Interval robotics

Chapter 3: Contractors

Luc Jaulin,
ENSTA-Bretagne, Brest, France

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),

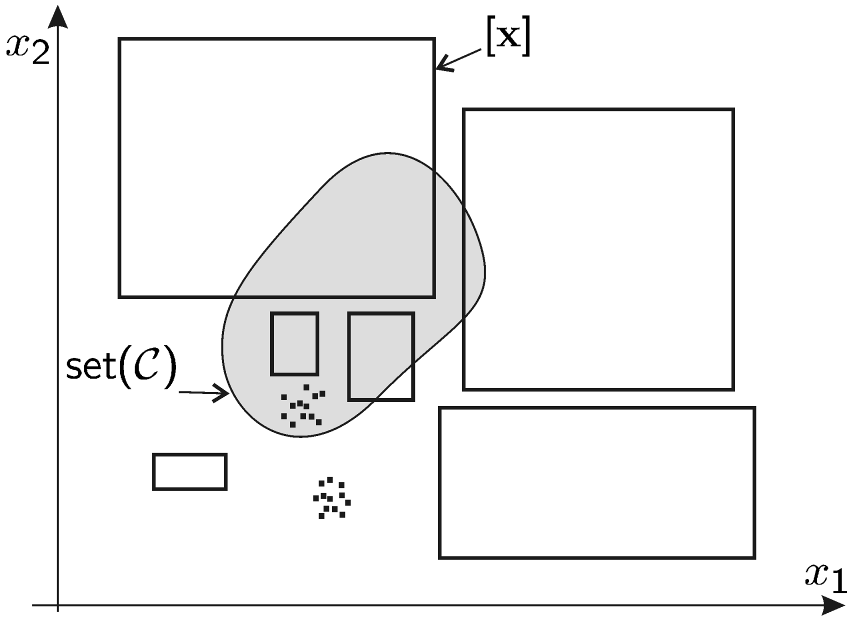
- contraction procedures are used as much as possible,

- bisections are used only as a last resort.

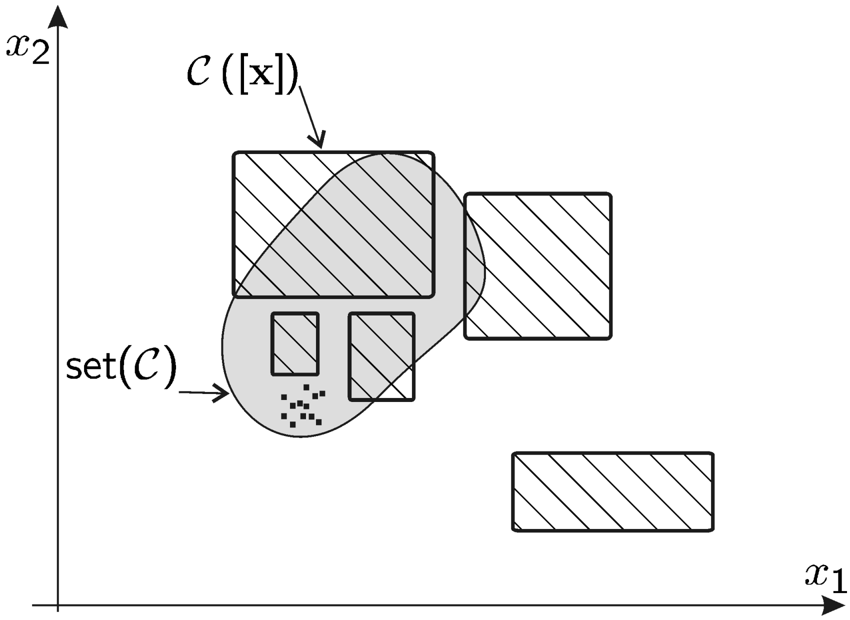
1 Definition

The operator $\mathcal{C}_{\mathbb{X}} : \mathbb{IR}^n \to \mathbb{IR}^n$ 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
$\mathcal{C}([\mathbf{x}])$
set$(\mathcal{C})$
x_1

The operator $\mathcal{C} : \mathbb{IR}^n \longrightarrow \mathbb{IR}^n$ is a *contractor* for the equation $f(\mathbf{x}) = 0$, if

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \begin{cases} \mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}] \\ \mathbf{x} \in [\mathbf{x}] \text{ et } f(\mathbf{x}) = 0 \Rightarrow \mathbf{x} \in \mathcal{C}([\mathbf{x}]) \end{cases}$$

$\mathcal{C}_{\mathbb{X}}$ 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	$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathcal{C}_{\mathbb{X}}\left(\mathcal{C}_{\mathbb{X}}([\mathbf{x}])\right) = \mathcal{C}_{\mathbb{X}}([\mathbf{x}])$.

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	$(\mathcal{C}_1 \circ \mathcal{C}_2)([\mathbf{x}]) \stackrel{\text{def}}{=} \mathcal{C}_1(\mathcal{C}_2([\mathbf{x}]))$
répétition	$\mathcal{C}^\infty \stackrel{\text{def}}{=} \mathcal{C} \circ \mathcal{C} \circ \mathcal{C} \circ \dots$

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

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

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

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

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: $[z], [x], [y]$)
1 $[z] := [z] \cap ([x] + [y])$;
2 $[x] := [x] \cap ([z] - [y])$;
3 $[y] := [y] \cap ([z] - [x])$.

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 $[z] := [z] \cap ([x] * [y])$;
2 $[x] := [x] \cap ([z] * 1/[y])$;
3 $[y] := [y] \cap ([z] * 1/[x])$.

Consider the binary constraint

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

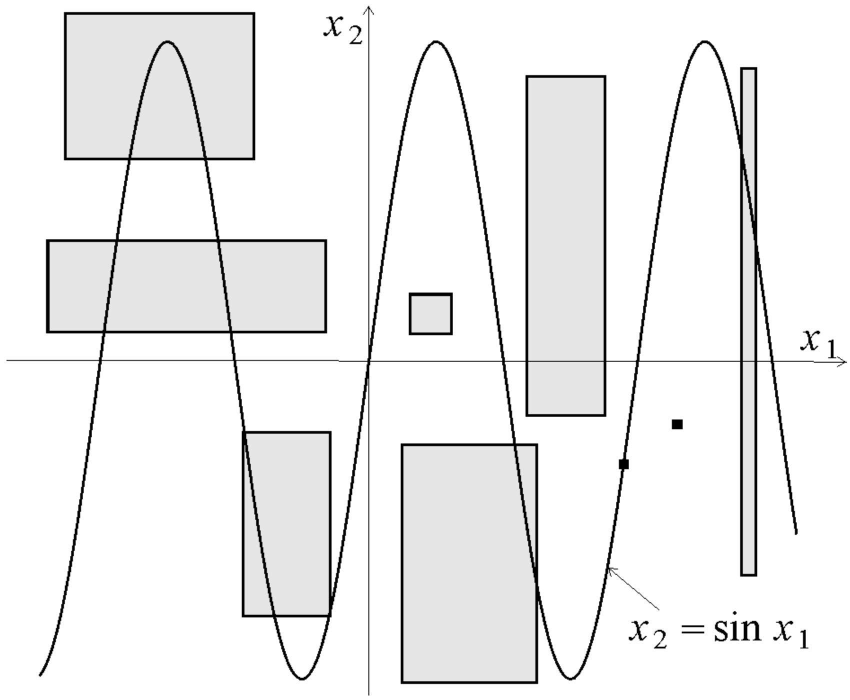
The associated contractor is

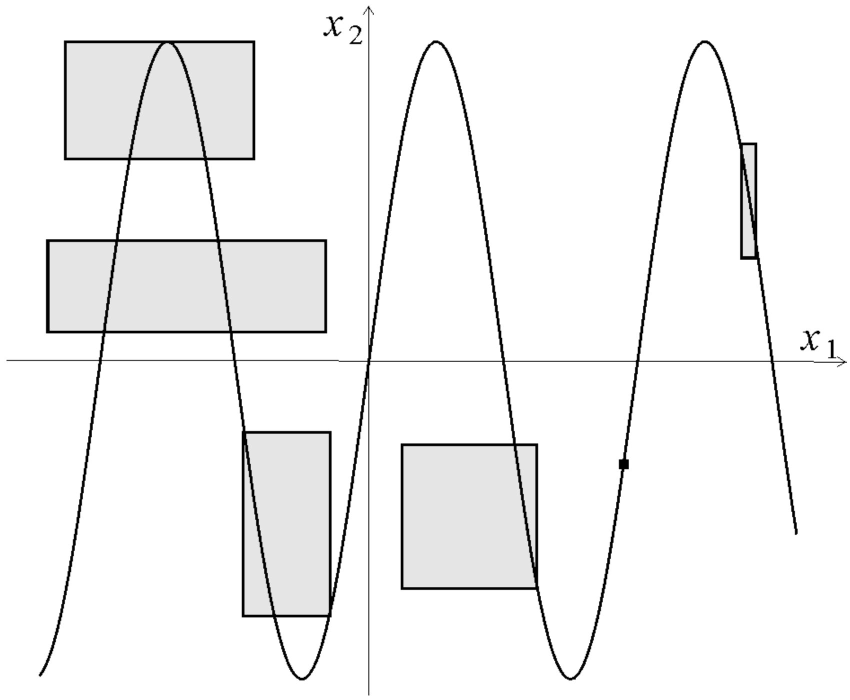
Algorithm pexp(inout: $[y],[x]$)	
1	$[y] := [y] \cap \mathsf{exp}\,([x])\,;$
2	$[x] := [x] \cap \mathsf{log}\,([y])\,.$

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

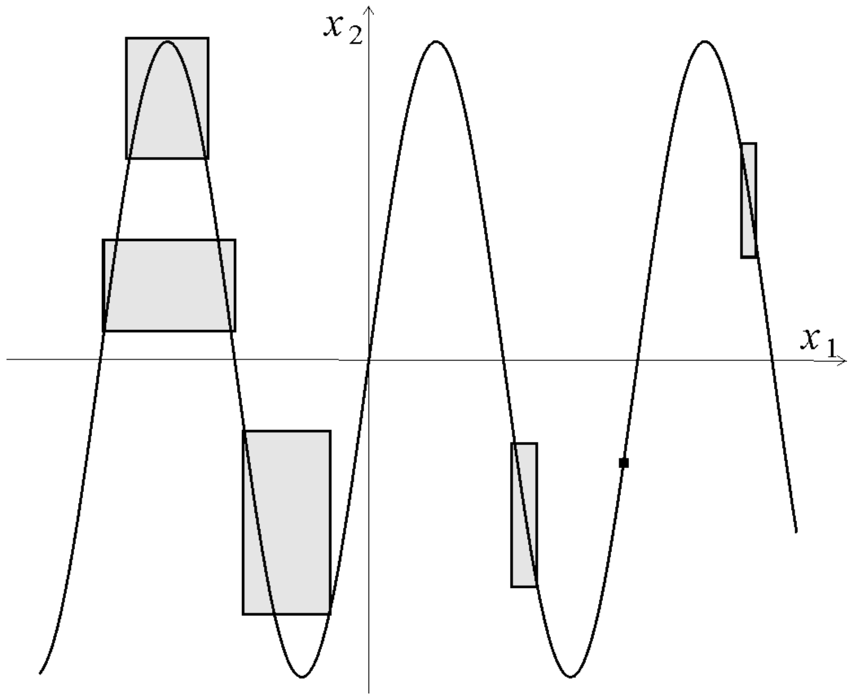
Example. Consider the primitive equation:

$$x_2 = \sin x_1.$$


x_2
x_1
$x_2 = \sin x_1$



Forward contraction



Backward contraction

Forward-backward contractor (HC4 revise)

For the equation

$$(x_1 + x_2) \cdot x_3 \in [1, 2]\,,$$

we have the following contractor:

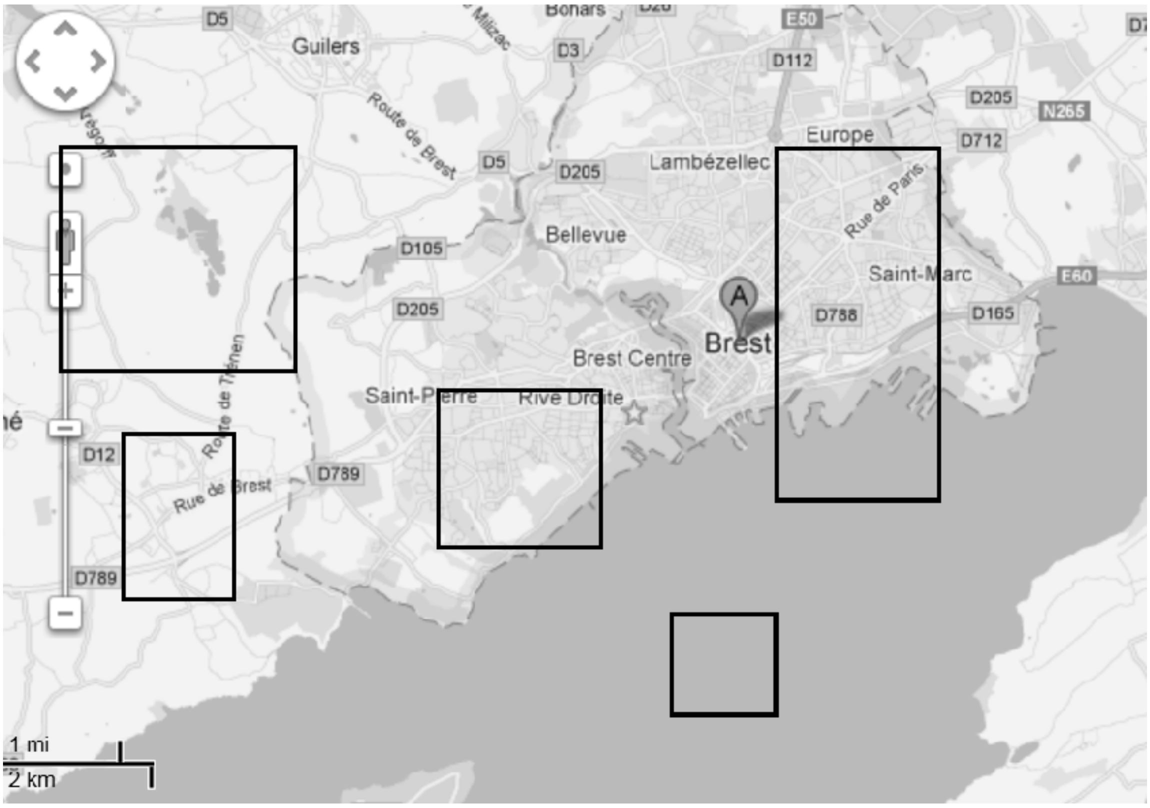
algorithm $\mathcal{C}$ (inout $[x_1]\,, [x_2]\,, [x_3]$)	
$[a] = [x_1] + [x_2]$	// $a = x_1 + x_2$
$[b] = [a] \cdot [x_3]$	// $b = a \cdot x_3$
$[b] = [b] \cap [1, 2]$	// $b \in [1, 2]$
$[x_3] = [x_3] \ \cap \ \frac{[b]}{[a]}$	// $x_3 = \frac{b}{a}$
$[a] = [a] \ \cap \ \frac{[b]}{[x_3]}$	// $a = \frac{b}{x_3}$
$[x_1] = [x_1] \ \cap \ \ [a] - [x_2]$	// $x_1 = a - x_2$
$[x_2] = [x_2] \ \cap \ \ [a] - [x_1]$	// $x_2 = a - x_1$

Properties

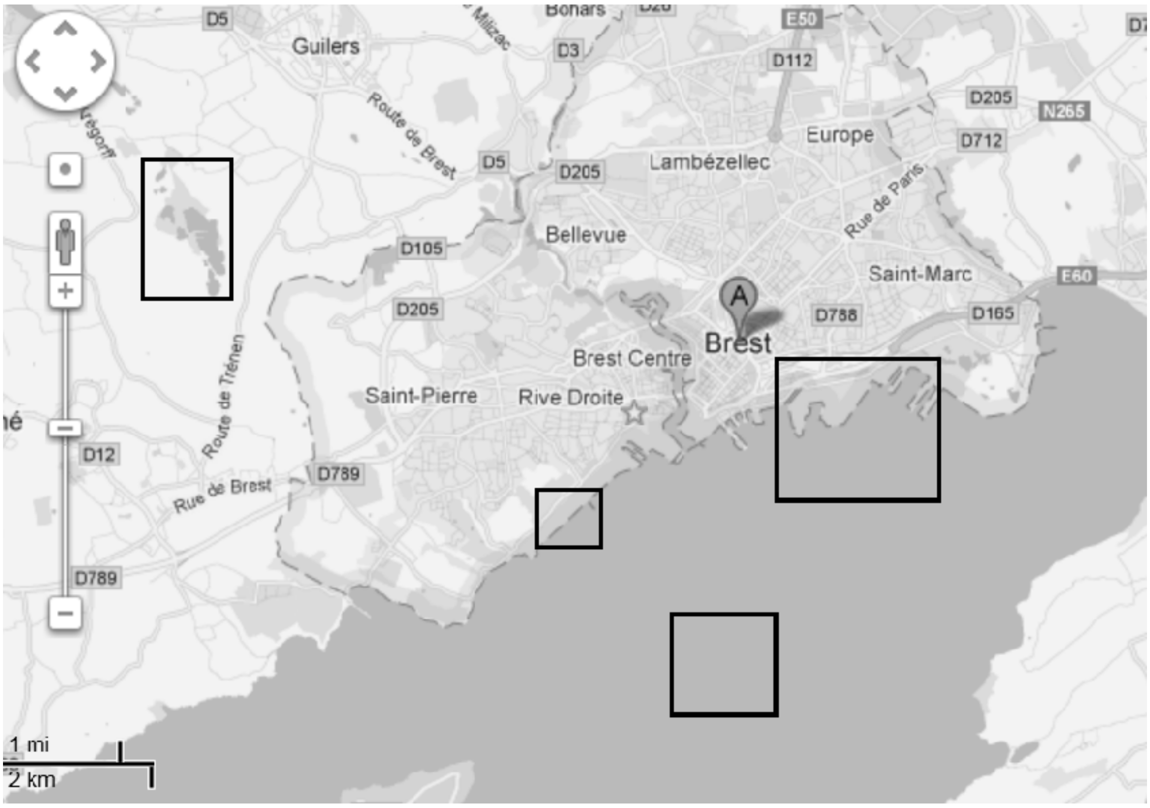
$$
\begin{aligned}
(\mathcal{C}_1^\infty \cap \mathcal{C}_2^\infty)^\infty &= (\mathcal{C}_1 \cap \mathcal{C}_2)^\infty \\
(\mathcal{C}_1 \cap (\mathcal{C}_2 \cup \mathcal{C}_3)) &\supset (\mathcal{C}_1 \cap \mathcal{C}_2) \cup (\mathcal{C}_1 \cap \mathcal{C}_3) \\
\left\{ \begin{array}{l} \mathcal{C}_1 \text{ minimal} \\ \mathcal{C}_2 \text{ minimal} \end{array} \right. &\Rightarrow \mathcal{C}_1 \cup \mathcal{C}_2 \text{ minimal}
\end{aligned}
$$

Contractor on images

The robot with coordinates (x_1, x_2) is in the water.



Guilers
Bohars
Route de Brest
Lambézellec
Europe
Bellevue
Saint-Marc
Rue de Paris
Brest
Brest Centre
Saint-Pierre
Rive Droite
Route de Trénen
Rue de Brest
D5
D3
D112
E50
D205
N265
D712
D105
E60
D788
D165
D12
D789
1 mi
2 km



Guilers
Route de Brest
D5
D205
Lambézellec
Europe
D112
E50
D205
N265
D712
Rue de Paris
Bellevue
D105
D205
Saint-Marc
E60
D788
D165
Brest
Brest Centre
Saint-Pierre
Rive Droite
Route de Trénen
D12
Rue de Brest
D789
D789
D3
D5
1 mi
2 km

4 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.

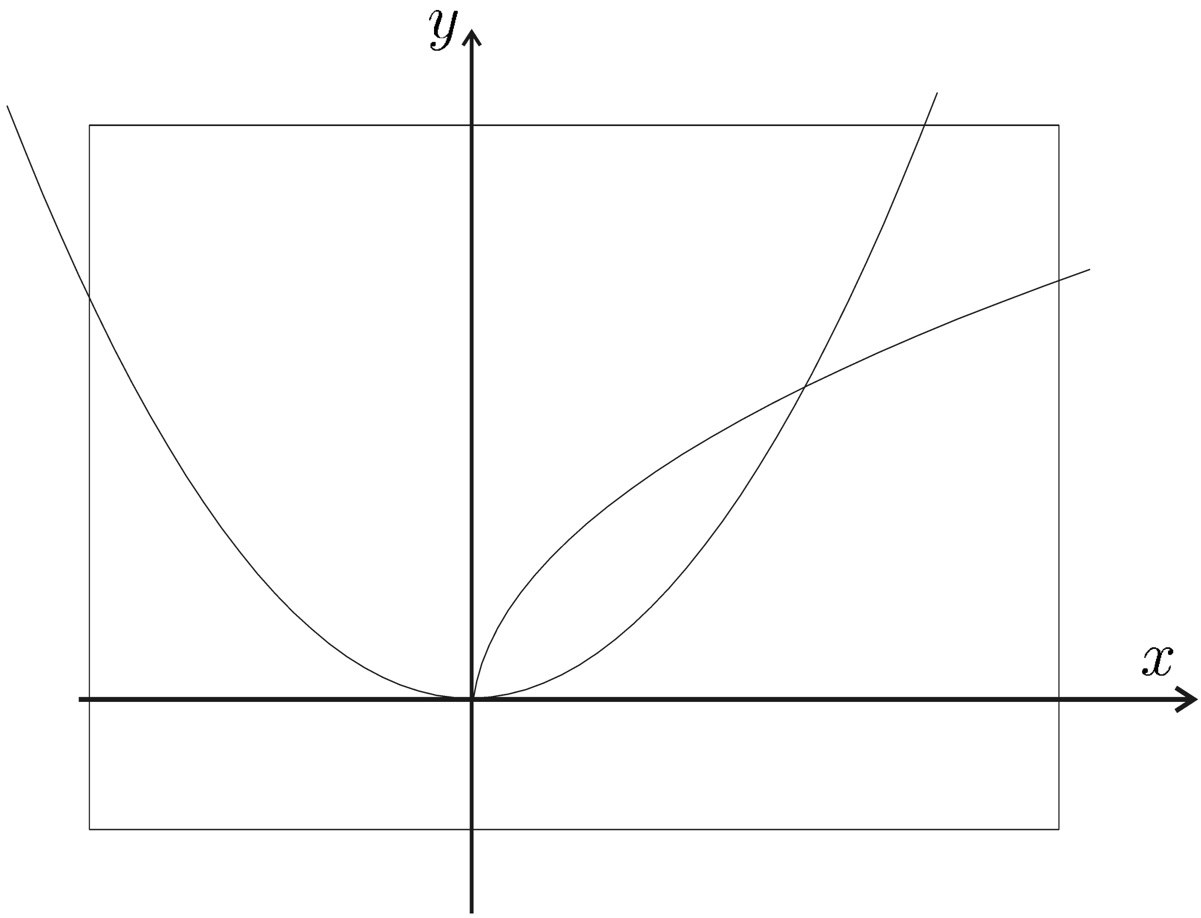
Example. Consider the system of two equations.

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

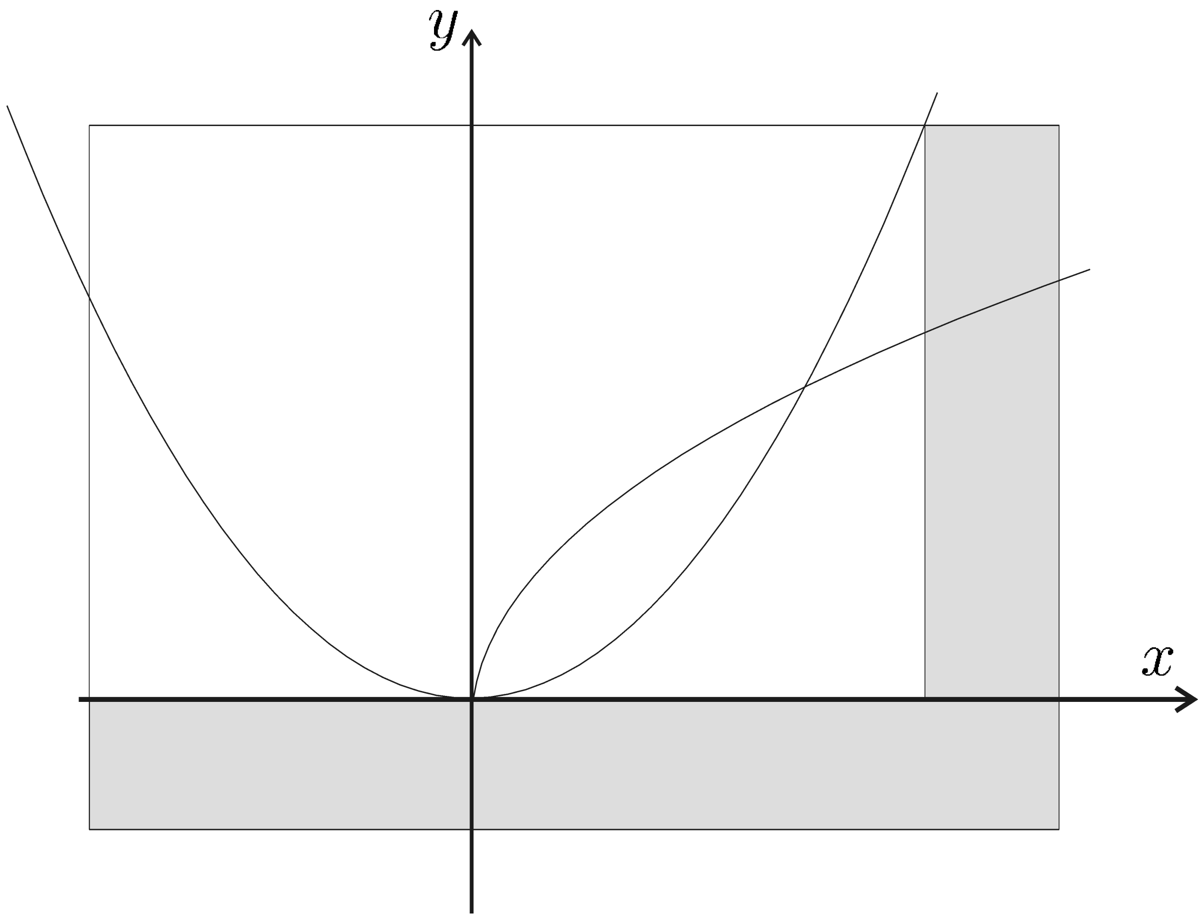
We can build two contractors

$$\mathcal{C}_1 : \left\{ \begin{array}{l} [y] = [y] \cap [x]^2 \\ [x] = [x] \cap \sqrt{[y]} \end{array} \right. \quad \text{associated to } y = x^2$$

