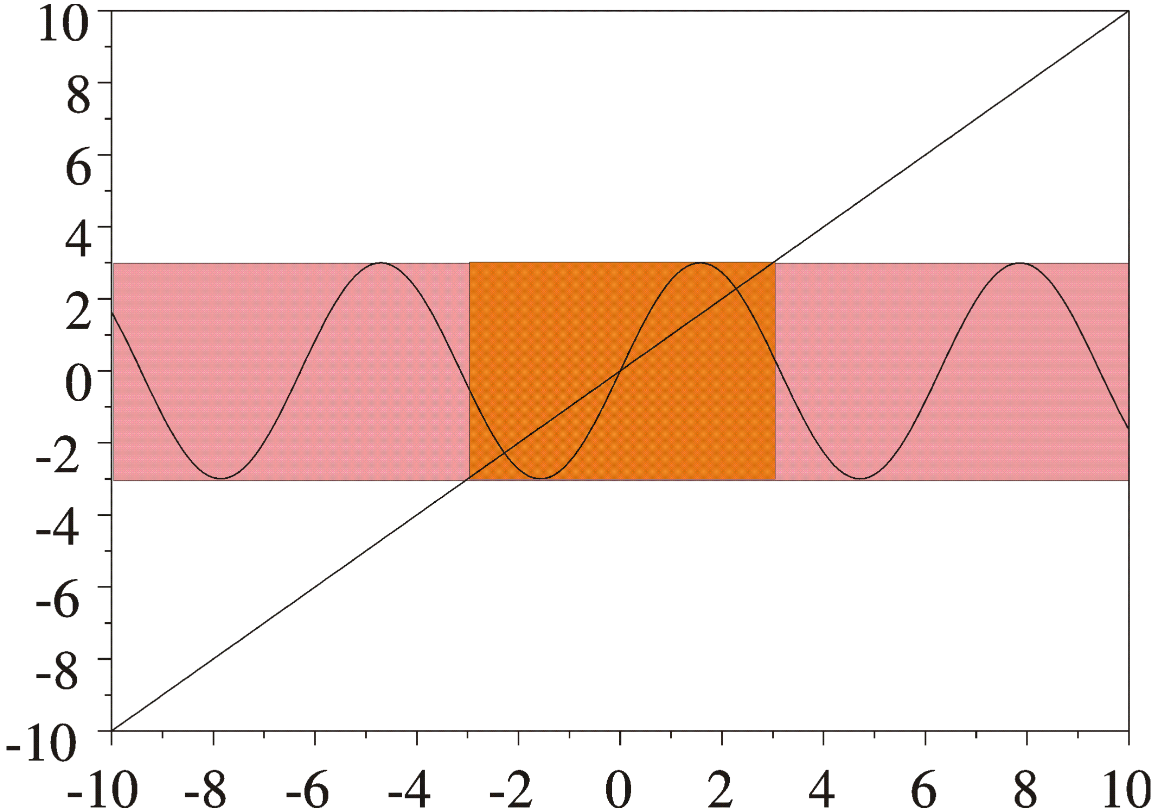
Exemple. Consider the system

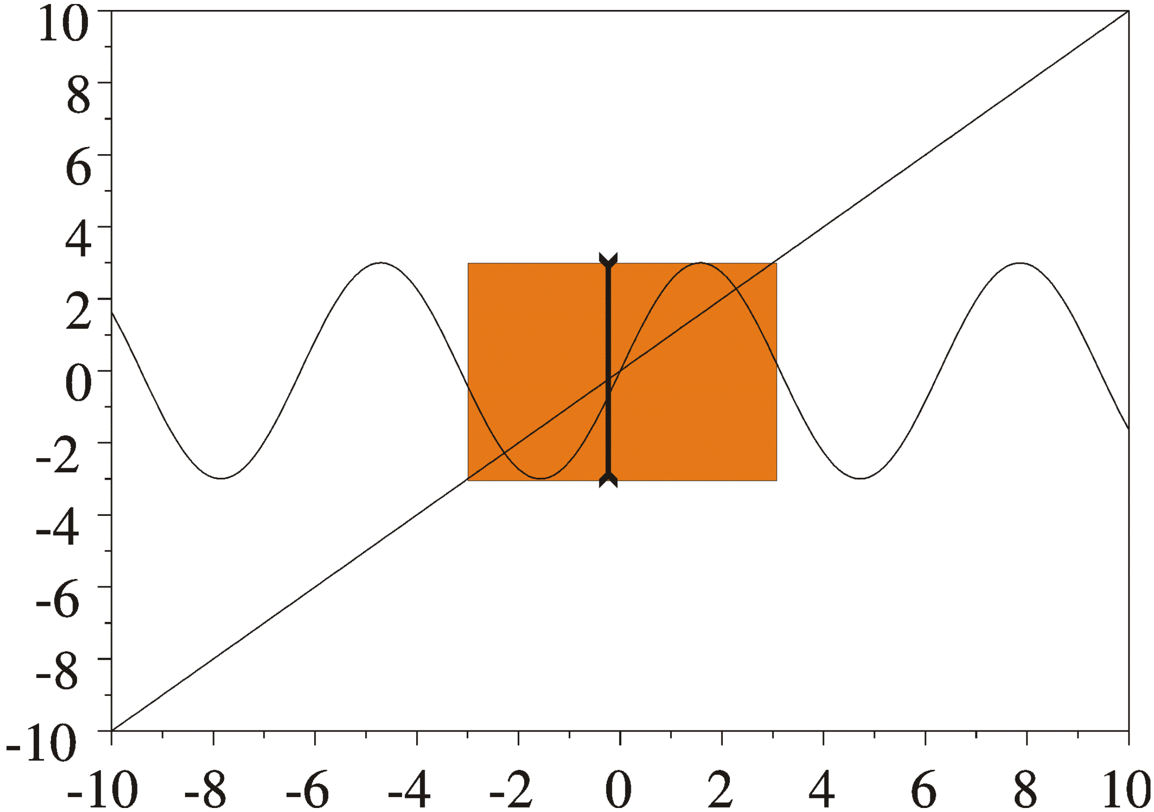
$$\left\{\begin{array}{rcl} y & = & 3\sin(x) \\ y & = & x \end{array}\right. \qquad x \in \mathbb{R},\ y \in \mathbb{R}.$$

