

Méthodes par intervalles, propagation de contraintes et
calcul formel
Principes et applications

Luc Jaulin, Xavier Baguenard, Massa Dao, Pau Herrero

LISA-Angers
Journées nationales de calcul formel
22 janvier 2003

Les méthodes intervalles permettent la résolution numérique mais garantie de problèmes non-linéaires, comme par exemple

- l'optimisation globale de critères non convexes,
- la résolution de systèmes d'égalités et/ou inégalités,
- intégration d'équations différentielles.
- preuve de théorème, . . .

Elles ont été utilisées en automatique et en robotique

- estimation de paramètres et d'états,
- path planning,
- commande robuste, . . .

Projection de contraintes

Soient 3 variables x, y, z telles que

$$x \in [1, 5]$$

$$y \in [2, 4]$$

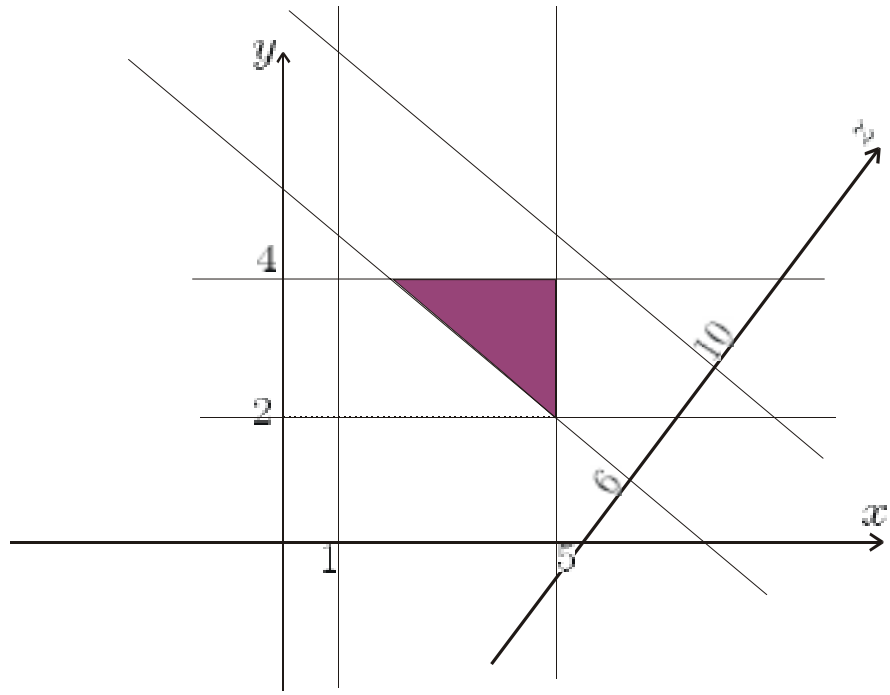
$$z \in [6, 10]$$

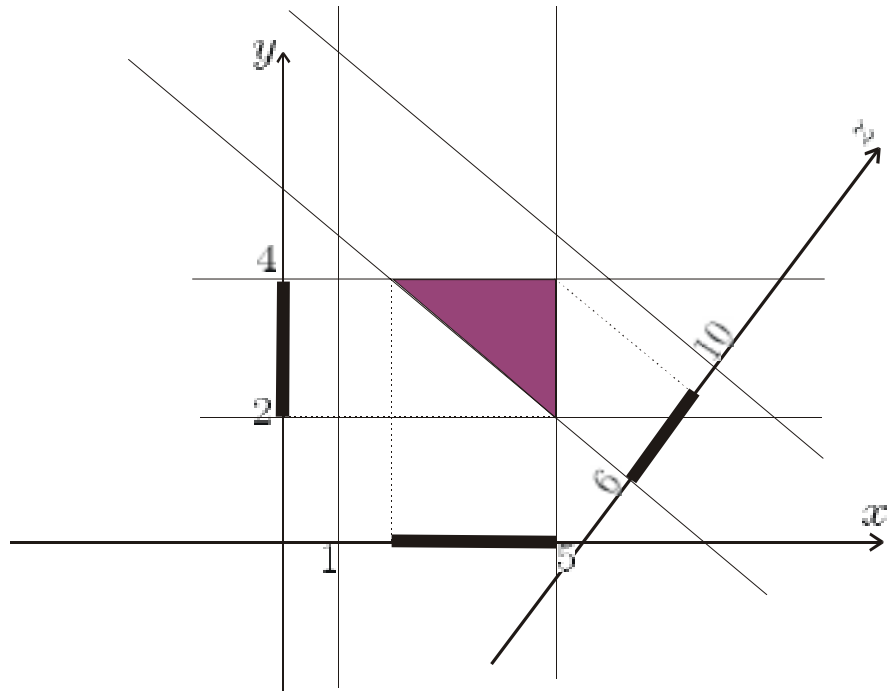
$$z = x + y$$

Les valeurs 1 pour x et 10 pour z sont dites inconsistantes.

Projeter une contrainte (ici, $z = x + y$), c'est calculer les plus petits intervalles qui ne contiennent que des valeurs consistantes. Pour notre exemple, cela revient à projeter 3 fois (suivant x, y , puis z) le sous-ensemble de \mathbb{R}^3 donné par