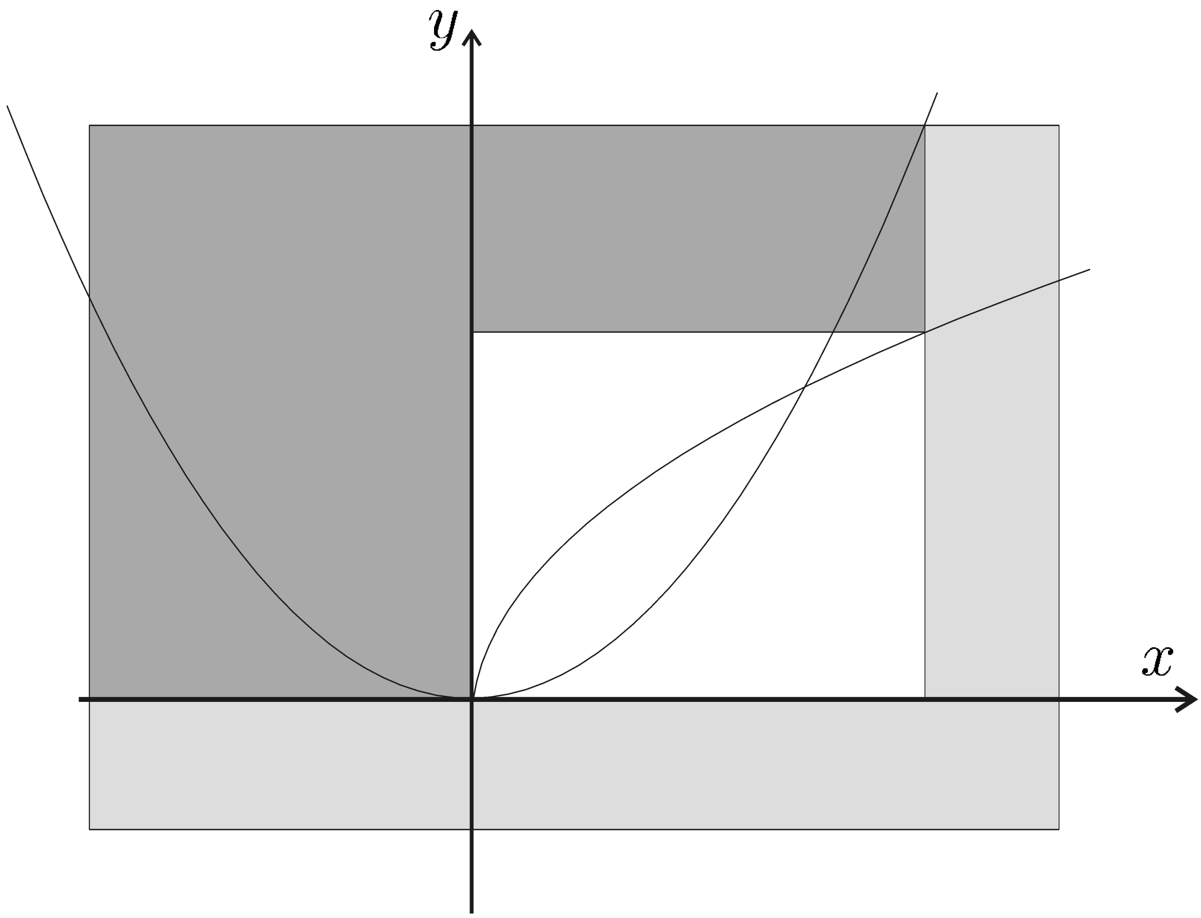
$$\mathcal{C}_2 : \left\{ \begin{array}{l} [y] = [y] \cap \sqrt{[x]} \\ [x] = [x] \cap [y]^2 \end{array} \right. \quad \text{associated to } y = \sqrt{x}$$