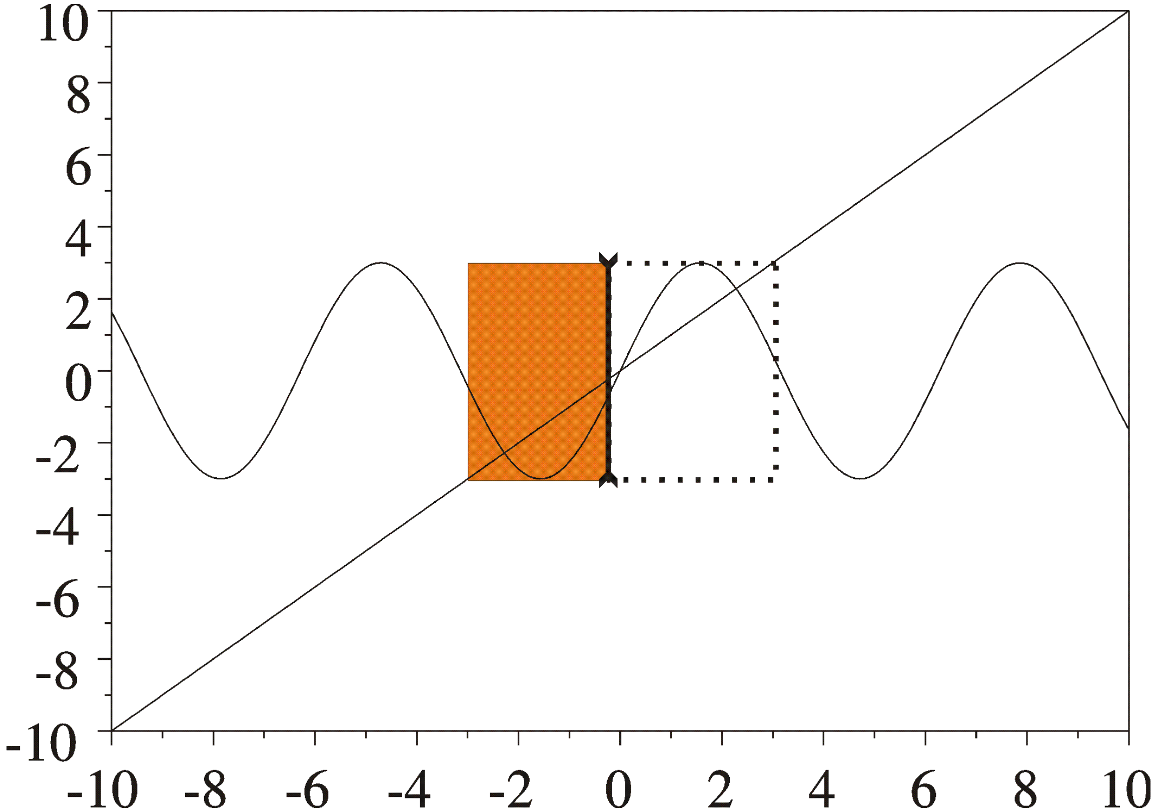

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10

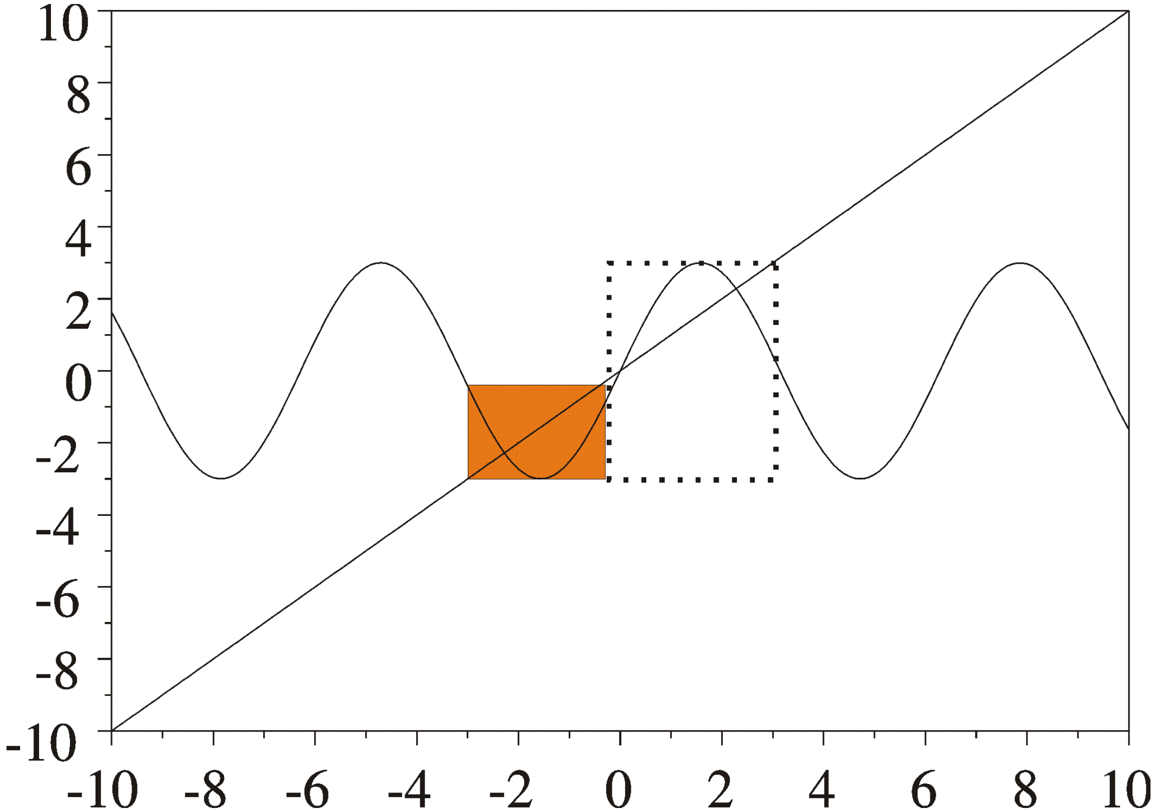


10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

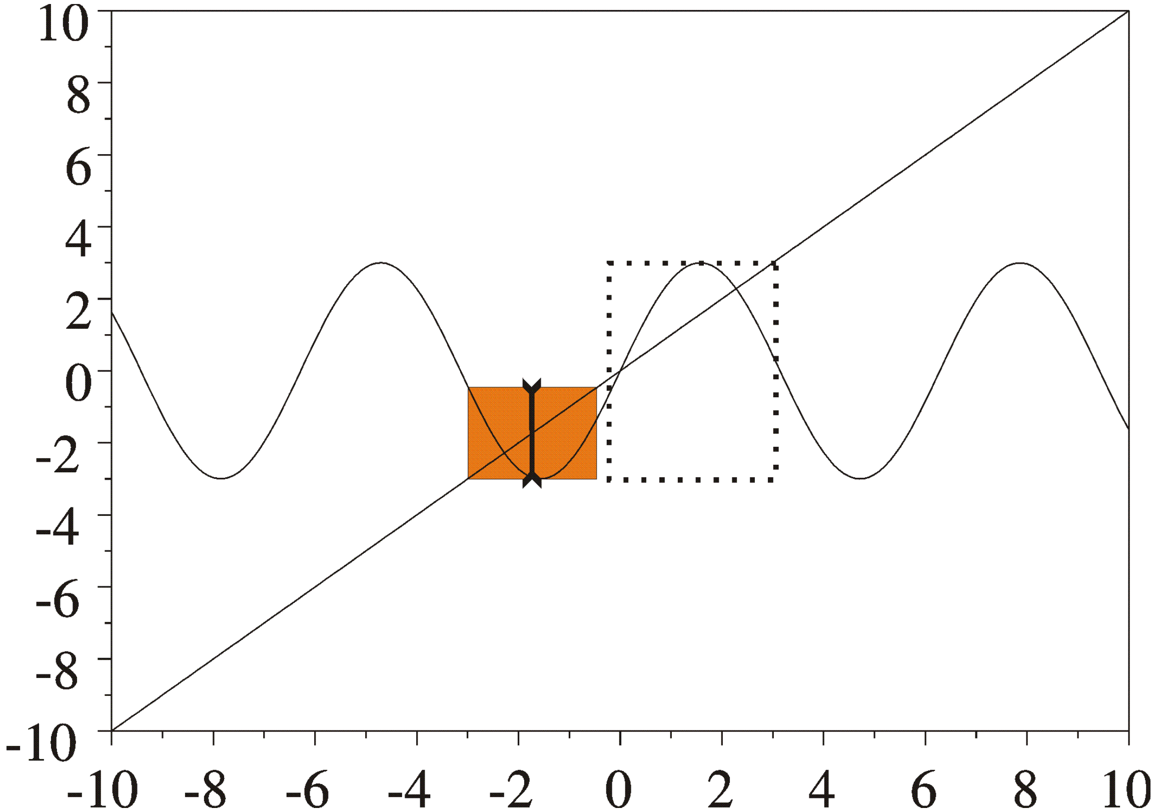

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10


10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

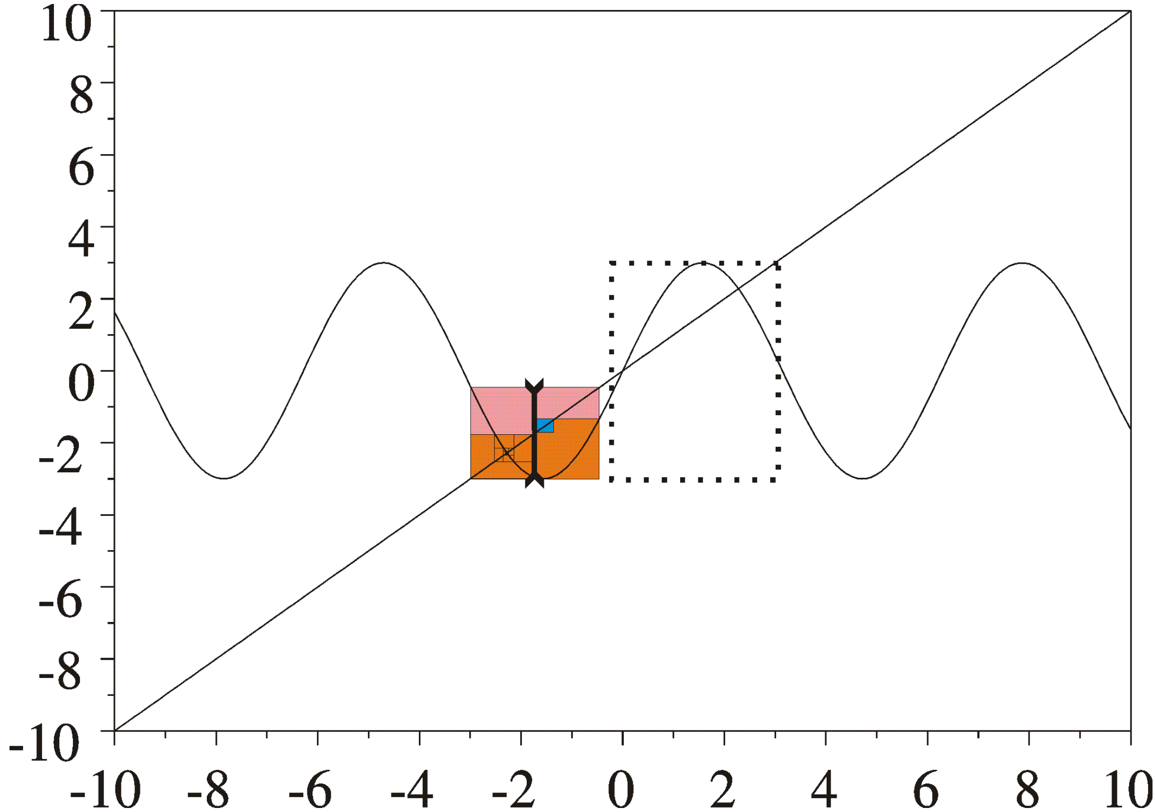

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

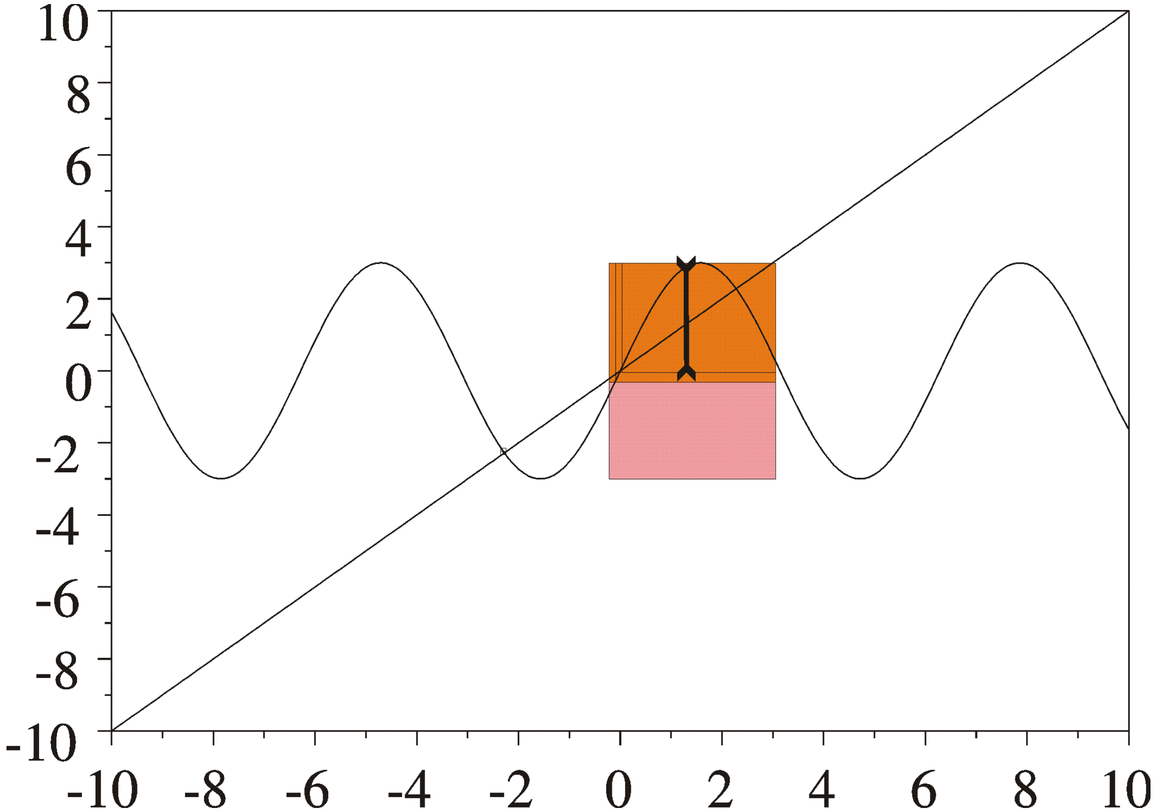


10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

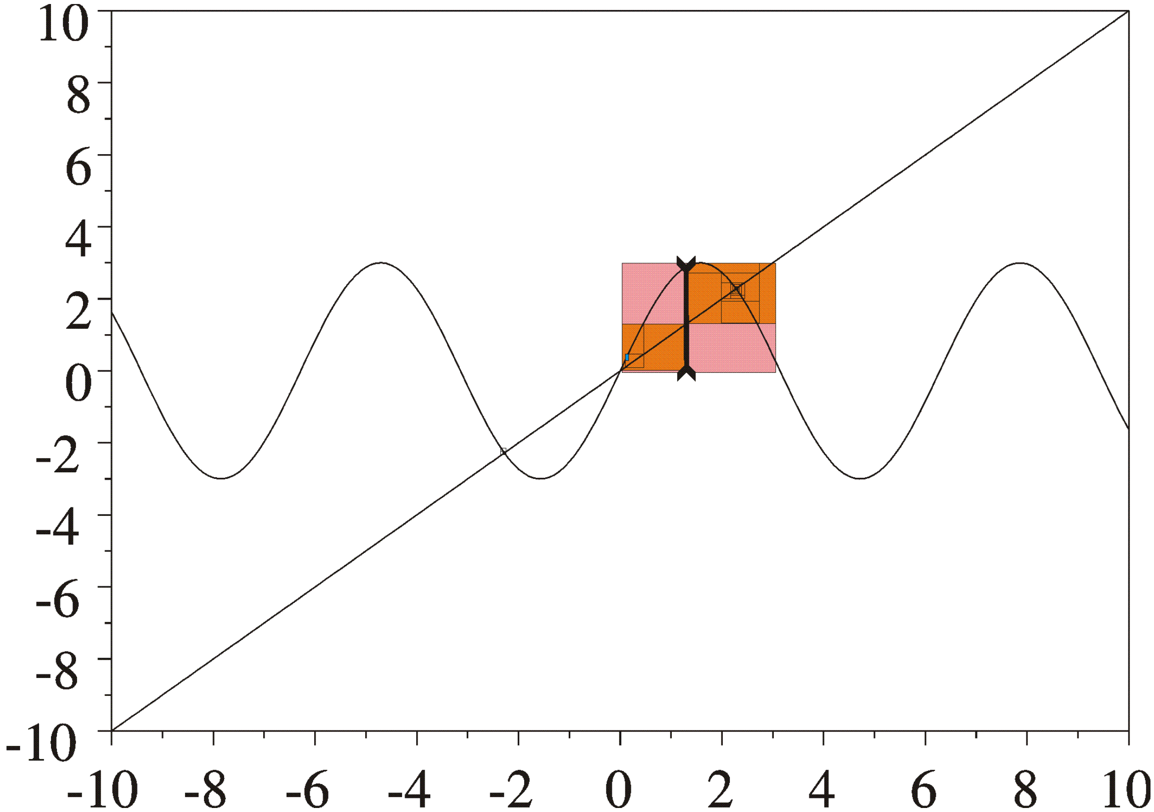


10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10