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





Propagation de contraintes

Soient les trois contraintes suivantes

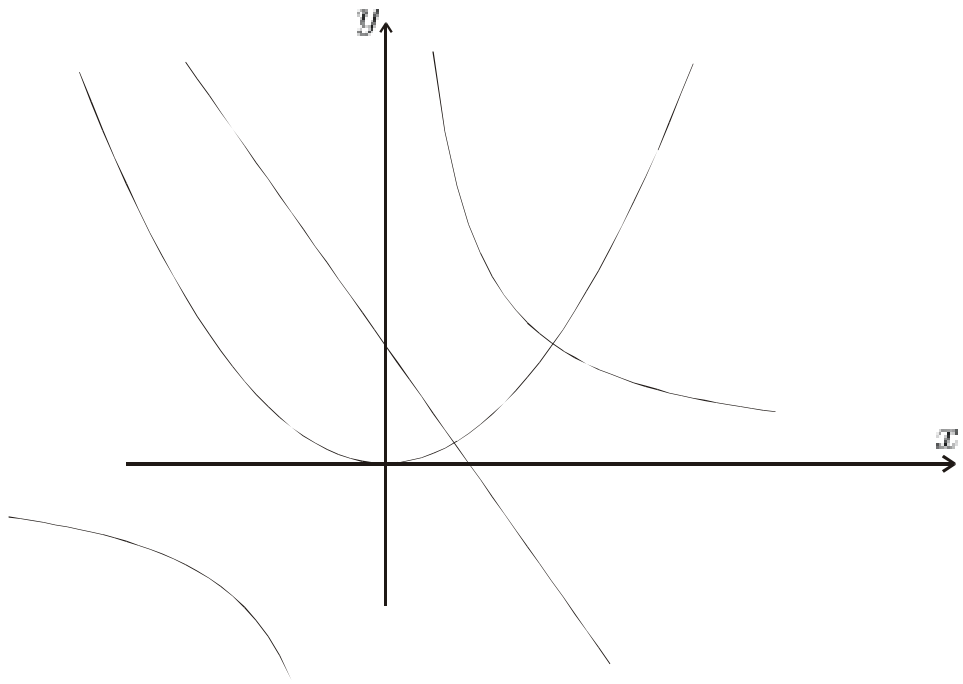
$$(C_1) : y = x^2$$

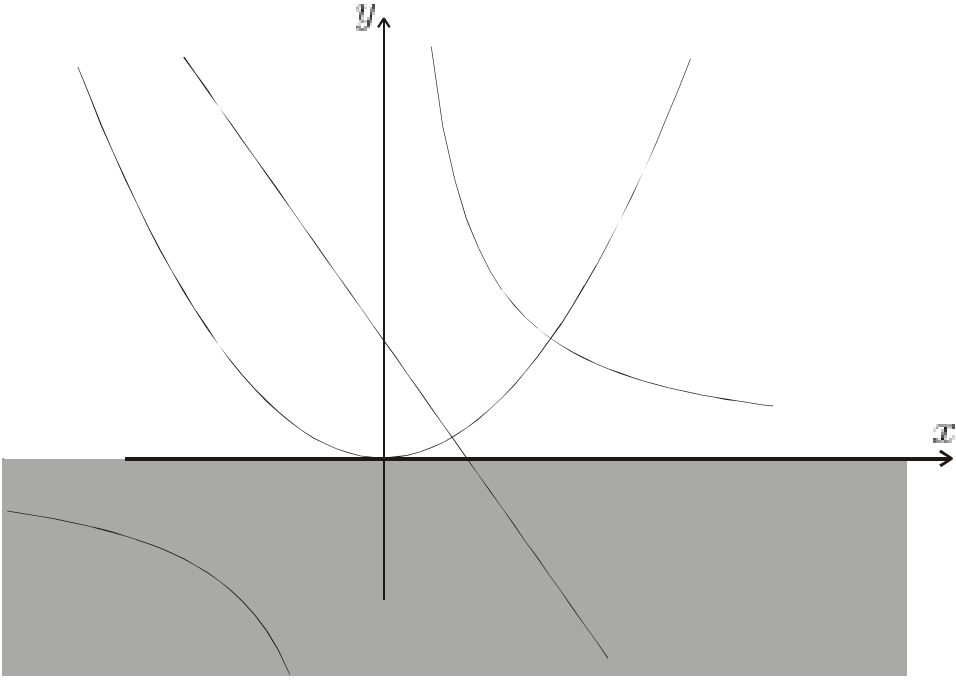
$$(C_2) : xy = 1$$

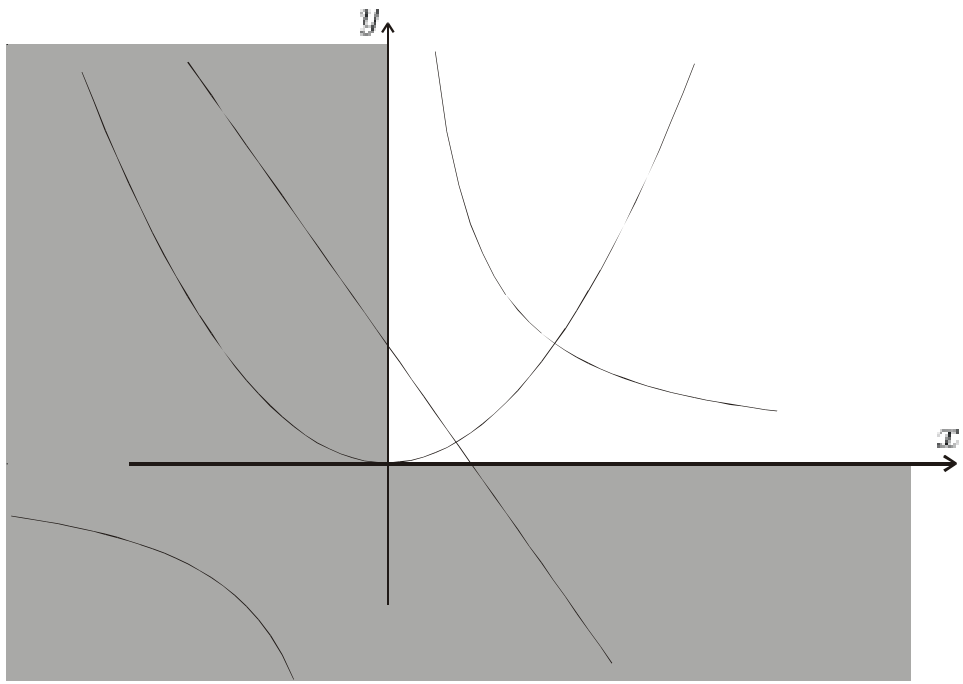
$$(C_3) : y = -2x + 1$$

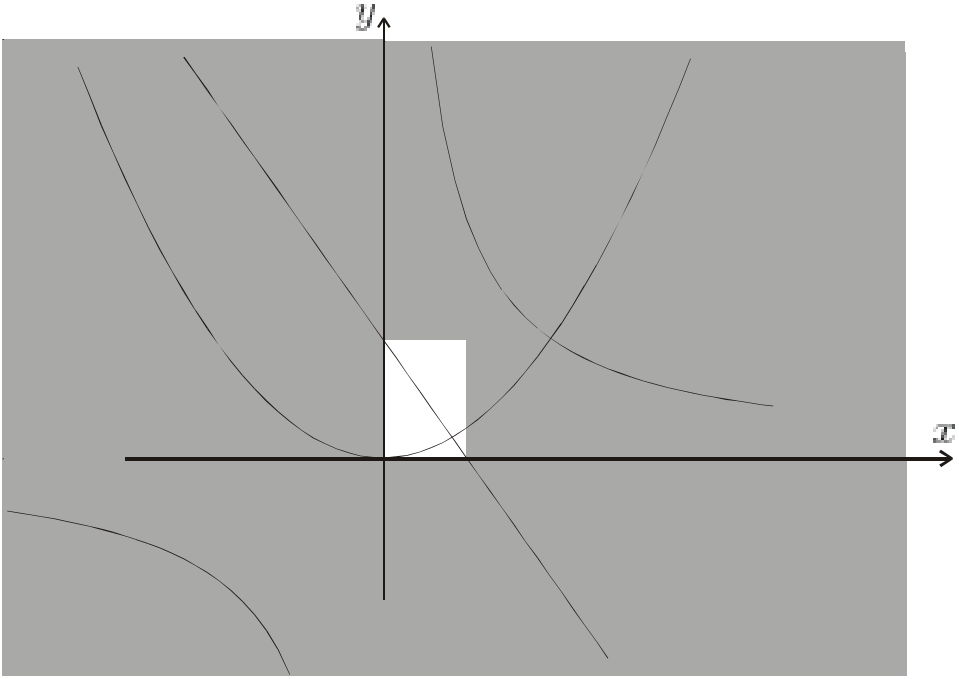
A chaque variable on effectue le domaine $] - \infty, \infty[$.

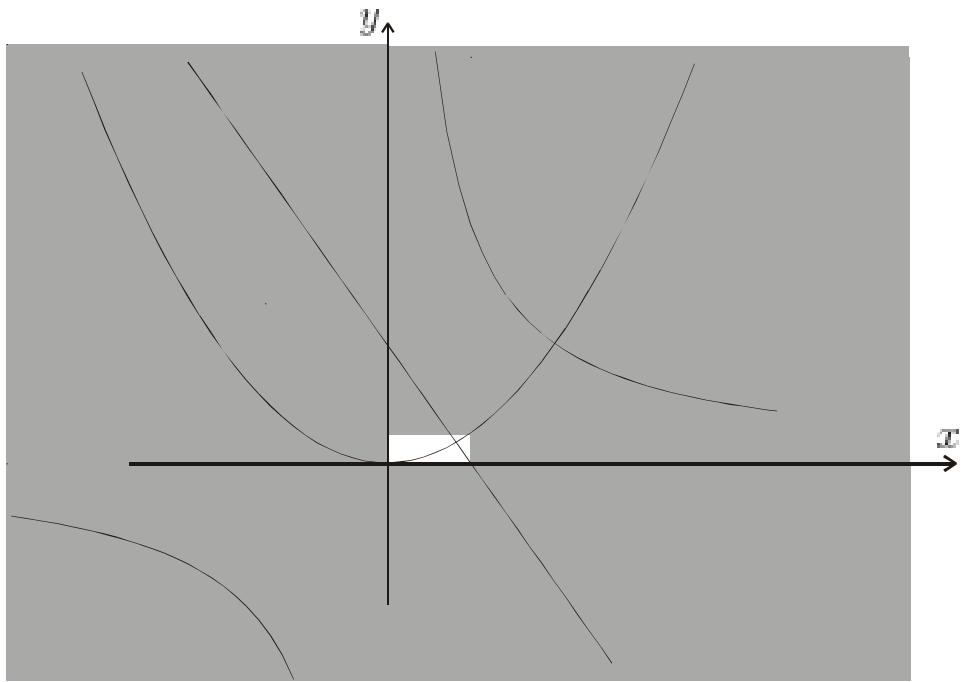
Une propagation de contrainte consiste à projeter les contraintes jusqu'au blocage.











Cas où ça marche moins bien

$$\begin{cases} (C_1) : y = x \\ (C_2) : y = -x \\ x \in [-1, 3], y \in [-1, 3] \end{cases}$$

Décomposition en contraintes primitives

Pour les contraintes plus complexes, il nous faut effectuer une décomposition. Par exemple,

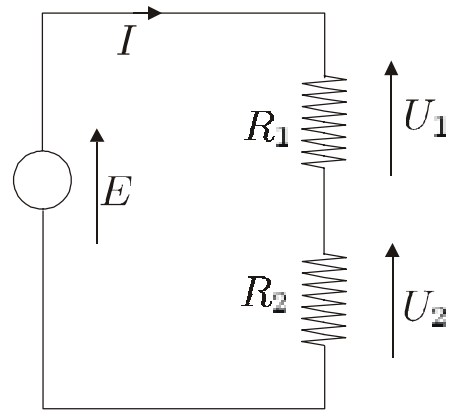
$$x + \sin(y) - xz \leq 0,$$

$$x \in [-1, 1], y \in [-1, 1], z \in [-1, 1]$$

se décompose en

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

Exemple (estimation à erreurs bornées)