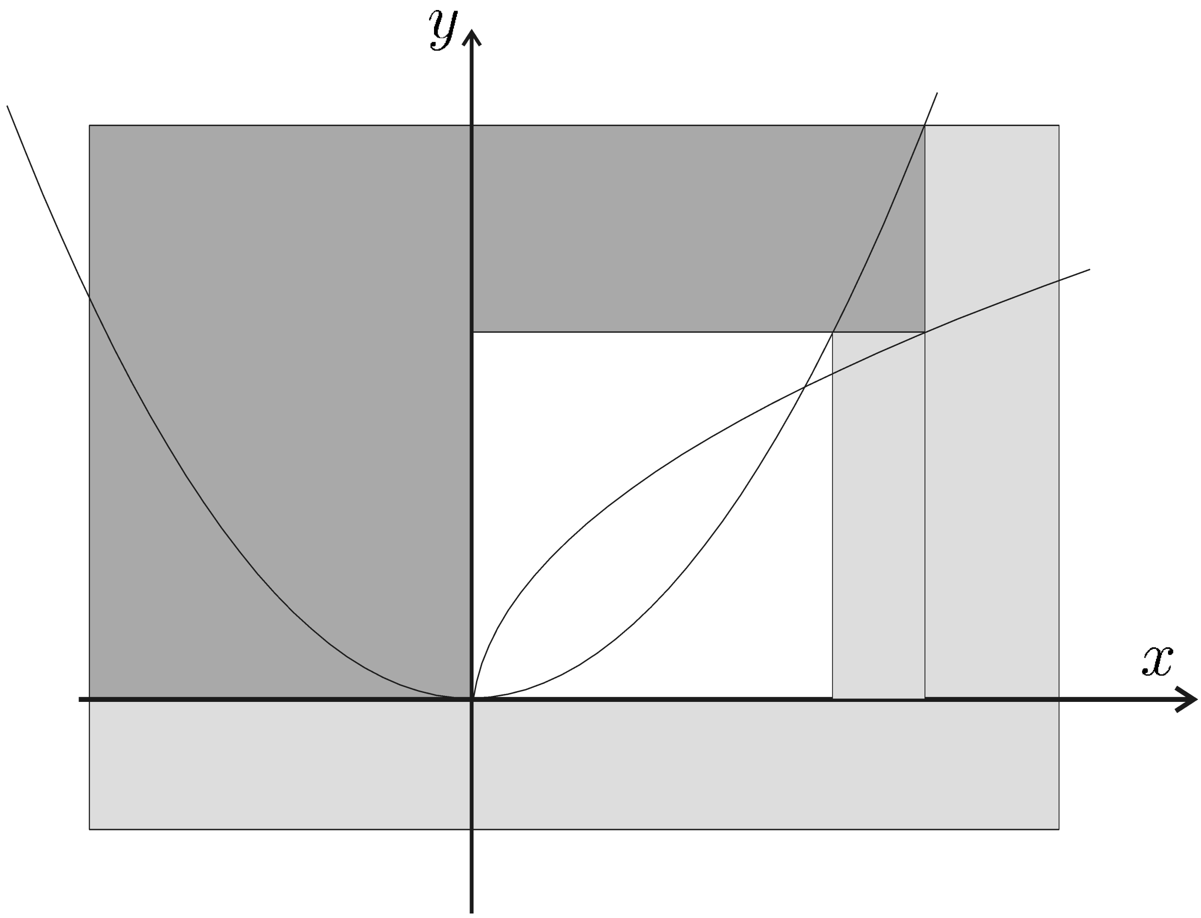
y
x



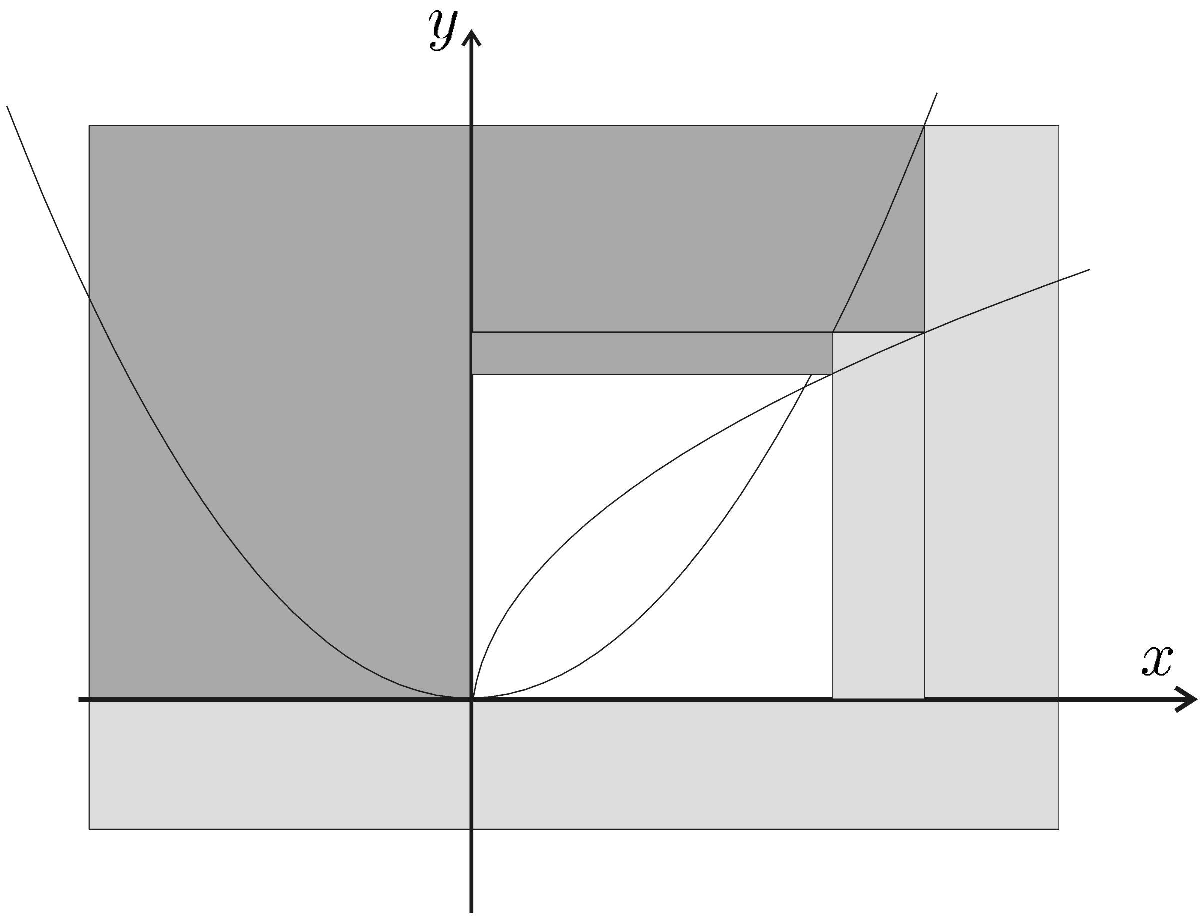
y
x



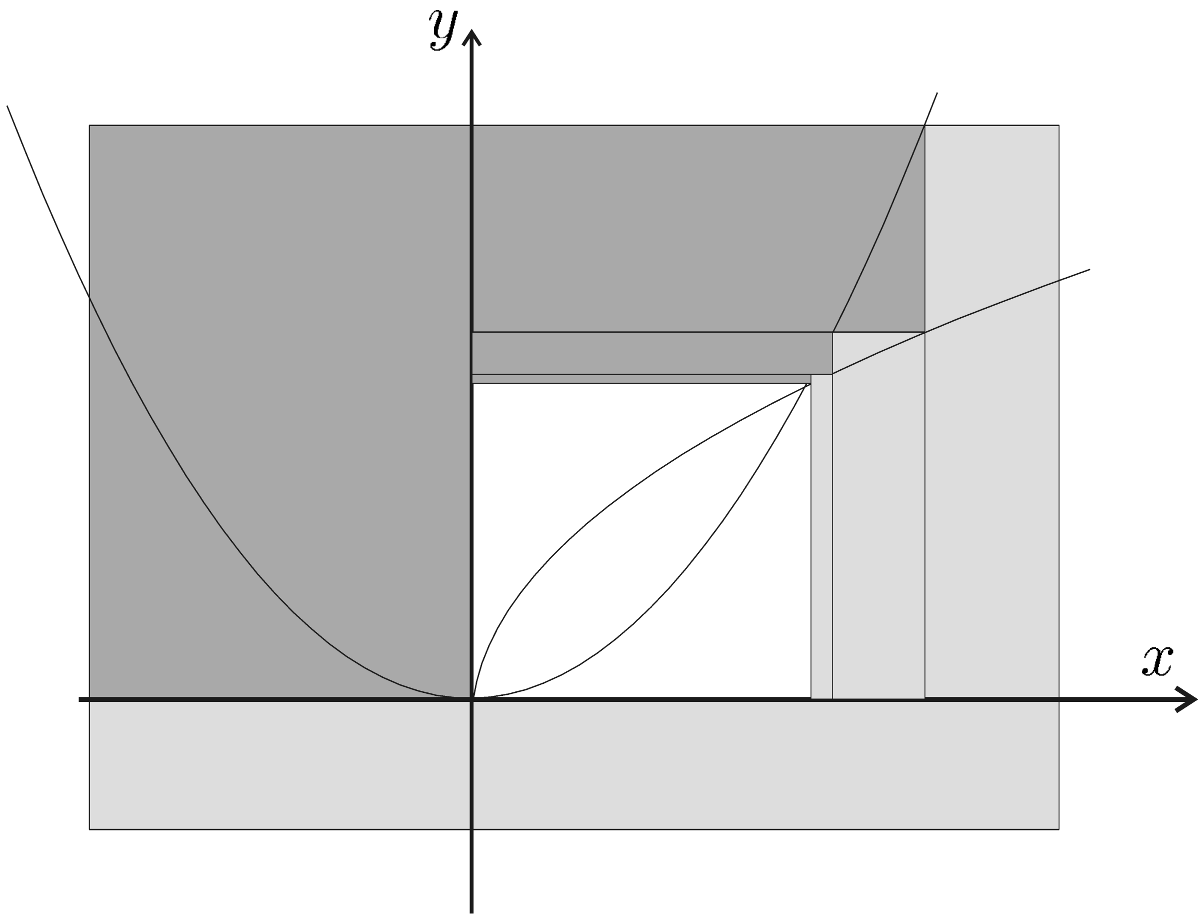
y
x



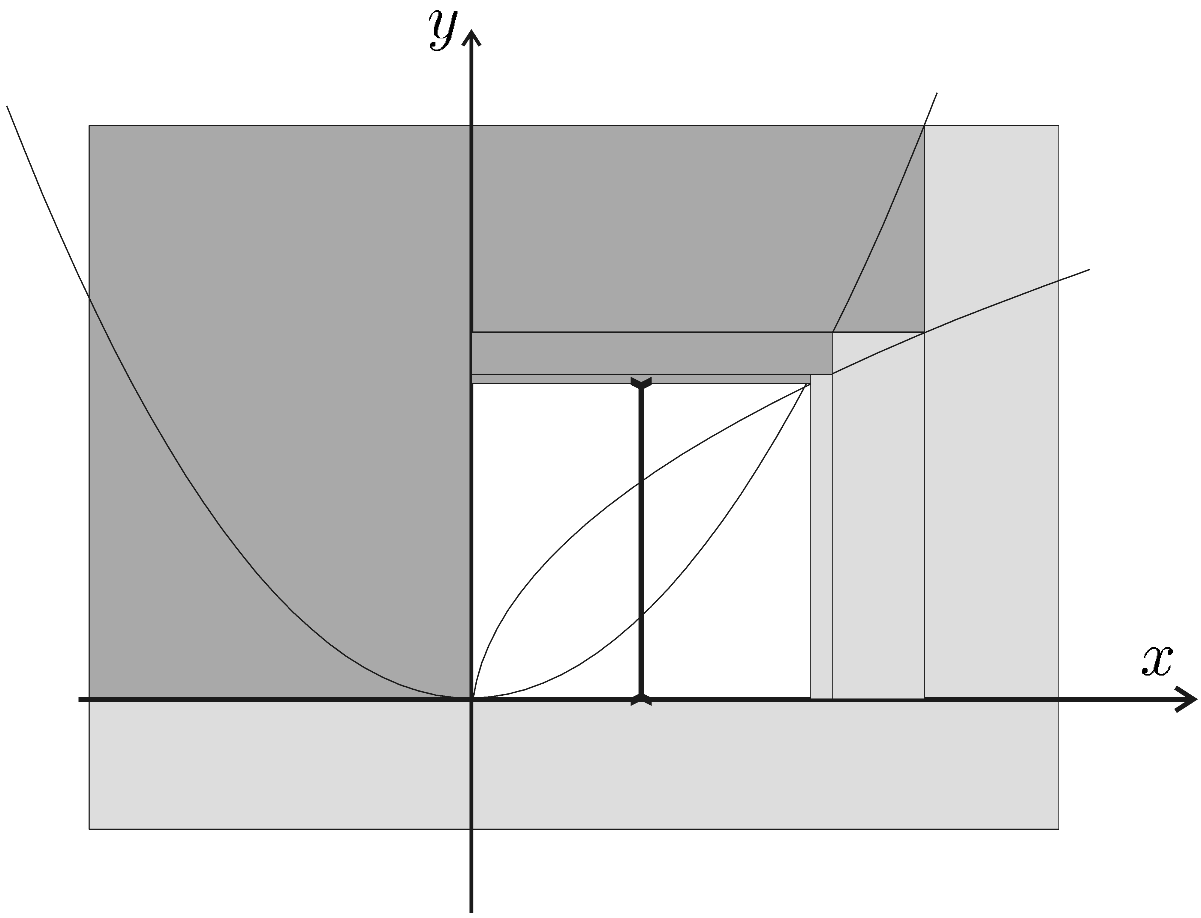
y
x



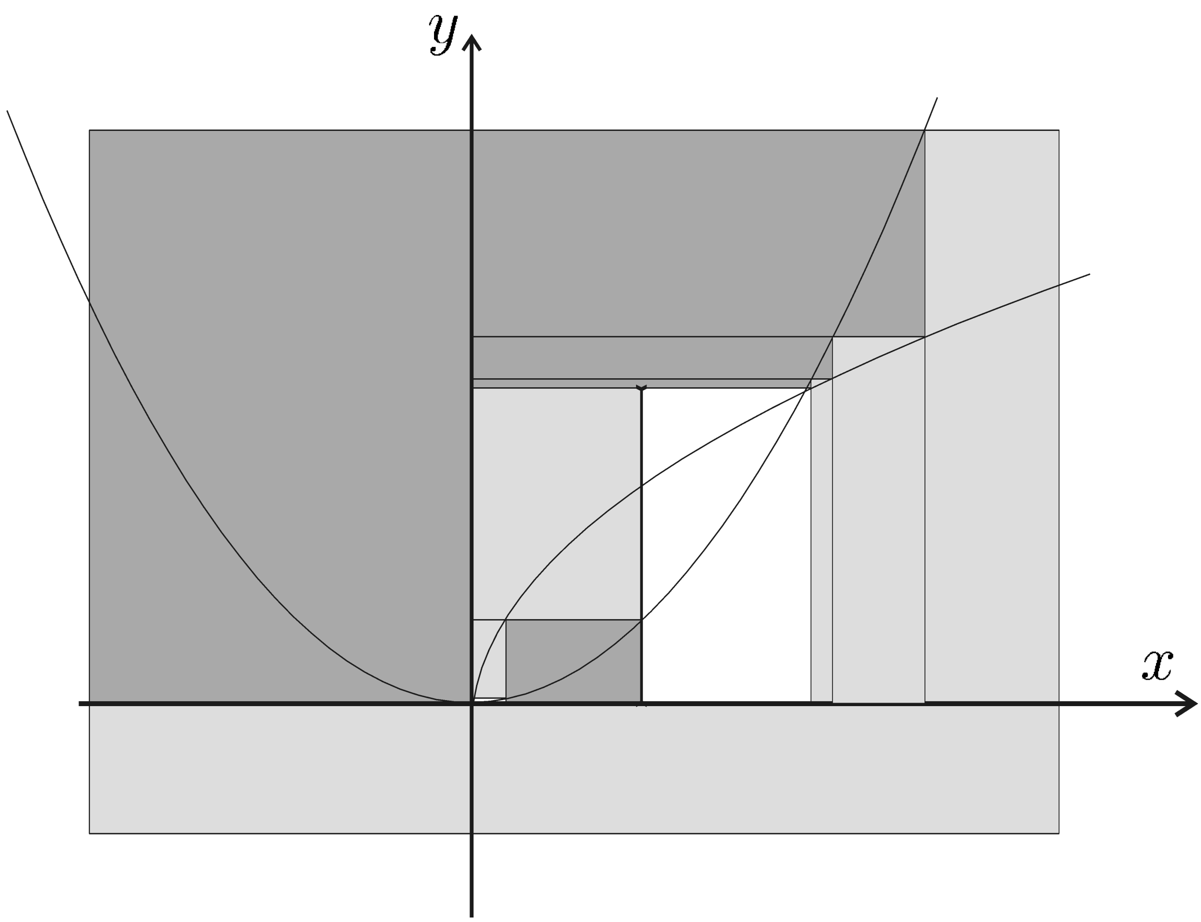
y
x



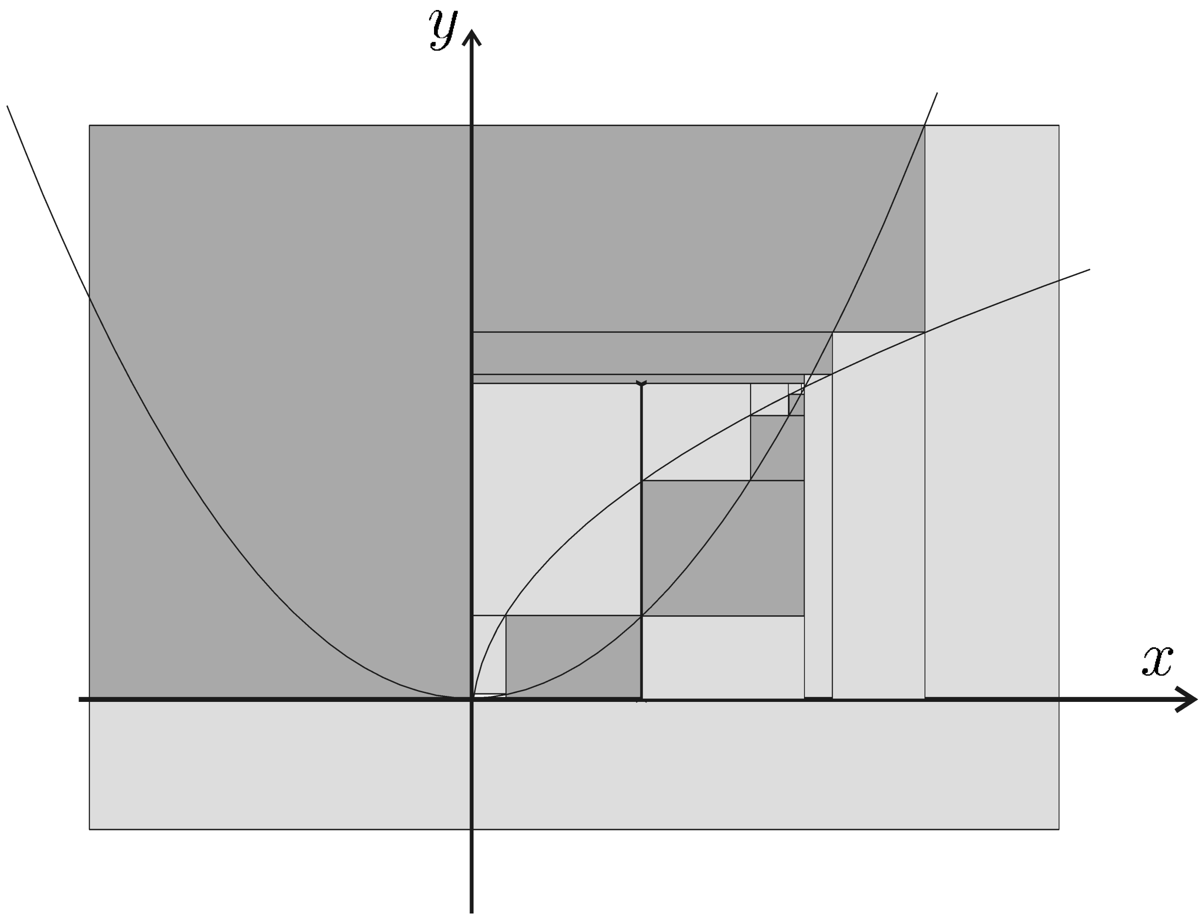
y
x



y
x



y
x



y
x

5 Local consistency

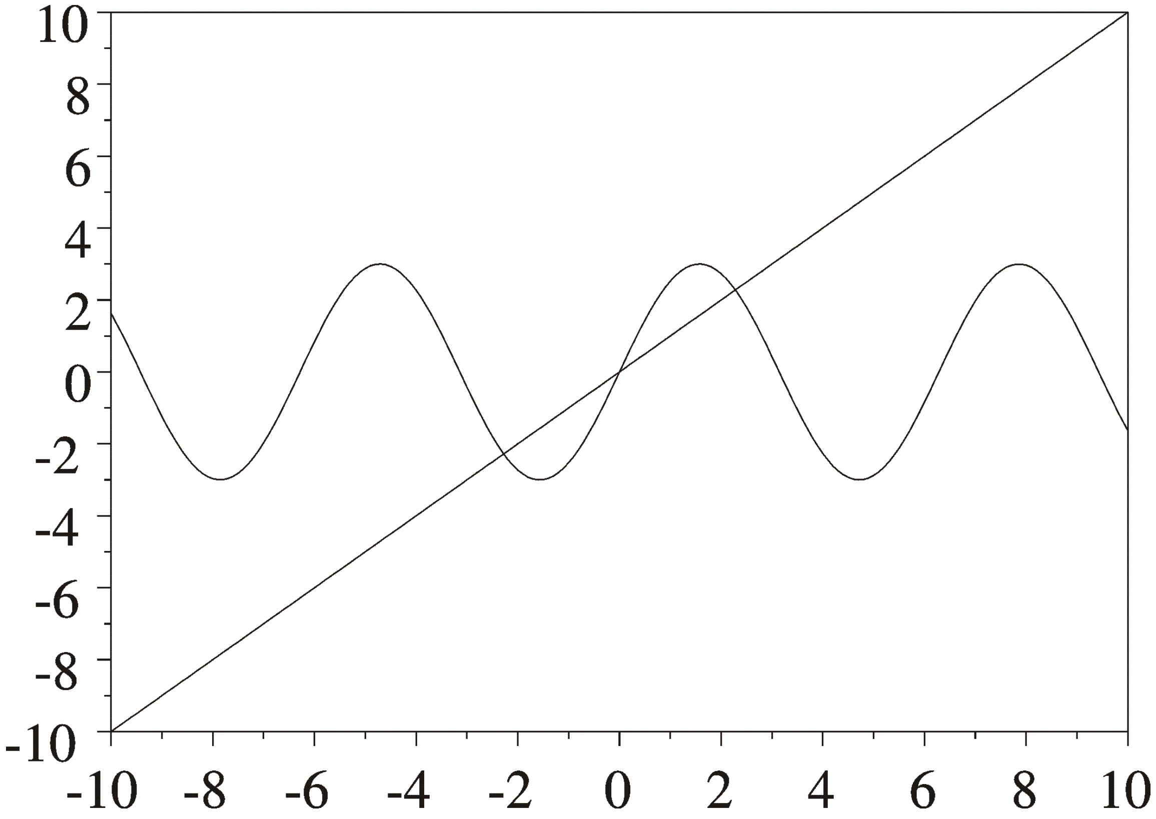
If $\mathcal{C}^*_{\mathbb{S}_1}$ and $\mathcal{C}^*_{\mathbb{S}_2}$ are two minimal contractors for $\mathbb{S}_1$ and $\mathbb{S}_2$ then

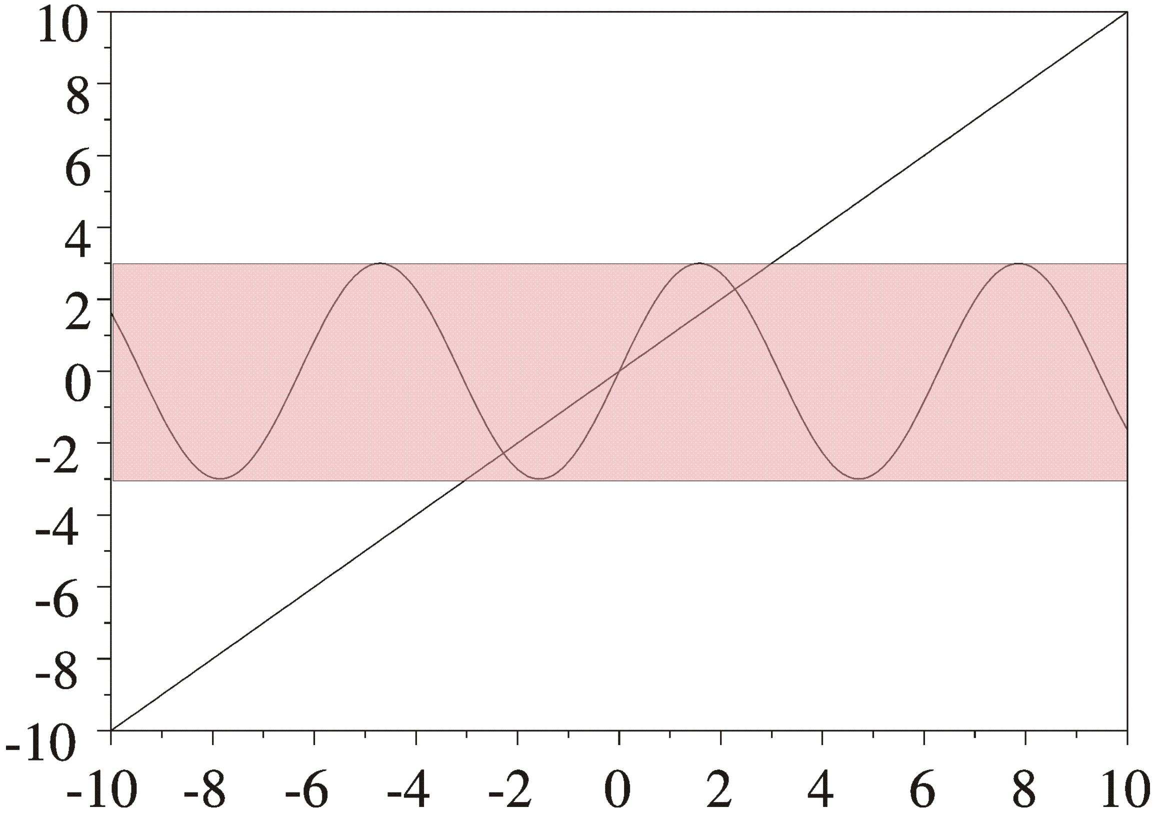