10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10



10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10



10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

3.6 Decomposition into primitive constraints

$$x + \mathsf{sin}(xy) \leq 0,$$
$$x \in [-1, 1], y \in [-1, 1], z \in [-1, 1]$$

can be decomposed into

$$\left\{ \begin{array}{l} a = xy \\ b = \mathsf{sin}(a) \\ c = x + b \end{array} \right. , \qquad \begin{array}{ll} x \in [-1, 1] & a \in [-\infty, \infty] \\ y \in [-1, 1] & b \in [-\infty, \infty] \\ z \in [-1, 1] & c \in [-\infty, 0] \end{array}$$

3.7 Set and contractors

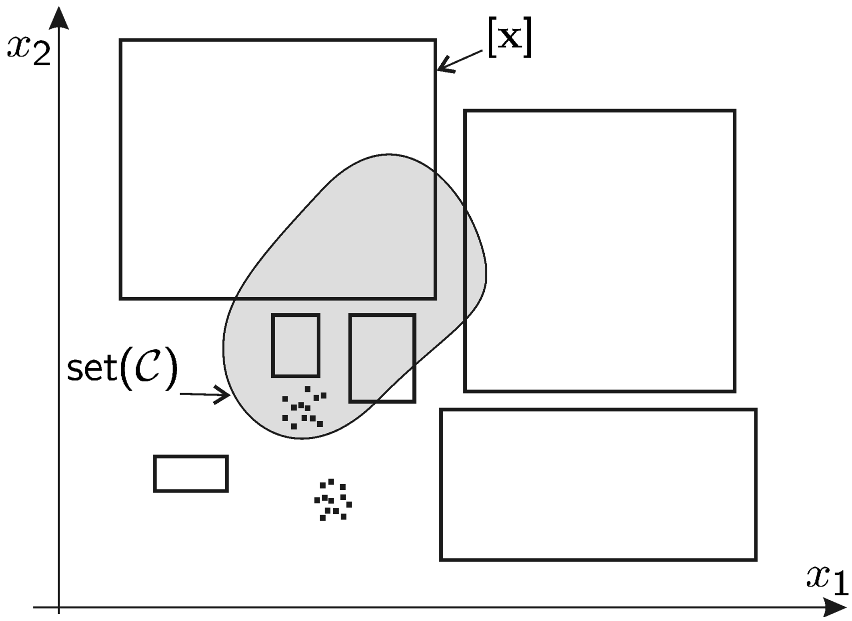
A contractor one way to represent a set of $\mathbb{R}^n$.