Contraintes :

$$P = EI; E = (R_1 + R_2) I;$$

$$U_1 = R_1 I; U_2 = R_2 I; E = U_1 + U_2.$$

Domaines initiaux

$$R_1 \in] - \infty, \infty[, R_2 \in] - \infty, \infty[,$$

$$E \in [23, 26], I \in [4, 8], U_1 \in [10, 11],$$

$$U_2 \in [14, 17], P \in [124, 130],$$

Domaines contractés

$$R_1 \in [1.84, 2.31], R_2 \in [2.58, 3.35],$$

$$I \in [4.77, 5.42], U_1 \in [10; 11], U_2 \in [14; 16],$$

$$E \in [24; 26], P \in [124, 130].$$

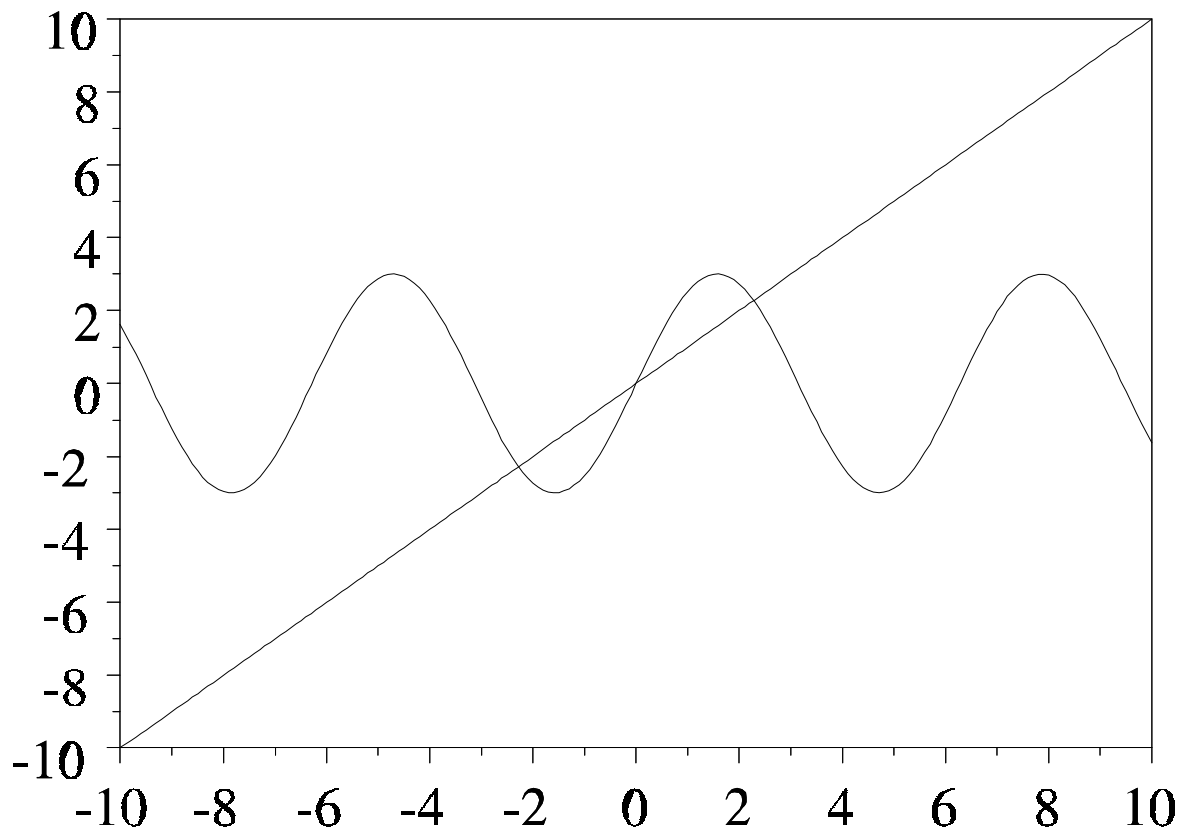
Résolution d'équations nonlinéaires

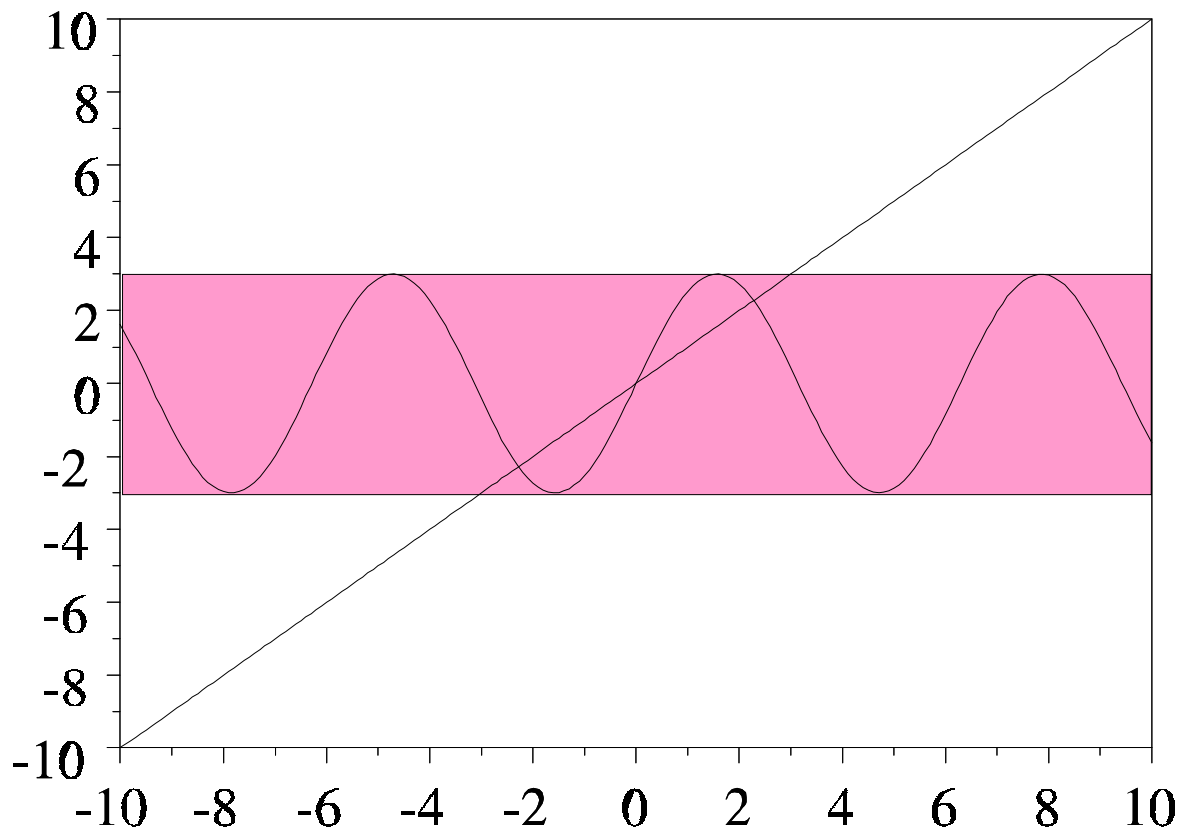
Cherchons par exemple à résoudre

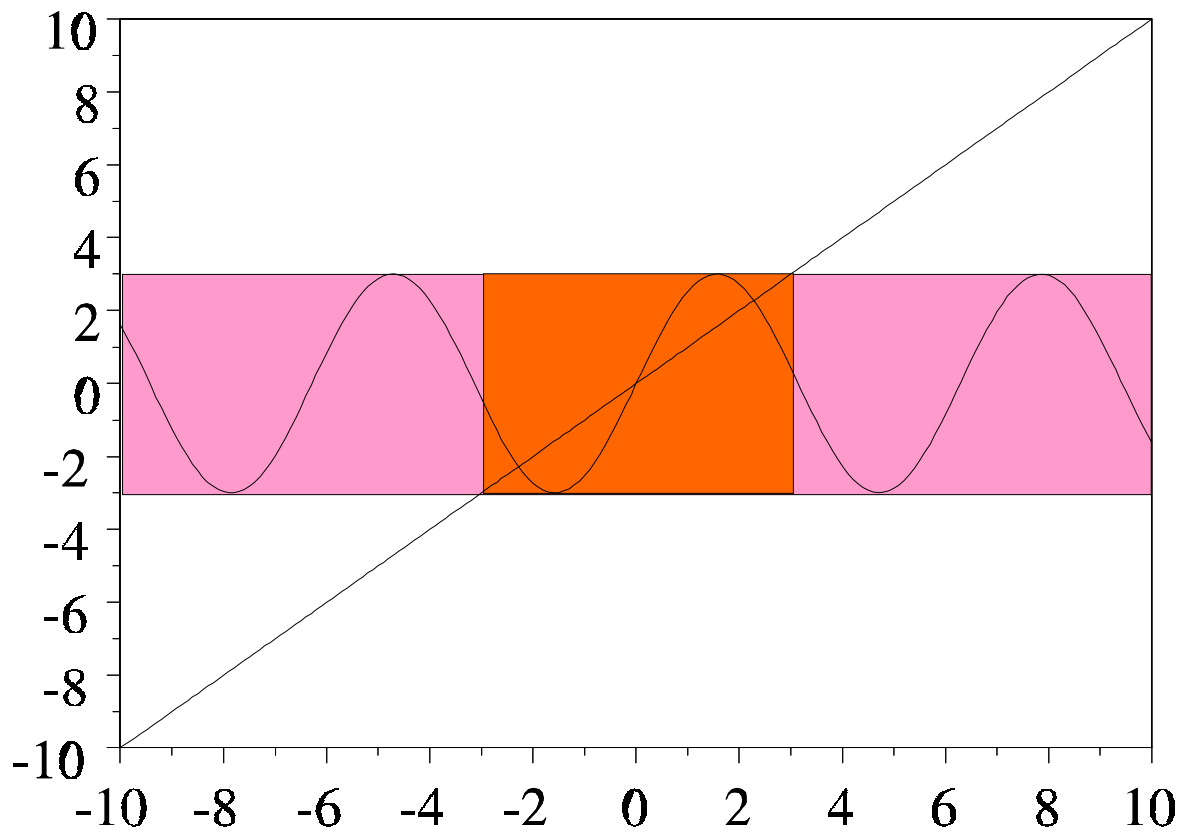
$$3 \sin(x) = x, \quad x \in \mathbb{R}.$$