$$\mathcal{C}_{\mathbb{S}} = \mathcal{C}^*_{\mathbb{S}_1} \circ \mathcal{C}^*_{\mathbb{S}_2} \circ \mathcal{C}^*_{\mathbb{S}_1} \circ \mathcal{C}^*_{\mathbb{S}_2} \circ \dots$$

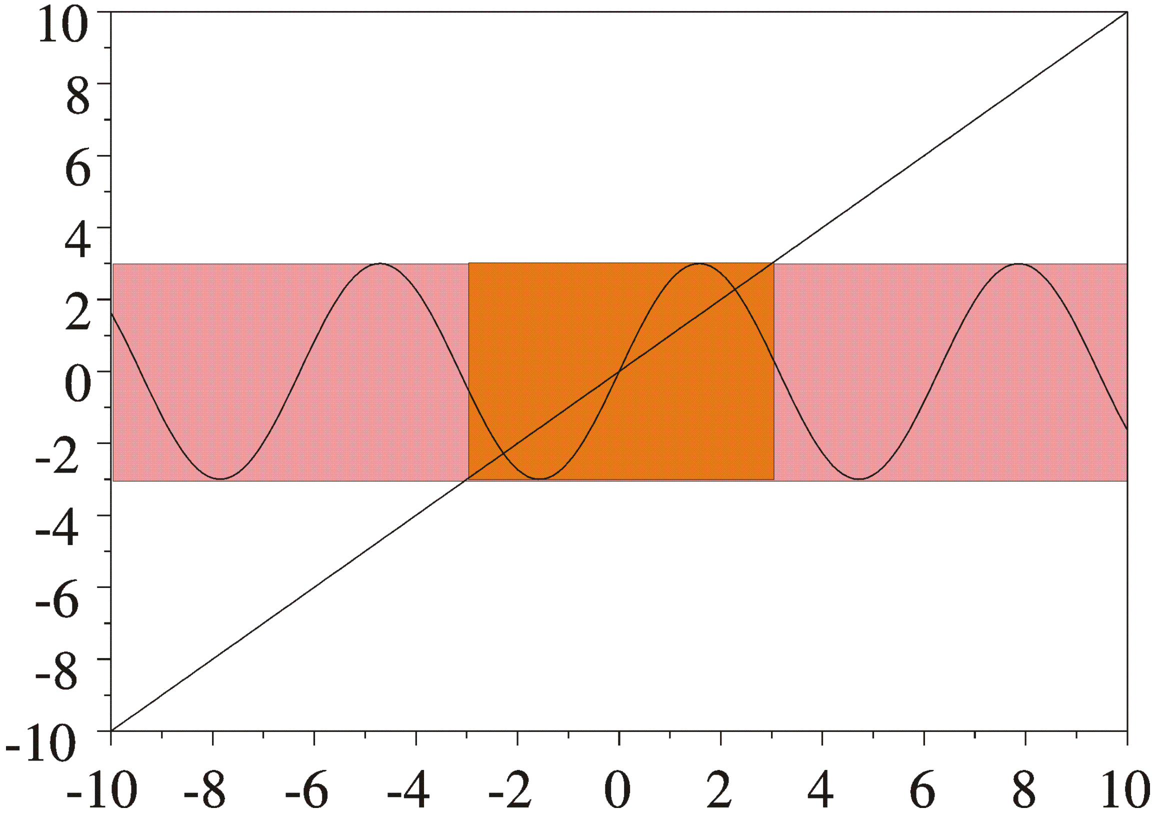
is a contractor for $\mathbb{S} = \mathbb{S}_1 \cap \mathbb{S}_2$, but it is not always optimal. This is the *local consistency effect.*

Exemple. Consider the system

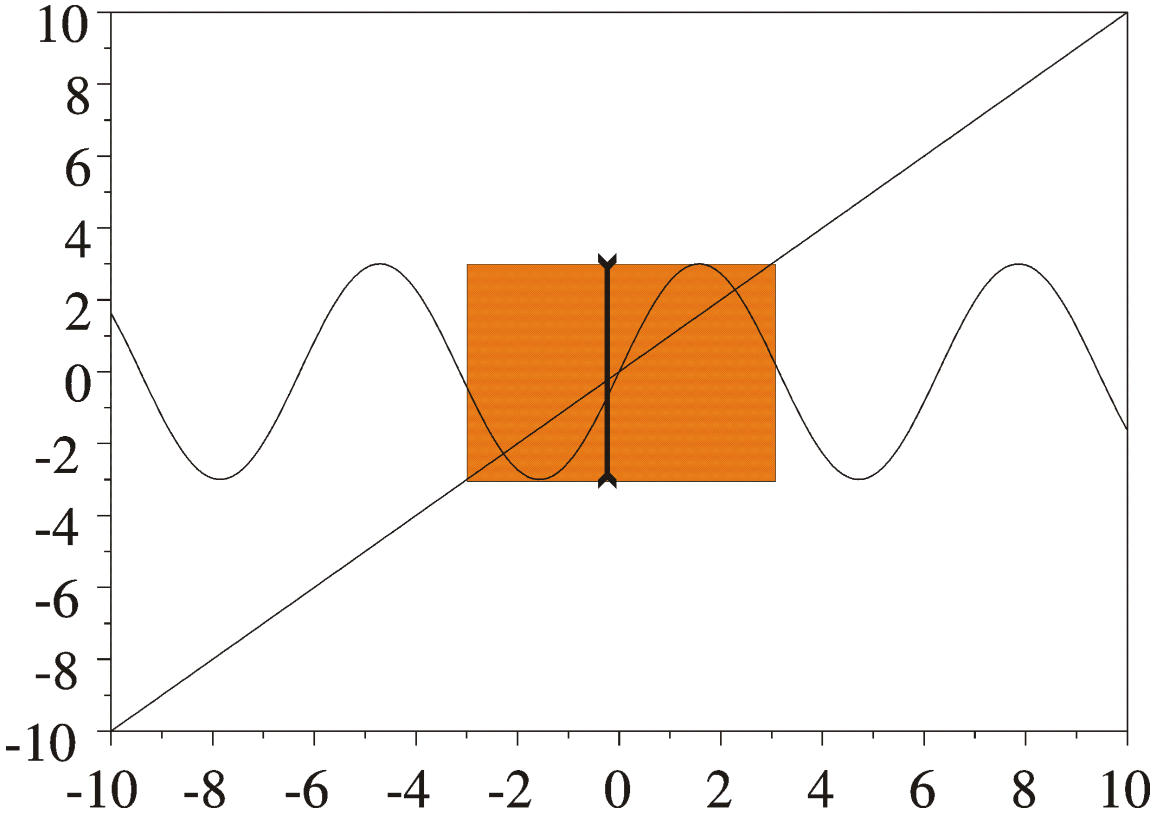
$$\begin{cases} y &= 3\sin(x) \\ y &= x \end{cases} \qquad x \in \mathbb{R},\ y \in \mathbb{R}.$$