The set associated with a contractor $\mathcal{C}$ is

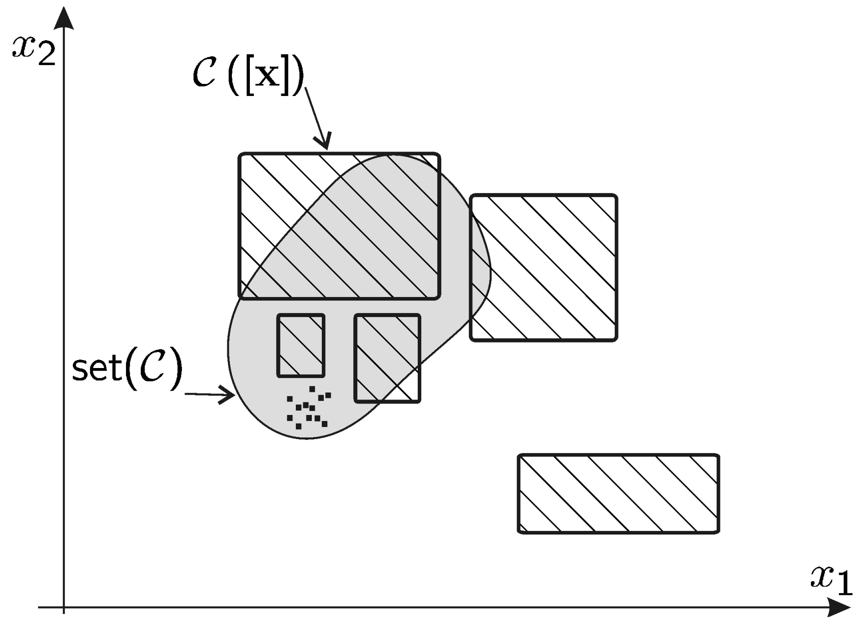
$$\mathrm{set}\,(\mathcal{C}) = \{\mathbf{x} \in \mathbb{R}^n, \mathcal{C}(\{\mathbf{x}\}) = \{\mathbf{x}\}\}\,.$$

Its domain is

$$\mathrm{dom}\,(\mathcal{C}) = \{\mathbf{x} \in \mathbb{R}^n, \mathcal{C}(\{\mathbf{x}\}) = \emptyset\}\,.$$



x_2
$[\mathbf{x}]$
set($\mathcal{C}$)
x_1


x_2
$\mathcal{C}([\mathbf{x}])$
set$(\mathcal{C})$
x_1

For instance, the set associated with the contractor

$$\mathcal{C}_1 \left(\begin{array}{c} [x_1] \\ [x_2] \\ [x_3] \end{array} \right) \stackrel{\text{def}}{=} \left(\begin{array}{c} [x_1] \cap ([x_3] - [x_2]) \\ [x_2] \cap ([x_3] - [x_1]) \\ [x_3] \cap ([x_1] + [x_2]) \end{array} \right)$$

is

$$\text{set}\left(\mathcal{C}_1\right) = \left\{ (x_1, x_2, x_3), x_3 = x_1 + x_2 \right\}.$$

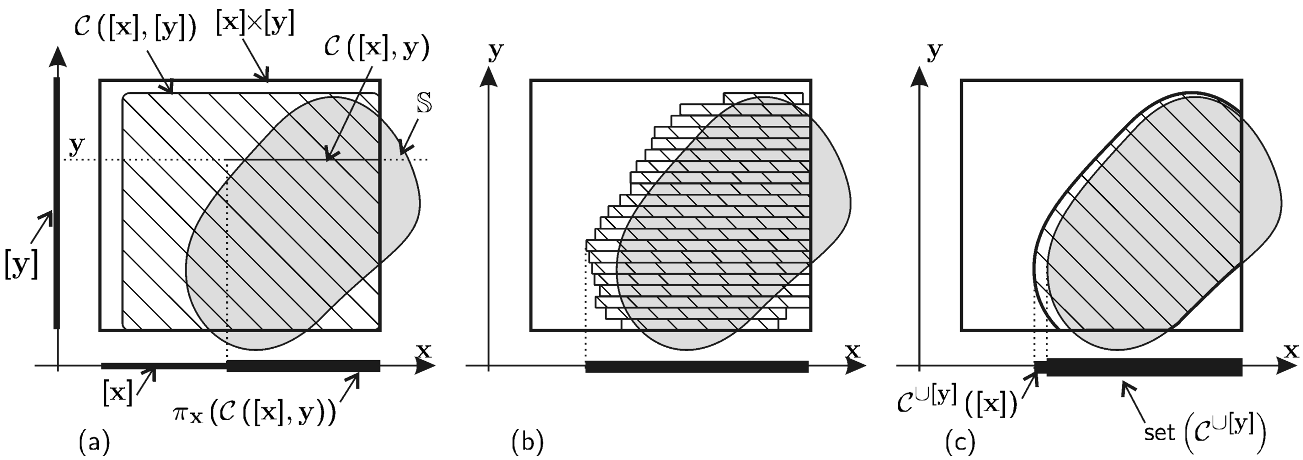
A contractor is also one way to represent one equation $x_3 = x_1 + x_2$.

3.8 Operations on contractors

intersection	$(\mathcal{C}_1 \cap \mathcal{C}_2)([\mathbf{x}]) \stackrel{\text{def}}{=} \mathcal{C}_1([\mathbf{x}]) \cap \mathcal{C}_2([\mathbf{x}])$
union	$(\mathcal{C}_1 \cup \mathcal{C}_2)([\mathbf{x}]) \stackrel{\text{def}}{=} [\mathcal{C}_1([\mathbf{x}]) \cup \mathcal{C}_2([\mathbf{x}])]$
composition	$(\mathcal{C}_1 \circ \mathcal{C}_2)([\mathbf{x}]) \stackrel{\text{def}}{=} \mathcal{C}_1(\mathcal{C}_2([\mathbf{x}]))$
repetition	$\mathcal{C}^\infty \stackrel{\text{def}}{=} \mathcal{C} \circ \mathcal{C} \circ \mathcal{C} \circ \dots$
repeat intersection	$\mathcal{C}_1 \sqcap \mathcal{C}_2 = (\mathcal{C}_1 \cap \mathcal{C}_2)^\infty$
repeat union	$\mathcal{C}_1 \sqcup \mathcal{C}_2 = (\mathcal{C}_1 \cup \mathcal{C}_2)^\infty$

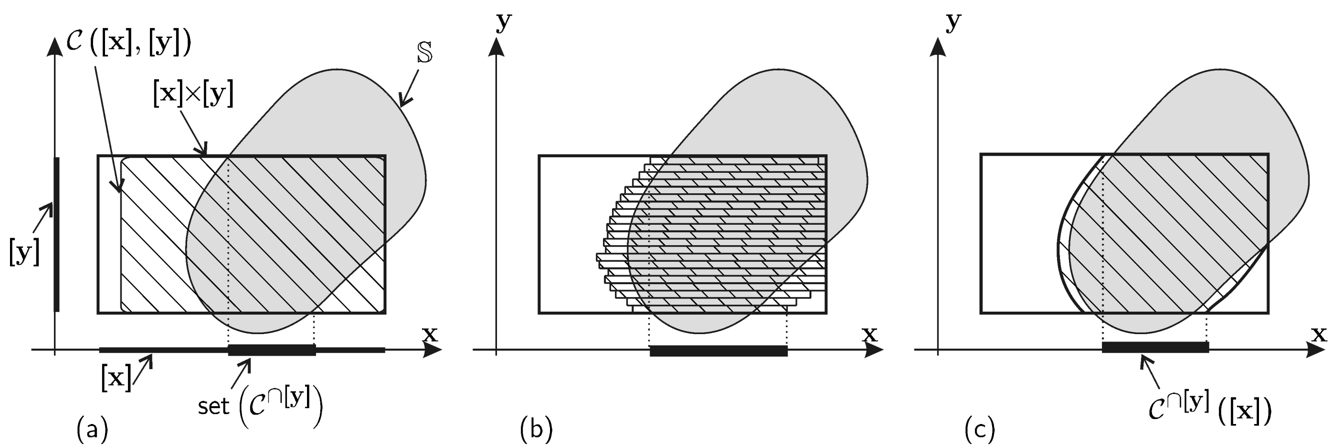
Consider the contractor $\mathcal{C}([\mathbf{x}],[\mathbf{y}])$, where $[\mathbf{x}] \in \mathbb{IR}^n, [\mathbf{y}] \in \mathbb{IR}^p$. We define the contractor

$$\mathcal{C}^{\cup[\mathbf{y}]}([\mathbf{x}]) = \left[\bigcup_{\mathbf{y}\in[\mathbf{y}]} \pi_{\mathbf{x}}\left(\mathcal{C}([\mathbf{x}],\mathbf{y})\right)\right] \qquad \text{(projected union)}$$



and also the contractor

$$\mathcal{C}^{\cap[\mathbf{y}]}([\mathbf{x}]) = \bigcap_{\mathbf{y}\in[\mathbf{y}]} \pi_{\mathbf{x}}(\mathcal{C}([\mathbf{x}],\mathbf{y})), \qquad \text{(projected intersection}$$