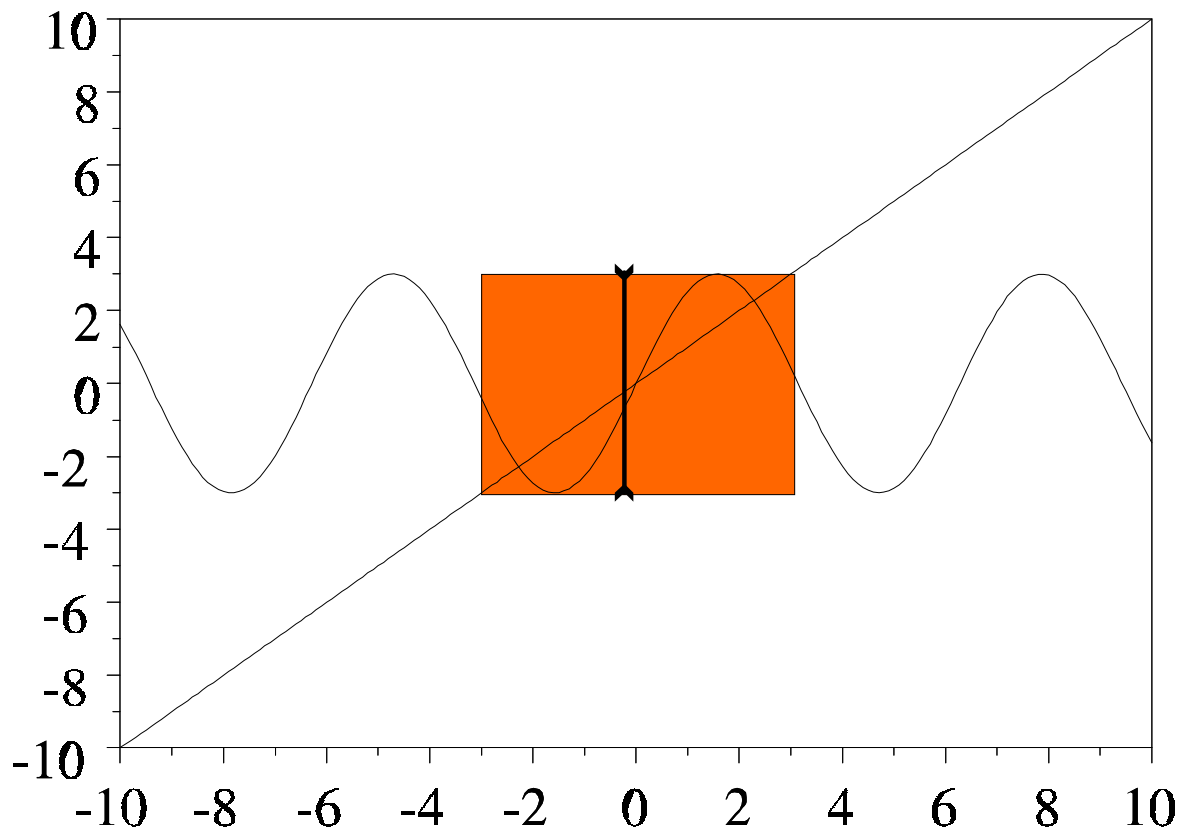
Cette équation se décompose en

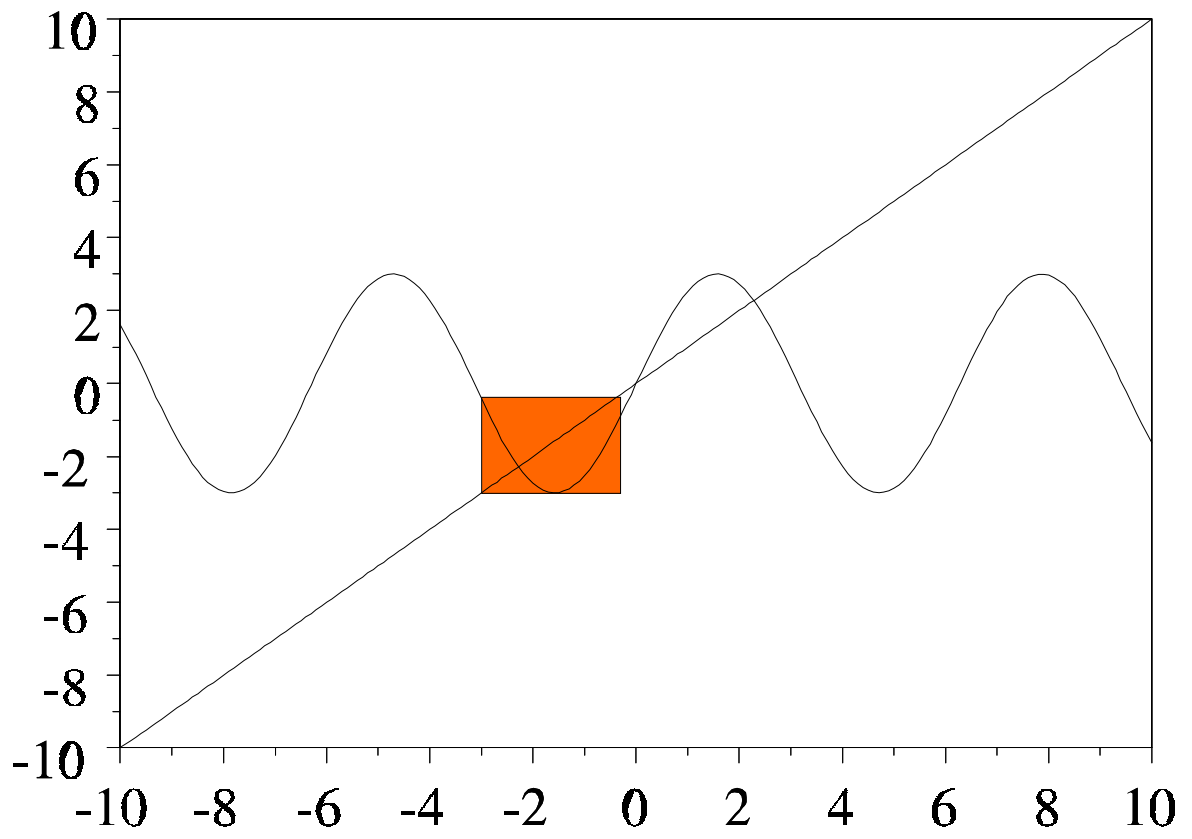
$$\begin{cases} y = 3 \sin(x) \\ y = x \end{cases} \quad x \in \mathbb{R}, y \in \mathbb{R}.$$