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

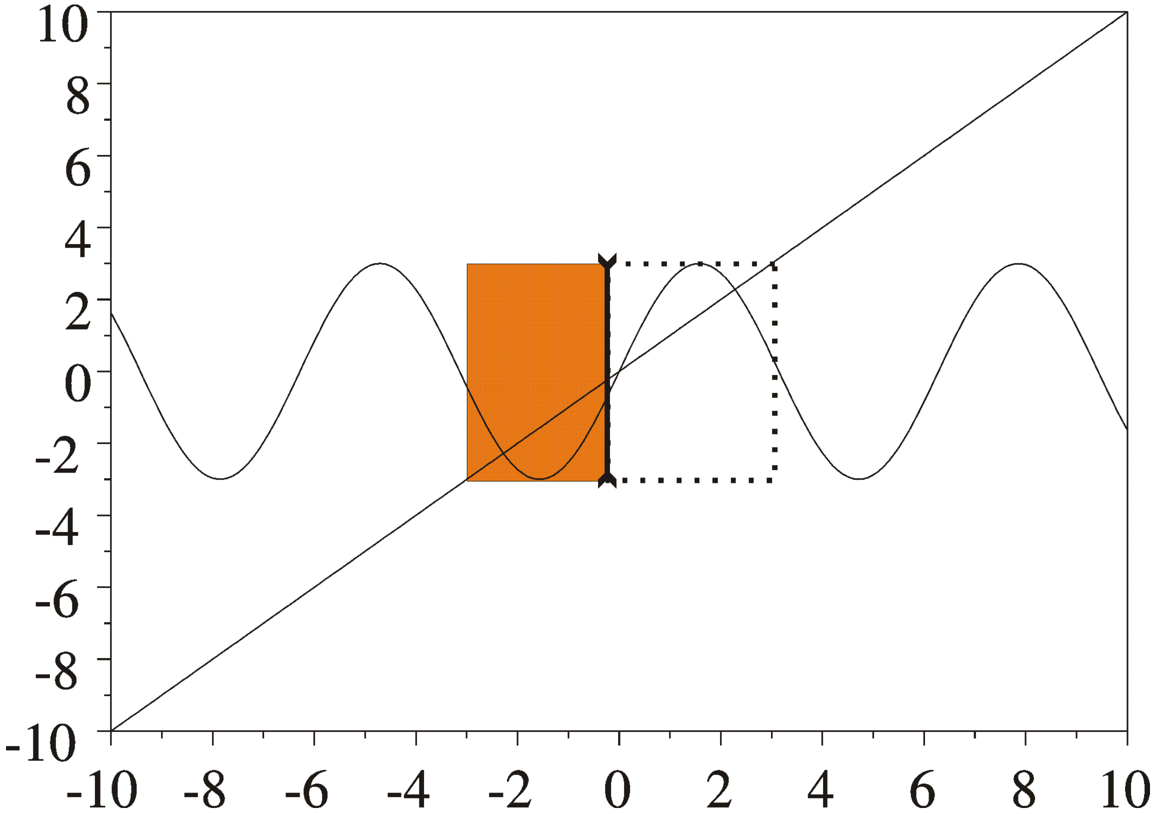

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10



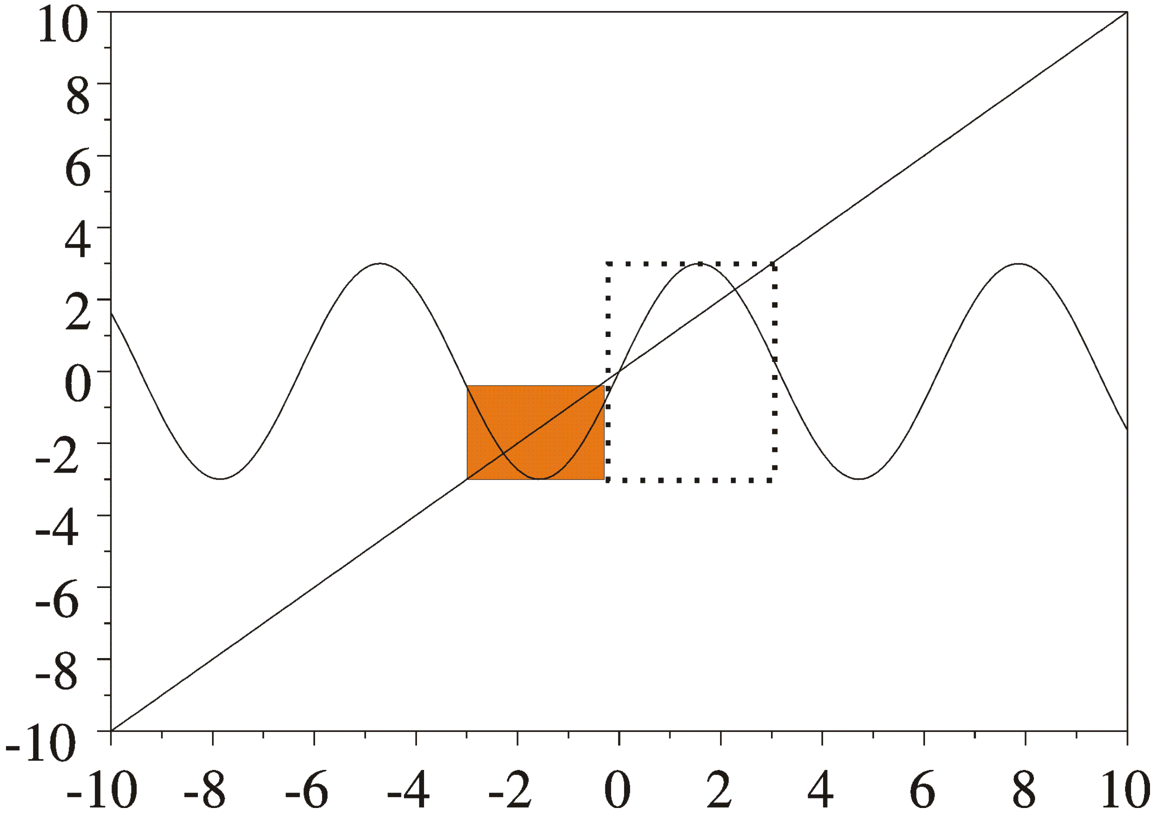
10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10



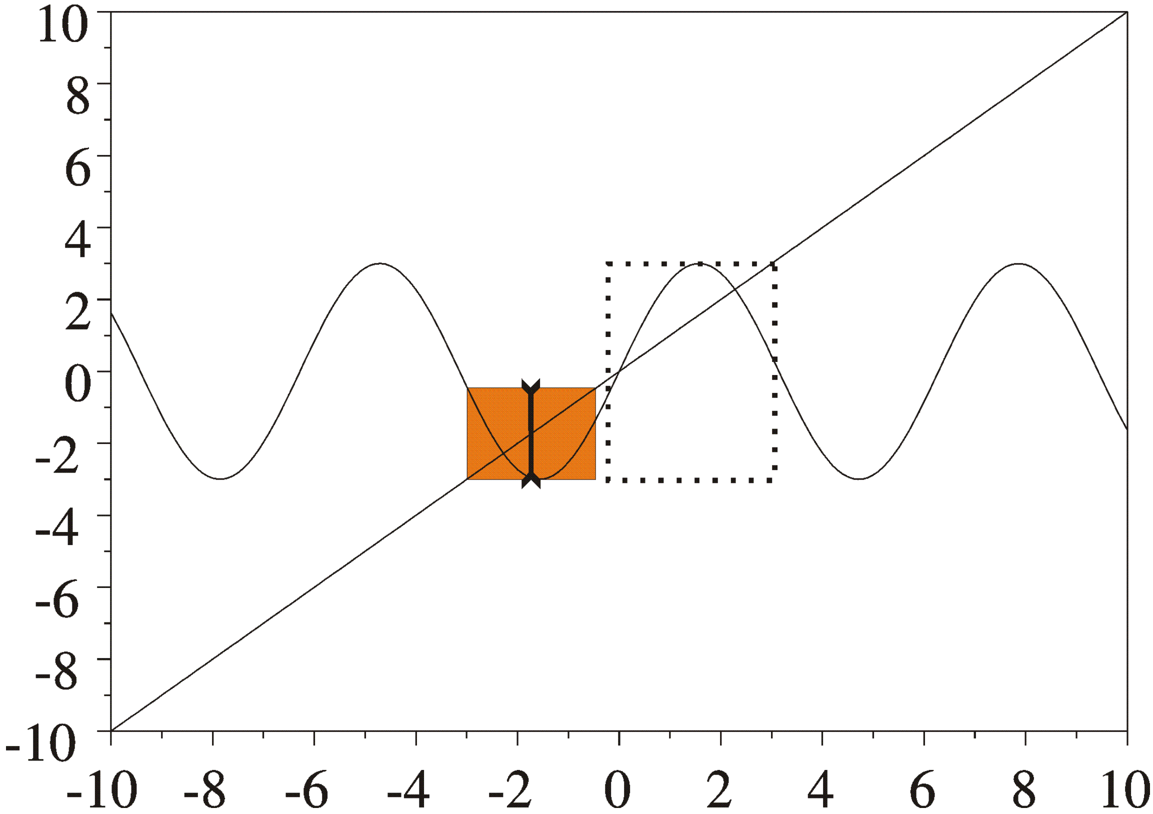
10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10



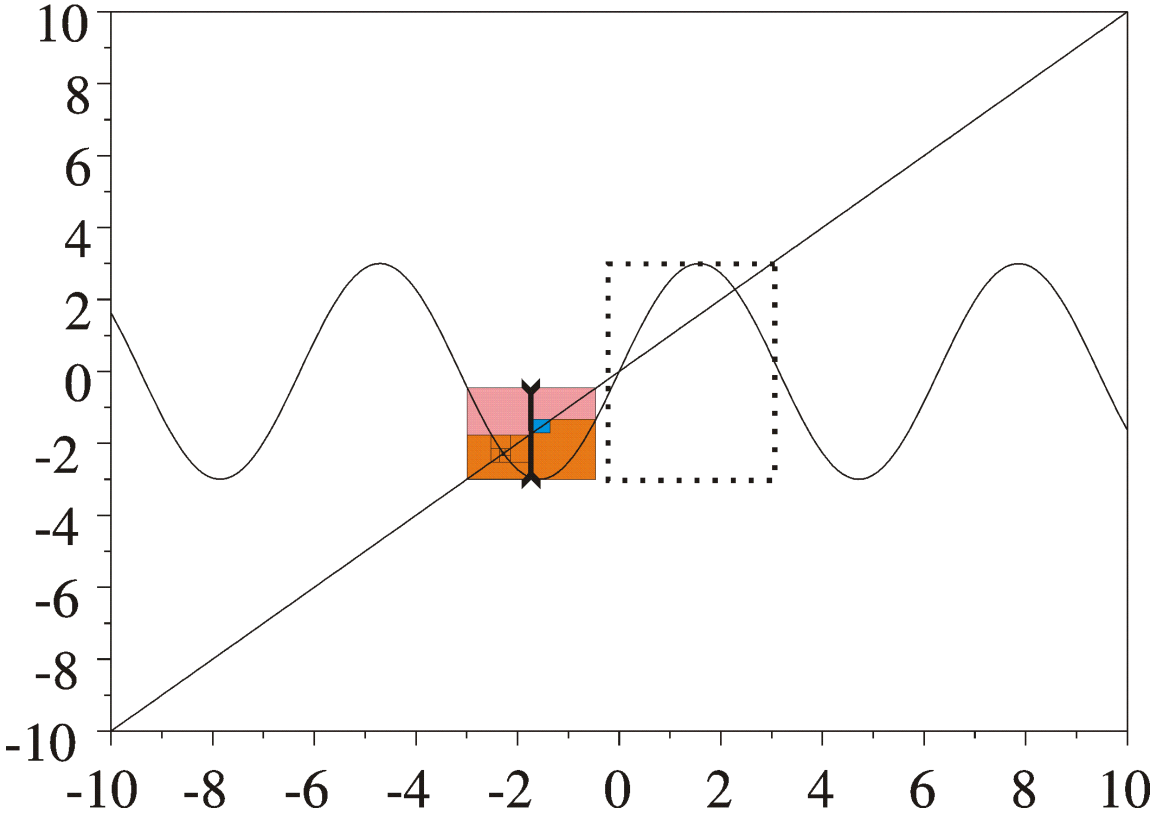
10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