We have

$$\mathsf{set}\left(\mathcal{C}^{\cup[\mathbf{y}]}\right) = \{\mathbf{x}, \exists \mathbf{y} \in [\mathbf{y}], (\mathbf{x}, \mathbf{y}) \in \mathsf{set}\left(\mathcal{C}\right)\}$$
$$\mathsf{set}\left(\mathcal{C}^{\cap[\mathbf{y}]}\right) = \{\mathbf{x}, \forall \mathbf{y} \in [\mathbf{y}], (\mathbf{x}, \mathbf{y}) \in \mathsf{set}\left(\mathcal{C}\right)\}.$$

3.9 QUIMPER

The collection of contractors $\{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ is *complementary* if

$$\operatorname{set}(\mathcal{C}_1) \cap \cdots \cap \operatorname{set}(\mathcal{C}_m) = \emptyset.$$

Quimper is a high-level language for QUick Interval Modeling and Programming in a bounded-ERror context.

Quimper is an interpreted language for set computation.

A Quimper program is a set of complementary contractors.

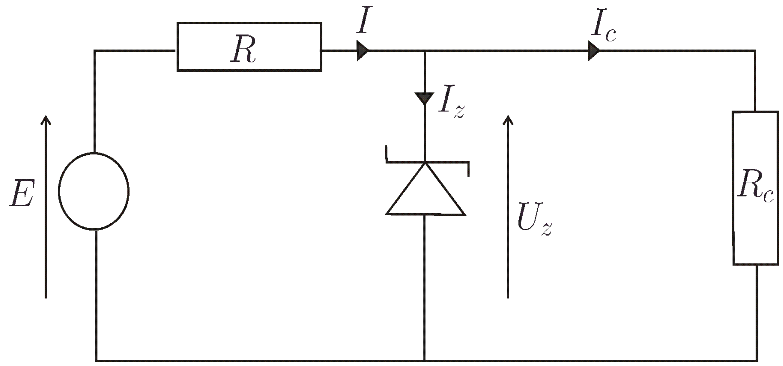
Quimper returns m subpavings, where m is the number of contractors

It is available at

`http://ibex-lib.org/`

4 Application of contractors

4.1 Bounded-error estimation



R
I
I_c
I_z
E
U_z
R_c

It is known that

$$U_z \in [6,7]V,\ r \in [7,8]\Omega,\ U_0 \in [6,6.2]V$$
$$R \in [100,110]\Omega,\ E \in [18,20]V,\ I_z \in [0,\infty]A$$
$$I \in]-\infty,\infty[A,\ I_c \in]-\infty,\infty[A, R_c \in [50,60]\Omega.$$

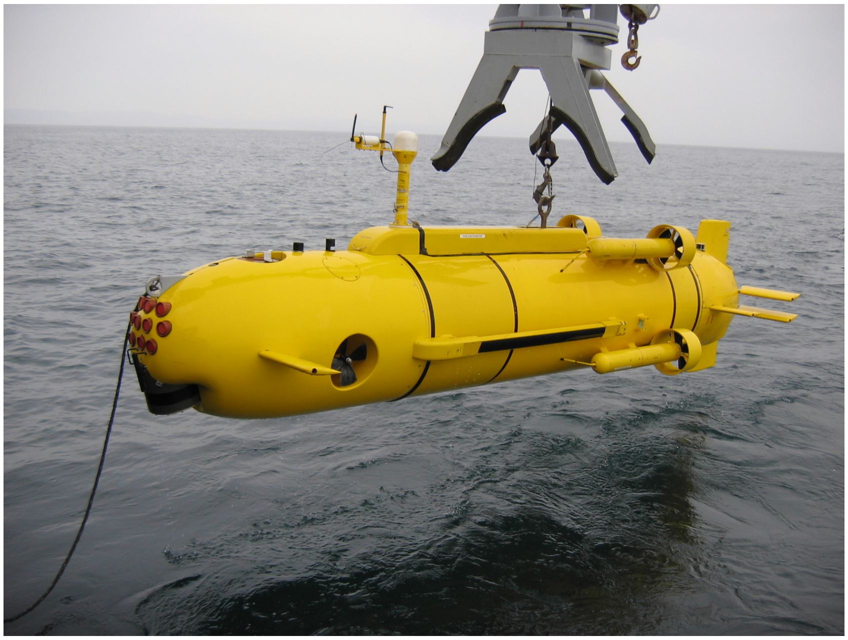
The constraints are

Zener diode	$I_z = \max(0, \frac{U_z - U_0}{r})$,
Ohm rule	$U_z = R_c I_c$,
Current rule	$I = I_c + I_z$,
Voltage rule	$E = RI + U_z$.

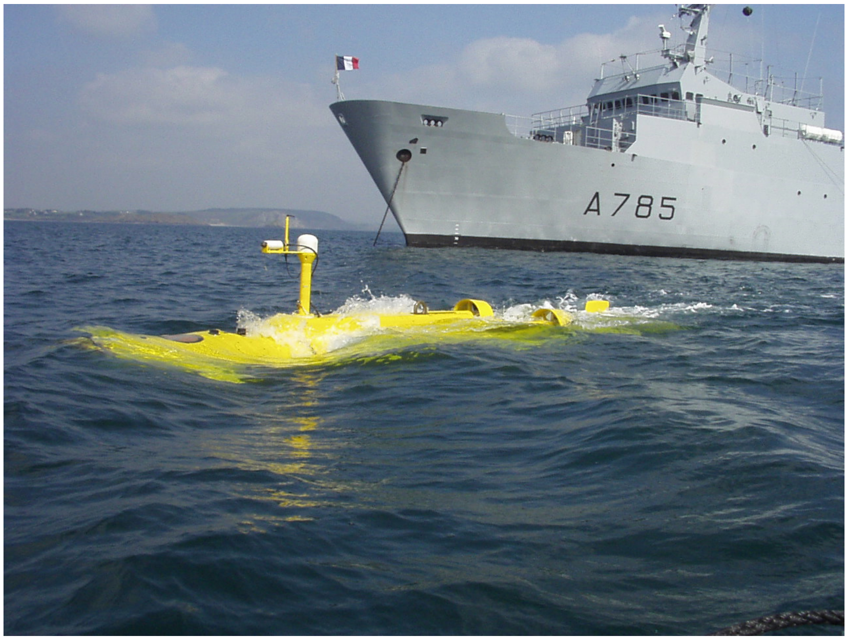
IntervalPeeler contracts the domains into:

$$
\begin{aligned}
&U_z \in [6,007;6,518], r \in [7,8]\Omega,\\
&U_0 \in [6,6.2]V, R \in [100,110]\Omega,\\
&E \in [18,20]V, I_z \in [0.,0.398]A\\
&I \in [0.11;0.14]A,\ \ I_c \in [0.1;0,13]A,\\
&R_c \in [50,60]\Omega
\end{aligned}
$$

4.2 SLAM



Redermor, GESMA
(Groupe d'Etude Sous-Marine de l'Atlantique)