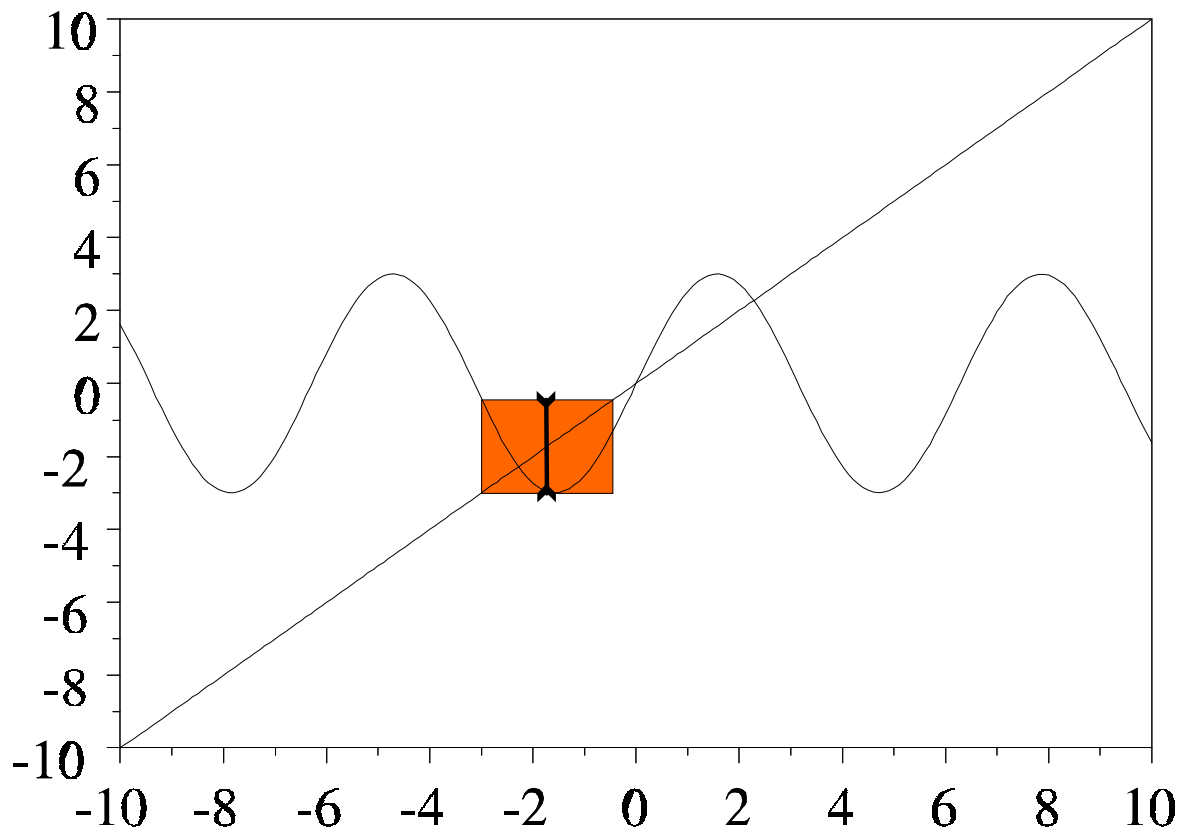


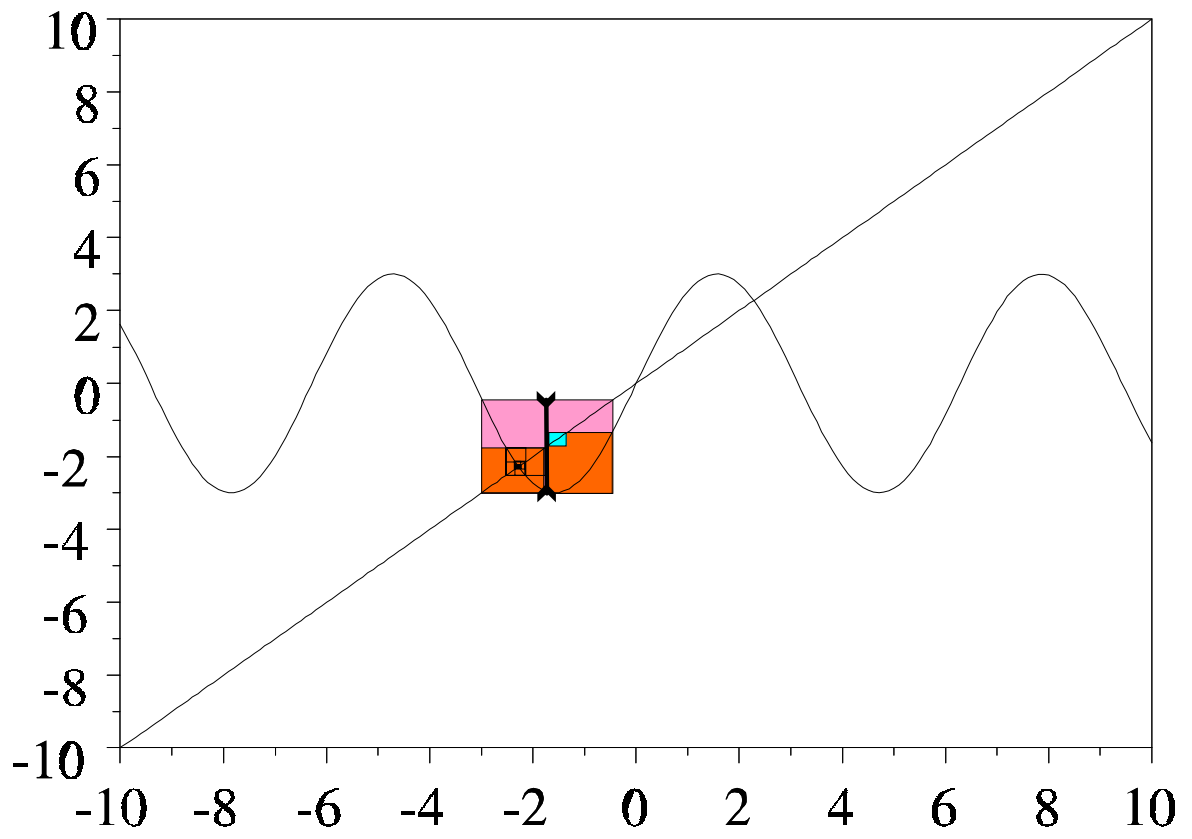












Prouver qu'un ensemble de contraintes
est toujours satisfait
(pour montrer la stabilité robuste, par exemple)

Idée : On complémente l'ensemble des contraintes.