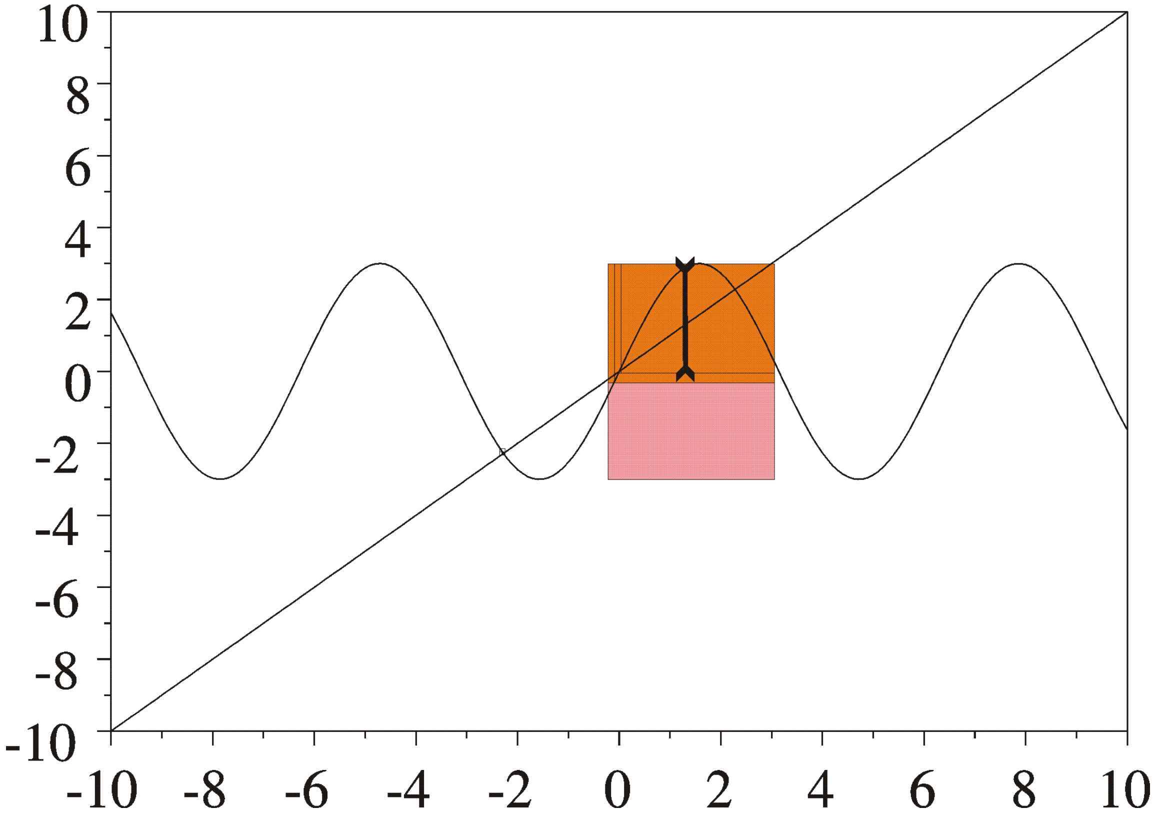


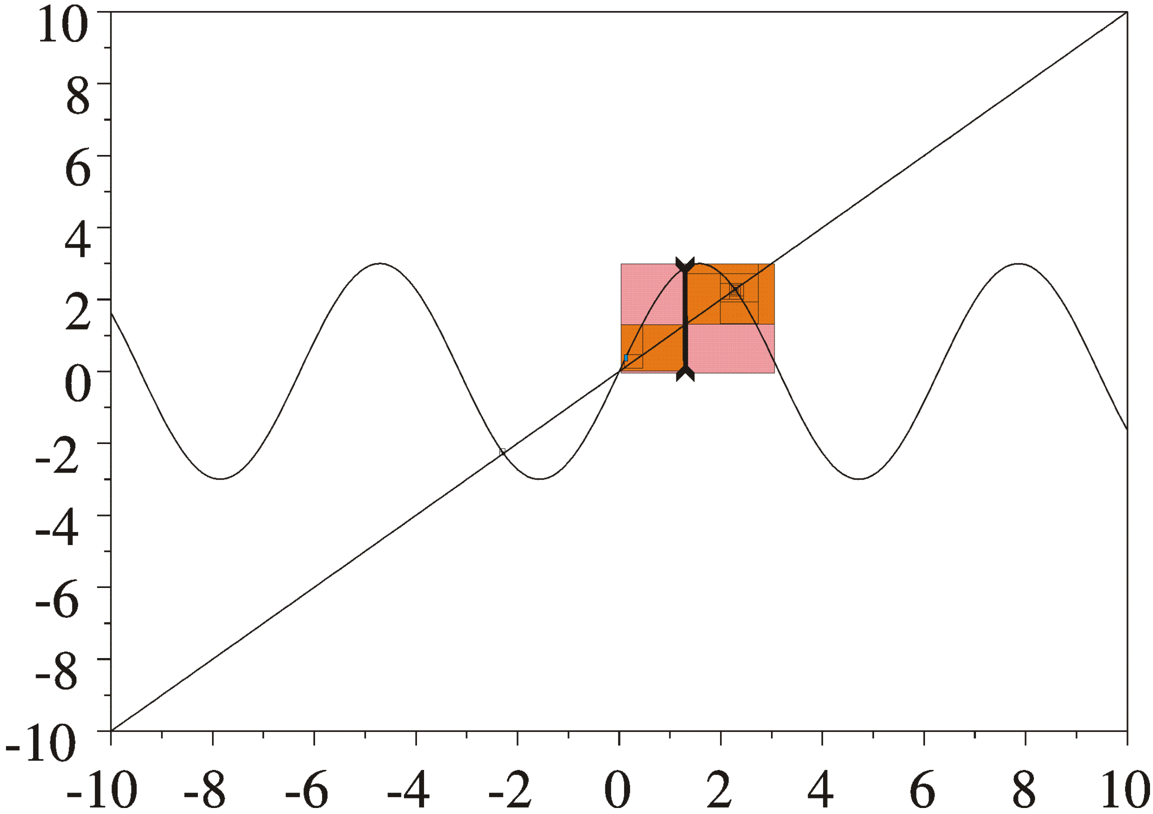
10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10



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

6 Decomposition into primitive constraints

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

can be decomposed into

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

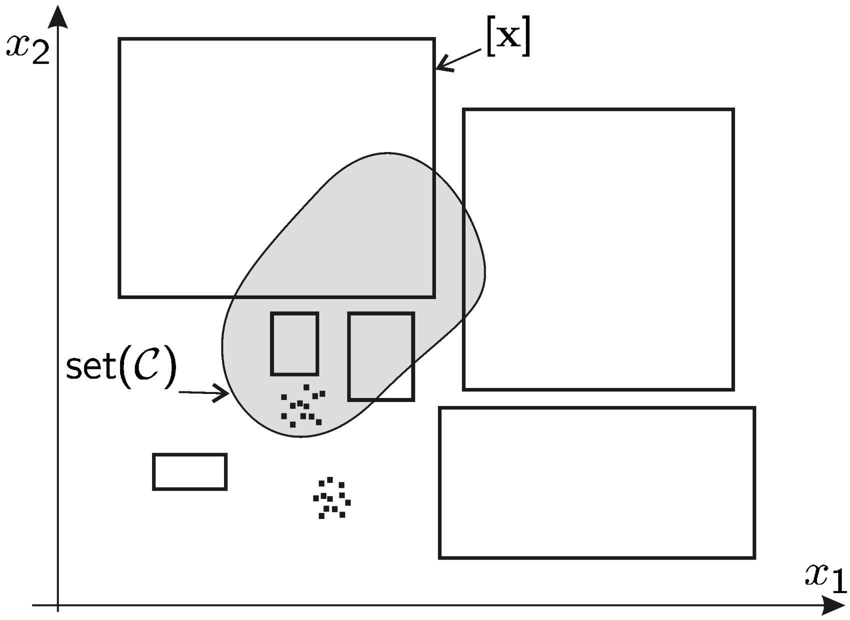
7 Set and contractors

A contractor represents a set of $\mathbb{R}^n$. The set associated with a contractor $\mathcal{C}$ is

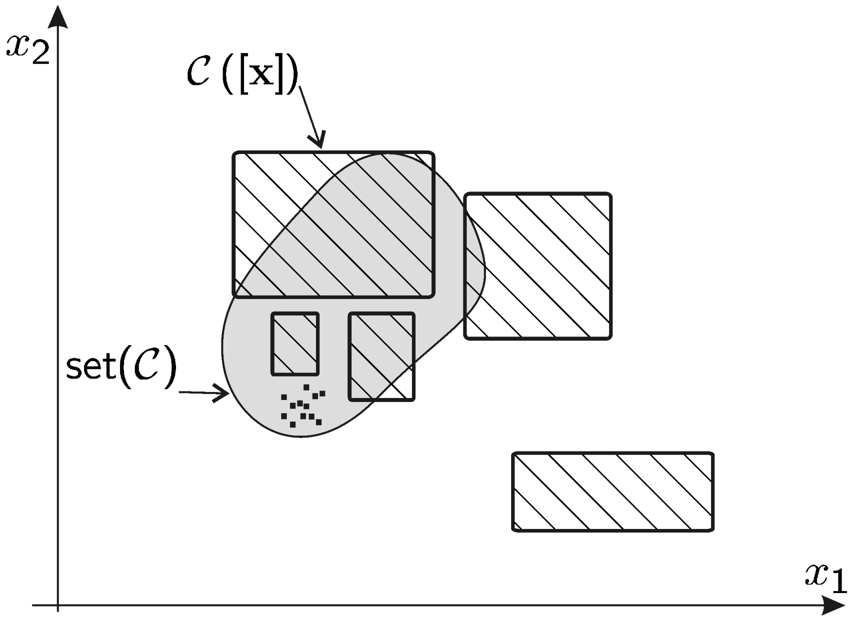
$$\mathrm{set}\,(\mathcal{C}) = \{\mathbf{x} \in \mathbb{R}^n, \mathcal{C}(\{\mathbf{x}\}) = \{\mathbf{x}\}\}\,.$$

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

For instance, the set associated with the contractor

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

is

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

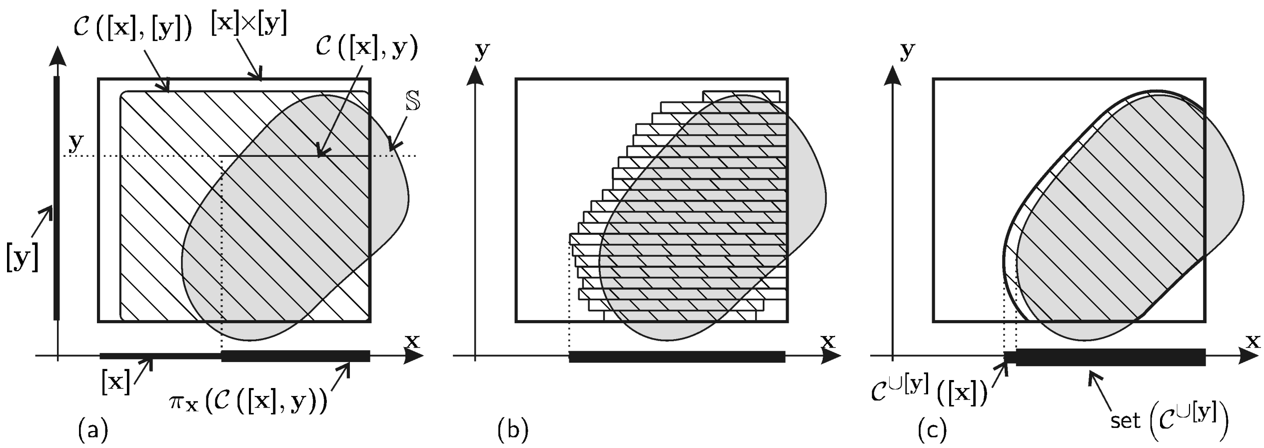
A contractor is also one way to represent one equation $x_3 = x_1 + x_2$.