A785

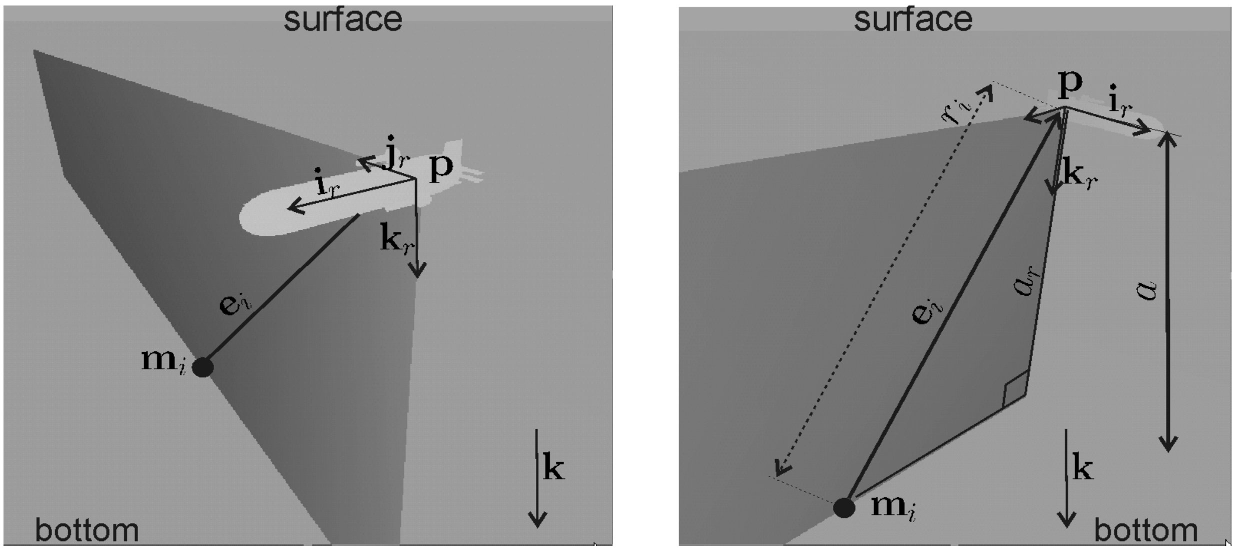
Montrer la simulation

4.2.1 Sensors

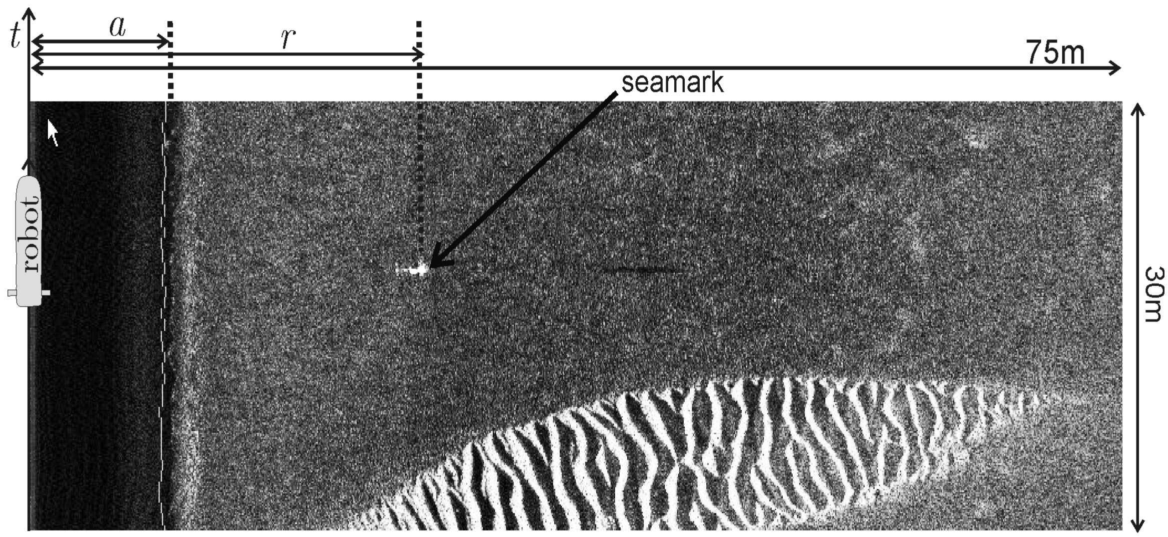
GPS (Global positioning system), only at the surface.

$$
\begin{array}{ll}
t_0 = 6000 \text{ s}, & \ell^0 = (-4.4582279^\circ, 48.2129206^\circ) \pm 2.5m \\
t_f = 12000 \text{ s}, & \ell^f = (-4.4546607^\circ, 48.2191297^\circ) \pm 2.5m
\end{array}
$$

Sonar (KLEIN 5400 side scan sonar).



Screenshot of SonarPro



Mine detecttion with SonarPro

Loch-Doppler returns the speed robot $\mathbf{v}_r$.

$$\mathbf{v}_r \in \tilde{\mathbf{v}}_r + 0.004 * [-1, 1] \, .\tilde{\mathbf{v}}_r + 0.004 * [-1, 1]$$

Inertial central (Octans III from IXSEA).

$$\left(\begin{array}{c}\phi\\ \theta\\ \psi\end{array}\right)\in\left(\begin{array}{c}\tilde{\phi}\\ \tilde{\theta}\\ \tilde{\psi}\end{array}\right)+\left(\begin{array}{c}1.75\times 10^{-4}.\,[-1,1]\\ 1.75\times 10^{-4}.\,[-1,1]\\ 5.27\times 10^{-3}.\,[-1,1]\end{array}\right).$$

Six mines have been detected.

i	0	1	2	3	4	5	6	7	8	9	10	11
$\tau(i)$	7054	7092	7374	7748	9038	9688	10024	10817	11172	11232	11279	11688
$\sigma(i)$	1	2	1	0	1	5	4	3	3	4	5	1
$\tilde{r}(i)$	52.42	12.47	54.40	52.68	27.73	26.98	37.90	36.71	37.37	31.03	33.51	15.05

4.2.2 Constraints

$$t \in \{6000.0, 6000.1, 6000.2, \ldots, 11999.4\},$$

$$i \in \{0, 1, \ldots, 11\},$$

$$\begin{pmatrix} p_x(t) \\ p_y(t) \end{pmatrix} = 111120 \begin{pmatrix} 0 & 1 \\ \cos\left(\ell_y(t) * \frac{\pi}{180}\right) & 0 \end{pmatrix} \begin{pmatrix} \ell_x(t) - \ell_x^0 \\ \ell_y(t) - \ell_y^0 \end{pmatrix}$$

$$\mathbf{p}(t) = (p_x(t), p_y(t), p_z(t)),$$

$$\mathbf{R}_\psi(t) = \begin{pmatrix} \cos\psi(t) & -\sin\psi(t) & 0 \\ \sin\psi(t) & \cos\psi(t) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{R}_\theta(t) = \begin{pmatrix} \cos\theta(t) & 0 & \sin\theta(t) \\ 0 & 1 & 0 \\ -\sin\theta(t) & 0 & \cos\theta(t) \end{pmatrix},$$

$$\mathbf{R}_{\varphi}(t) = \left(\begin{array}{ccc} 1 & 0 & 0 \\ 0 & \cos\varphi(t) & -\sin\varphi(t) \\ 0 & \sin\varphi(t) & \cos\varphi(t) \end{array} \right),$$

$$\mathbf{R}(t) = \mathbf{R}_{\psi}(t)\mathbf{R}_{\theta}(t)\mathbf{R}_{\varphi}(t),$$

$$\dot{\mathbf{p}}(t) = \mathbf{R}(t).\mathbf{v}_r(t),$$

$$||\mathbf{m}(\sigma(i)) - \mathbf{p}(\tau(i))|| \; = r(i),$$

$$\mathbf{R}^{\mathsf{T}}(\tau(i))\,(\mathbf{m}(\sigma(i)) - \mathbf{p}(\tau(i))) \in [0] \times [0, \infty]^{\times 2},$$

$$m_z(\sigma(i)) - p_z(\tau(i)) - a(\tau(i)) \in [-0.5, 0.5]$$

```
Constants
N = 59996; // Number of time steps
Variables
  R[N-1][3][3], // rotation matrices
  p[N][3], // positions
  v[N-1][3], // speed vectors
  phi[N-1],theta[N-1],psi[N-1]; // Euler angles
  px[N],py[N]; // for display only
```

```
function R[3][3]=euler(phi,theta,psi)
  cphi = cos(phi);
  sphi = sin(phi);
  ctheta = cos(theta);
  stheta = sin(theta);
  cpsi = cos(psi);
  spsi = sin(psi);
  R[1][1]=ctheta*cpsi;
  R[1][2]=-cphi*spsi+stheta*cpsi*sphi;
  R[1][3]=spsi*sphi+stheta*cpsi*cphi;
  R[2][1]=ctheta*spsi;
  R[2][2]=cpsi*cphi+stheta*spsi*sphi;
  R[2][3]=-cpsi*sphi+stheta*cphi*spsi;
  R[3][1]=-stheta;
  R[3][2]=ctheta*sphi;
  R[3][3]=ctheta*cphi;
end
```

```
contractor-list rotation
  for k=1:N-1;
    R[k]=euler(phi[k],theta[k],psi[k]);
  end
end
//--------------------------------
contractor-list statequ
  for k=1:N-1;
    p[k+1]=p[k]+0.1*R[k]*v[k];
  end
end
//--------------------------------
contractor init
  inter k=1:N-1;
    rotation(k)
  end
end
```

```
contractor fwd
  inter k=1:N-1;
    statequ(k)
  end
end
//-------------------------------
contractor bwd
  inter k=1:N-1;
    statequ(N-k)
  end
end
```

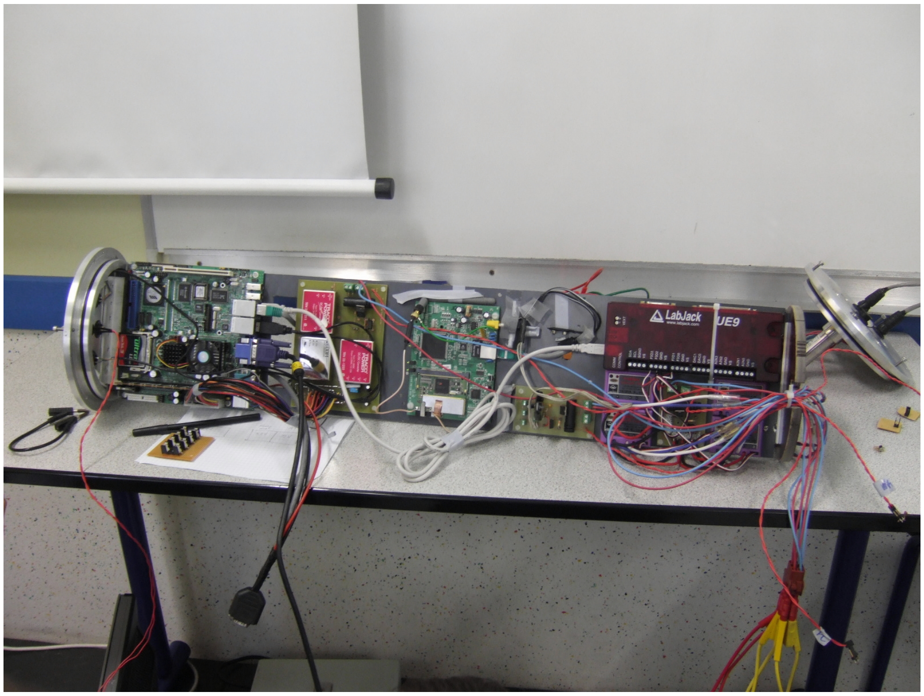
main

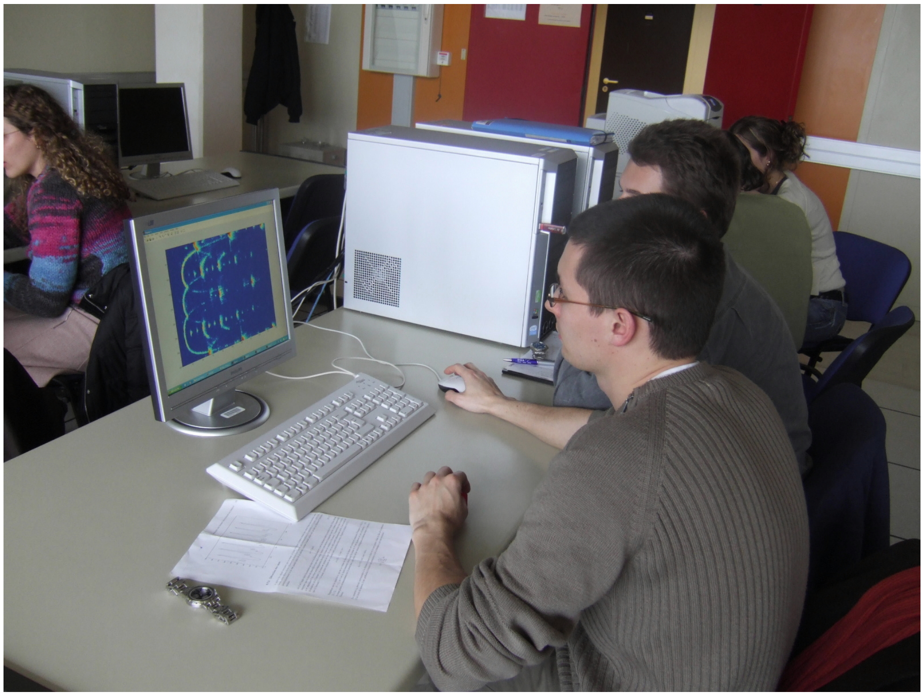
```
  p[1] :=read("gps_init.dat");
  v :=read("Quimper_v.dat");
  phi :=read("Quimper_phi.dat");
  theta :=read("Quimper_theta.dat");
  psi :=read("Quimper_psi.dat");
  init;
  fwd;
  bwd;
  column(p,px,1);
  column(p,py,2);
  print("--- Robot positions: ---");
  newplot("gesmi.dat");
  plot(px,py,color(rgb(1,1,1),rgb(0,0,0)));
end
```

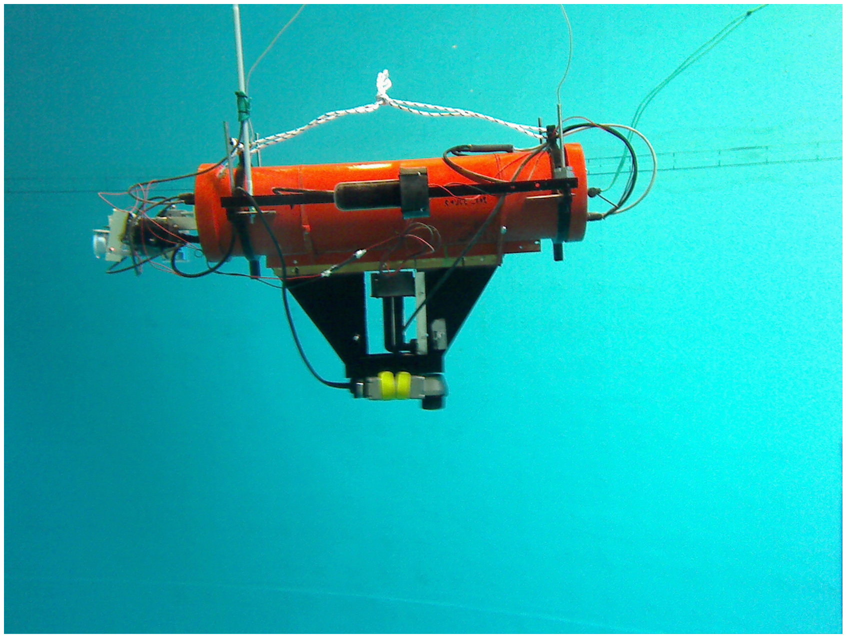
4.2.3 GESMI

4.3 Robust state estimation

Portsmouth, July 12-15, 2007.


LabJack
UE9





Sauc'isse robot

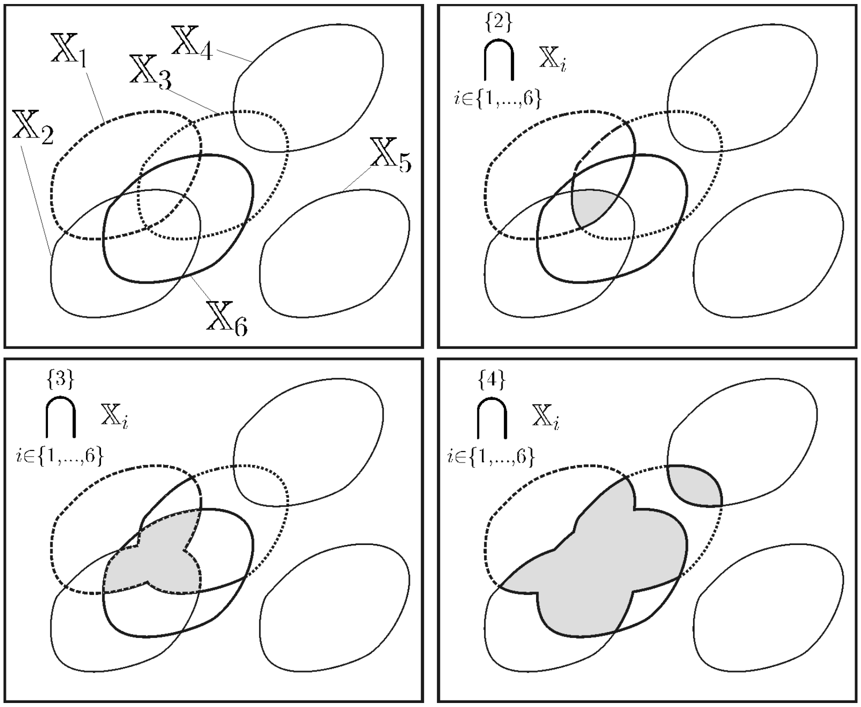


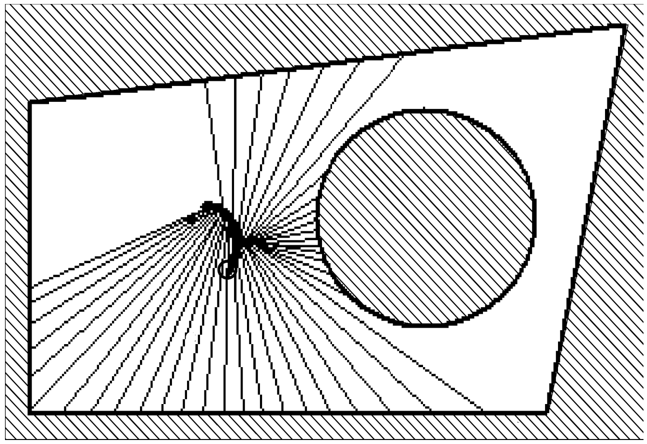
Illustration (in gray) of the q-relaxed intersection the 6 sets $\mathbb{X}_1, \ldots, \mathbb{X}_6$ where $q \in \{2, 3, 4\}$

Assumption. Within any time window of length ℓ there are less than q outliers.