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	$\mathcal{C}_1 \sqcup \mathcal{C}_2 = (\mathcal{C}_1 \cup \mathcal{C}_2)^{\infty}$

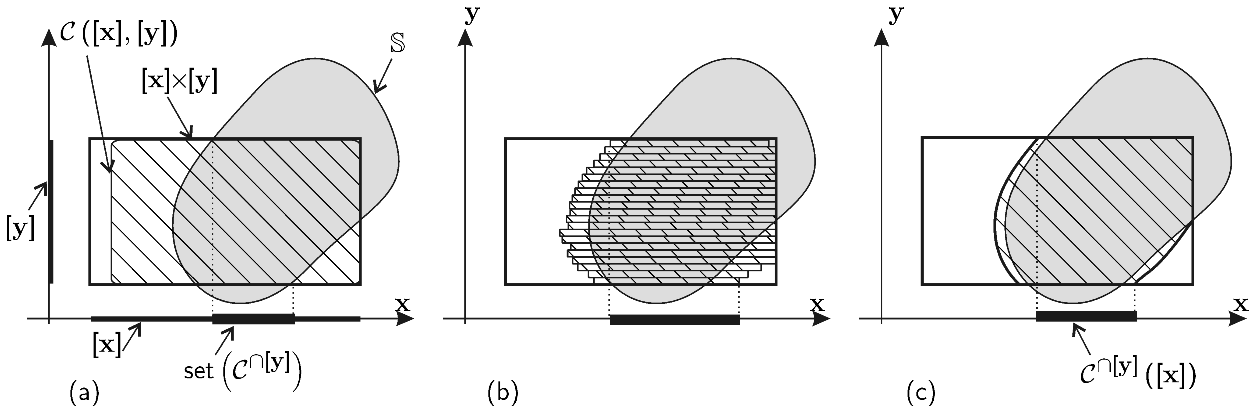
Consider the contractor $\mathcal{C}([\mathbf{x}],[\mathbf{y}])$, where $[\mathbf{x}] \in \mathbb{R}^n$, $[\mathbf{y}] \in \mathbb{R}^p$. We define the contractor

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



and also the contractor

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



We have

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

9 QUIMPER (or IBEX 2.0)

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

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

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.

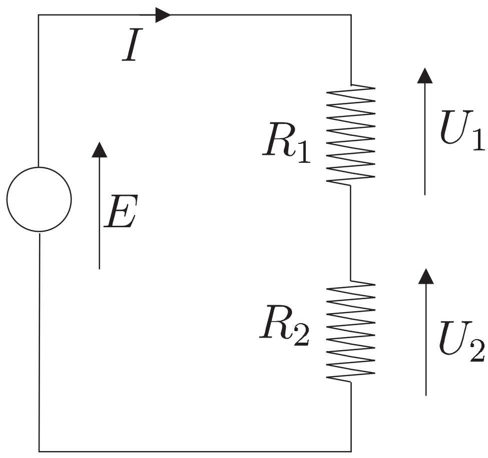
Quimper returns m subpavings, where m is the number of contractors

It is available at

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

10 Circuits

Example 1



Domains

$$\begin{aligned} E &\in [23V, 26V]; I \in [4A, 8A]; \\ U_1 &\in [10V, 11V]; U_2 \in [14V, 17V]; \\ P &\in [124W, 130W]; R_1 \in [0, \infty[\text{ and } R_2 \in [0, \infty[. \end{aligned}$$

Constraints

(i) $P = EI$, (ii) $E = (R_1 + R_2)\, I$, (iii) $U_1 = R_1 I$,
(iv) $U_2 = R_2 I$, (v) $E = U_1 + U_2$.

Solution set

$$\mathbb{S} = \left\{ \begin{pmatrix} E \\ R_1 \\ R_2 \\ I \\ U_1 \\ U_2 \\ P \end{pmatrix} \in \begin{pmatrix} [23, 26] \\ [0, \infty[\\ [0, \infty[\\ [4, 8] \\ [10, 11] \\ [14, 17] \\ [124, 130]; \end{pmatrix}, \quad \left\{ \begin{array}{l} P = EI \\ E = (R_1 + R_2)\, I \\ U_1 = R_1 I \\ U_2 = R_2 I \\ E = U_1 + U_2 \end{array} \right. \right\}$$

```
variables
 E in [23 ,26];
 I in [4,8];
 U1 in [10,11];
 U2 in [14 ,17];
 P in [124,130];
 R1 in [0 ,1e08 ];
 R2 in [0 ,1e08 ];
contractor_list L
 P=E*I;
 E=(R1+R2)*I;
 U1=R1*I;
 U2=R2*I;
 E=U1+U2;
end
```

```
contractor C
   compose(L);
end
contractor epsilon
   precision(1);
end
```

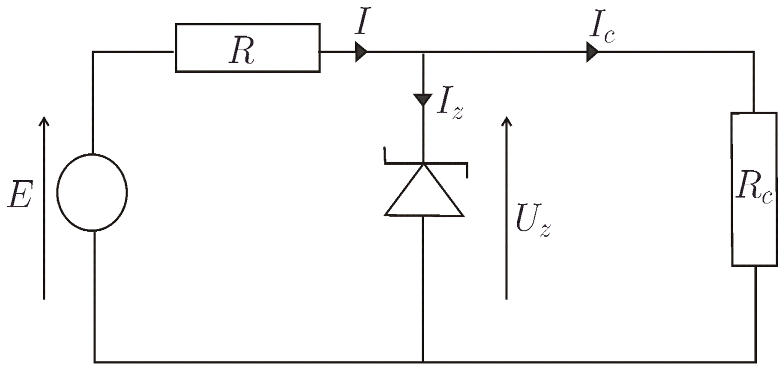
Quimper returns

$$[24; 26] \times [1.846; 2.307] \times [2.584; 3.355]$$
$$\times [4.769; 5.417] \times [10; 11] \times [14; 16] \times [124; 130]\,,$$

i.e.,

$$\begin{array}{ll} E \in [24; 26]\,, & R_1 \in [1.846; 2.307]\,, \\ R_2 \in [2.584; 3.355], & I \in [4.769; 5.417]\,, \\ U_1 \in [10; 11]\,, & U_2 \in [14; 16]\,, \\ P \in [124; 130]\,. & \end{array}$$

Example 2



It is known that

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

The constraints are

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

Quimper contracts the domains into:

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

Exercise.

A robot measures its own distance to three marks. The distances and the coordinates of the marks are as follows

mark	x_i	y_i	d_i
1	0	0	$[22, 23]$
2	10	10	$[10, 11]$
3	30	-30	$[53, 54]$

Build the contractor associated with the pose of the robot.

11 Proving 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

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

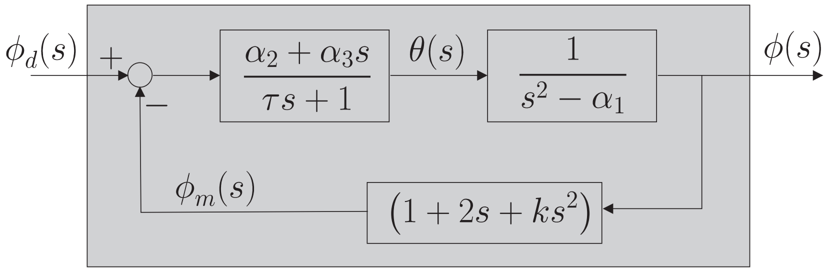
Consider a motorbike with a speed of 1m/s.

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 :

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

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

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

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

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

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.

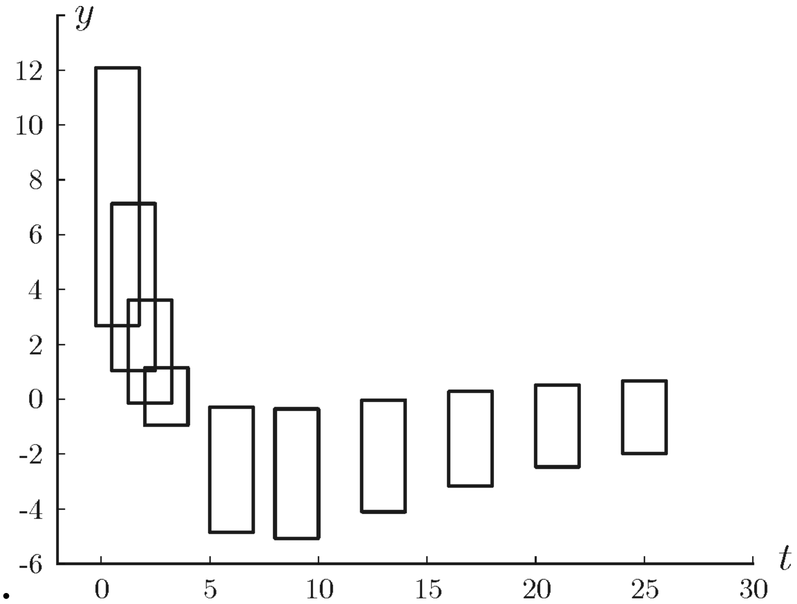
This is easily proven by Quimper

```
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
```

12 Estimation problem

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



i	$[t_i]$	$[y_i]$
1	$[-0.25, 1.75]$	$[2.7, 12.1]$
2	$[0.5, 2.5]$	$[1.04, 7.14]$
3	$[1.25, 3.25]$	$[-0.13, 3.61]$
4	$[2, 4]$	$[-0.95, 1.15]$
5	$[5, 7]$	$[-4.85, -0.29]$
6	$[8, 10]$	$[-5.06, -0.36]$
7	$[12, 14]$	$[-4.1, -0.04]$
8	$[16, 18]$	$[-3.16, 0.3]$
9	$[20, 22]$	$[-2.5, 0.51]$
10	$[24, 26]$	$[-2, 0.67]$

The feasible set is

$$\mathbb{S} \;=\; \bigcap_{i\in\{1,\dots,10\}} \underbrace{\left\{\mathbf{p}\in\mathbb{R}^2 \mid \exists t_i\in[t_i] \mid y_m(\mathbf{p},t_i)\in[y_i]\right\}}_{\mathbb{S}_i}.$$

The complementary set is

$$\bar{\mathbb{S}} \;=\; \bigcup_{i\in\{1,\dots,10\}} \underbrace{\left\{\mathbf{p}\in\mathbb{R}^2 \mid \forall t_i\in[t_i] \mid y_m(\mathbf{p},t_i)\notin[y_i]\right\}}_{\bar{\mathbb{S}}_i}$$

Define two contractors $\mathcal{C}_i(\mathbf{p},t_i)$ and $\bar{\mathcal{C}}_i(\mathbf{p},t_i)$ such that

$$\left\{\begin{array}{lll}\text{set}\left(\mathcal{C}_i(\mathbf{p},t_i)\right) & = & \{(\mathbf{p},t_i), y_m(\mathbf{p},t_i)\in[y_i]\}\\ \text{set}\left(\bar{\mathcal{C}}_i(\mathbf{p},t_i)\right) & = & \{(\mathbf{p},t_i), y_m(\mathbf{p},t_i)\notin[y_i]\}.\end{array}\right.$$

We have

$$\begin{array}{lll}\text{set}\left(\mathcal{C}_i^{\cup[t_i]}\right) & = & \mathbb{S}_i\\ \text{set}\left(\bar{\mathcal{C}}_i^{\cap[t_i]}\right) & = & \bar{\mathbb{S}}_i.\end{array}$$

Define two contractors

$$\begin{array}{rcl}\mathcal{C}\left([\mathbf{p}]\right) & = & \bigcap\limits_{i\in\{1,\dots,10\}} \mathcal{C}_i^{\cup[t_i]}\left([\mathbf{p}],t_i\right)\\ \bar{\mathcal{C}}\left([\mathbf{p}]\right) & = & \bigcup\limits_{i\in\{1,\dots,10\}} \bar{\mathcal{C}}_i^{\cap[t_i]}\left([\mathbf{p}],t_i\right).\end{array}$$

We have $\text{set}(\mathcal{C}) = \mathbb{S}$ et $\text{set}\left(\bar{\mathcal{C}}\right) = \bar{\mathbb{S}}$.

```
constant
  Y[10] = [[2.7,12.1]; [1.04,7.14];
           [-0.13,3.61];[-0.95,1.15];
           [-4.85,-0.29];[-5.06,-0.36];
           [-4.1,-0.04]; [-3.16,0.3];
           [-2.5,0.51]; [-2,0.67]];
variables
  p1 in [0,1.2]; p2 in [0,0.5];
parameters
  t[10] in [[-0.25,1.75]; [0.5,2.5]; [1.25,3.25];
            [2,4]; [5,7]; [8,10]; [12,14];
            [16,18]; [20,22]; [24,26]];
function z=f(p1,p2,t)
  z=20*exp(-p1*t)-8*exp(-p2*t);
end
```

```
contractor outer
   inter (i=1:10,
       proj_union(f(p1,p2,t[i]) in Y[i]),t[i]);
   end
end
contractor inner
   union (i=1:10,
       proj_inter(f(p1,p2,t[i]) notin Y[i]),t[i]);
   end
end
contractor epsilon
   precision(0.01)
en
```