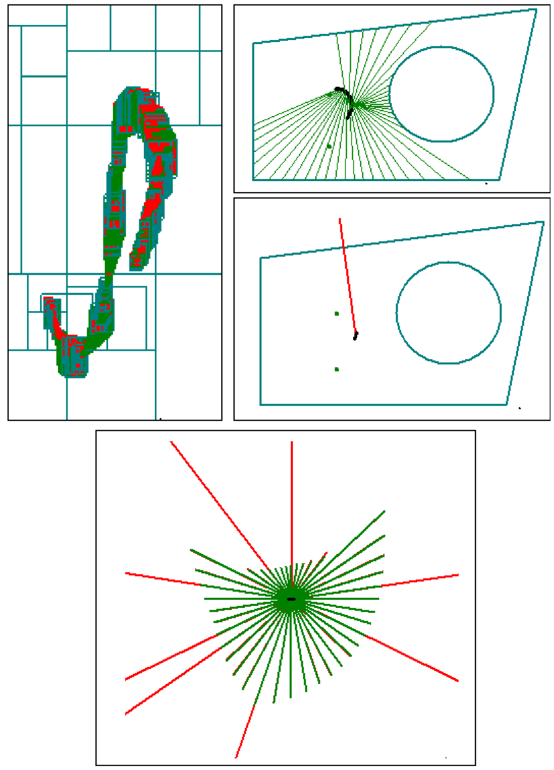
The set of feasible state can be computed recursively by

$$\mathbb{X}_{k+1} = \mathbf{f}_k(\mathbb{X}_k) \cap \bigcap_{i\in\{0,\ldots,\ell\}}^{\{q\}} \mathbf{f}_k \circ \mathbf{f}_{k-1} \circ \ldots \circ \mathbf{f}_{k-i} \circ \mathbf{g}_{k-i}^{-1}(\mathbb{Y}_{k-i}) .$$

Principle of the control of the underwater robot



Superposition of the poses
of the robot



4.4 Robust stability

A CSP is *infallible* if any arbitrary instantiation of the variables is a solution.

Consider the CSP

$$\begin{array}{rcl} \mathcal{V} & = & \{x, y\} \\ \mathcal{D} & = & \{[x],[y]\} \\ \mathcal{C} & = & \{\ f(x,y) \leq 0,\ g(x,y) \leq 0\}\,. \end{array}$$

The CSP is infallible if

$$\begin{array}{rl} & \forall x \in [x], \forall y \in [y],\ f(x,y) \leq 0 \text{ and } g(x,y) \leq 0, \\ \Leftrightarrow & \{(x,y) \in [x] \times [y] \mid f(x,y) > 0 \text{ or } g(x,y) > 0\} = \emptyset \\ \Leftrightarrow & \{(x,y) \in [x] \times [y] \mid \ \max\left(f(x,y), g(x,y)\right) > 0\} = \emptyset. \end{array}$$

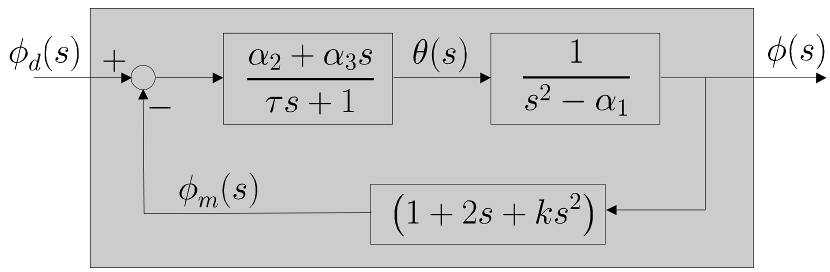
Consider a motorbike with a speed of 1m/s.

Angle of the handlebars: θ.

Rolling angle: ϕ

Wanted rolling angle: ϕ_d

Measured rolling angle: ϕ_m.



The input-output relation of the closed-loop system is :

$$\phi(s) = \frac{\alpha_2 + \alpha_3 s}{\left(s^2 - \alpha_1\right)(\tau s + 1) + (\alpha_2 + \alpha_3 s)\left(1 + 2s + ks^2\right)} \phi_d(s)$$

Its characteristic polynomial is thus

$$\begin{aligned} P(s) &= \left(s^2 - \alpha_1\right)(\tau s + 1) + (\alpha_2 + \alpha_3 s)\left(1 + 2s + ks^2\right) \\ &= a_3 s^3 + a_2 s^2 + a_1 s + a_0, \end{aligned}$$

with

$$\begin{array}{cc} a_3 = \tau + \alpha_3 k & a_2 = \alpha_2 k + 2\alpha_3 + 1 \\ a_1 = \alpha_3 - \alpha_1 \tau + 2\alpha_2 & a_0 = -\alpha_1 + \alpha_2. \end{array}$$

The Routh table is :

a_3	a_1
a_2	a_0
$\frac{a_2 a_1 - a_3 a_0}{a_2}$	0
a_0	0

The closed-loop system is stable if $a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}$ and a_0 have the same sign.

Assume that it is known that

$$\begin{array}{ll} \alpha_1 \in [8.8; 9.2] & \alpha_2 \in [2.8; 3.2] \\ \alpha_3 \in [0.8; 1.2] & \tau \in [1.8; 2.2] \\ k \in [-3.2; -2.8]. & \end{array}$$

The system is robustly stable if,

$\forall \alpha_1 \in [\alpha_1], \forall \alpha_2 \in [\alpha_2], \forall \alpha_3 \in [\alpha_3], \forall \tau \in [\tau], \forall k \in [k]$,
$a_3,\ a_2,\ \frac{a_2 a_1 - a_3 a_0}{a_2}$ and a_0 have the same sign.

Now, we have the equivalence

$$\begin{aligned} & b_1,\ b_2,\ b_3 \text{ and } b_4 \text{ have the same sign} \\ \Leftrightarrow\ \ & \max\left(\min\left(b_1, b_2, b_3, b_4\right), -\max\left(b_1, b_2, b_3, b_4\right)\right) > 0 \end{aligned}$$

The robust stability condition amounts to proving that

$$\begin{aligned} \exists \alpha_1 \ \ \in \ \ & [\alpha_1], \exists \alpha_2 \in [\alpha_2], \exists \alpha_3 \in [\alpha_3], \exists \tau \in [\tau], \exists k \in [k], \\ & \max(\quad \min\left(a_3, a_2, \tfrac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right), \\ & \qquad\quad -\max(a_3, a_2, \tfrac{a_2 a_1 - a_3 a_0}{a_2}, a_0) \quad) \leq 0 \end{aligned}$$

is false,. . .

i.e., that the CSP

$$
\begin{aligned}
\mathcal{V} &= \{a_0, a_1, a_2, a_3, \alpha_1, \alpha_2, \alpha_3, \tau, k\}, \\
\mathcal{D} &= \{[\alpha_0], [\alpha_1], [\alpha_2], [\alpha_3], [\alpha_2], [\alpha_3], [\tau], [k]\}, \\
\mathcal{C} &= \left\{ \begin{array}{l}
a_3 = \tau + \alpha_3 k \ ; \ a_2 = \alpha_2 k + 2\alpha_3 + 1 \ ; \\
a_1 = \alpha_3 - \alpha_1 \tau + 2\alpha_2, \\
a_0 = -\alpha_1 + \alpha_2 \ ; \\
m_1 = \min\left(a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right); \\
m_2 = \max\left(a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right) \\
\max(m_1, -m_2) \leq 0.
\end{array} \right\}
\end{aligned}
$$

has no solution.

This is easily proven by IntervalPeeler

Or with QUIMPER.

```
variables
 alpha1 in [8.8,9.2];
 alpha2 in [2.8,3.2];
 alpha3 in [0.8,1.2];
 tau in [1.8,2.2];
 k in [-3.2,-2.8];
 r in [-1e08,0];
 b1 in [-1e08,0];
 b2 in [0,-1e08];
 a3,a2,a1,a0,b;
```

```
contractor_list L
 a3=tau+alpha3*k;
 a2=alpha2*k+2*alpha3+1;
 a1=alpha3-alpha1*tau+2*alpha2;
 a0=alpha2-alpha1;
 b1=min(a3,a2,(a2*a1-a3*a0)/a2,a0);
 b2=max(a3,a2,(a2*a1-a3*a0)/a2,a0);
end
contractor C
   compose(L)
end
```