Par exemple, montrer que

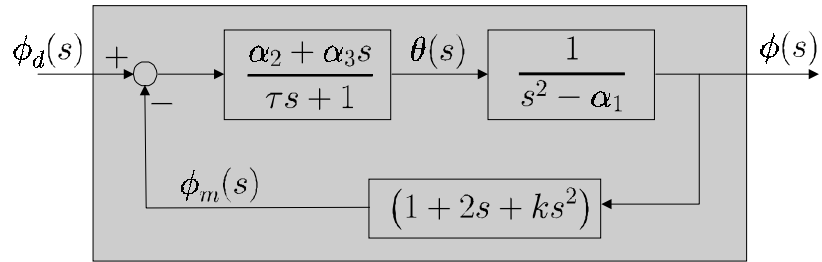
$$\forall x \in [x], \forall y \in [y], f(x, y) \leq 0 \text{ et } g(x, y) \leq 0,$$

revient à montrer que

$$\{(x, y) \in [x] \times [y] \mid f(x, y) > 0 \text{ ou } g(x, y) > 0\} = \emptyset$$

c'est-à-dire que

$$\{(x, y) \in [x] \times [y] \mid \max(f(x, y), g(x, y)) > 0\} = \emptyset.$$



$$\alpha_1 \in [8.8; 9.2], \alpha_2 \in [2.8; 3.2], \alpha_3 \in [0.8; 1.2],$$

$$\tau \in [1.8; 2.2], k \in [-3.2; -2.8]$$

Polynôme caractéristique : $a_3 s^3 + a_2 s^2 + a_1 s + a_0$

avec

$$a_3 = \tau + \alpha_3 k, \quad a_2 = \alpha_2 k + 2\alpha_3 + 1,$$

$$a_1 = \alpha_3 - \alpha_1 \tau + 2\alpha_2, \quad a_0 = -\alpha_1 + \alpha_2$$

La table de Routh :

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

Ainsi, le système est stable si $a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}$ et a_0 sont de même signe.

On a la stabilité robuste si

$\forall (\alpha_1 \in [\alpha_1], \alpha_2 \in [\alpha_2], \alpha_3 \in [\alpha_3], \tau \in [\tau], k \in [k]),$
 $a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}$ et a_0 sont de même signe.

c'est-à-dire

$\exists (\alpha_1 \in [\alpha_1], \alpha_2 \in [\alpha_2], \alpha_3 \in [\alpha_3], \tau \in [\tau], k \in [k]),$

$$\max \left(\min \begin{pmatrix} a_3 \\ a_2 \\ \frac{a_2 a_1 - a_3 a_0}{a_2} \\ a_0 \end{pmatrix}, - \max \begin{pmatrix} a_3 \\ a_2 \\ \frac{a_2 a_1 - a_3 a_0}{a_2} \\ a_0 \end{pmatrix} \right) \leq 0$$

est fausse.

Prouver la stabilité robuste revient donc à montrer que le CSP

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

possède un ensemble solution vide. La preuve est obtenue en moins de 10^{-3} seconde par RealPaver.

Variables

a0 in]-oo,+oo[,
a1 in]-oo,+oo[,
a2 in]-oo,+oo[,
a3 in]-oo,+oo[,
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],
\$b in] -oo,+oo[•

Constraints

a3 = tau + alpha3*k,
a2 = alpha2*k + 2*alpha3 + 1,
a1 = alpha3 - alpha1*tau + 2*alpha2,
a0 = alpha2 - alpha1,
b = (a2*a1 - a3*a0)/a2,
0 >= max(min(a3,min(a2,min(b,a0))),
-max(a3,max(a2,max(b,a0)))) •

Calibration du robot Säubli rx90

$\mathbf{q} = (\theta_1, \dots, \theta_6) \in \mathbb{R}^6$ coordonnées articulaires (mesuré).

$\mathbf{p} = (\alpha_0, d_0, r_0, \dots, \alpha_7, d_7, r_7, \theta_0, \theta_8) \in \mathbb{R}^{26}$
paramètres géométriques (à calibrer).

$\mathbf{x} = (x, y, z) \in \mathbb{R}^3$ coordonnées de l'organe terminal (mesuré).

On a :

$$\mathbf{x} = \mathbf{f}(\mathbf{p}, \mathbf{q})$$

Avec \mathbf{f} : fonction donnant le modèle géométrique direct (Khalil Kleinfinger)

A chaque corps du robot est associé un repère $(\mathbf{i}, \mathbf{j}, \mathbf{k})$.

On définit la matrice de changement de repère, entre les corps liés :

$${}^2T_3 = Rot(\mathbf{i}_2, \alpha_3) Trans(\mathbf{i}_2, d_3) Rot(\mathbf{k}_2, \theta_3) Trans(\mathbf{k}_2, r_3)$$

Et celle entre les corps 0 à 7 :

$${}^0T_7 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6 \cdot {}^6T_7$$

D'où la contrainte matricielle :

$$\mathbf{x} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6 \cdot {}^6T_7 \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Méthode

- 1) Positionner le robot dans différentes configurations $\mathbf{q}(1), \dots, \mathbf{q}(50)$
- 2) Mesurer les positions $\mathbf{x}(1), \dots, \mathbf{x}(50)$
- 3) Générer les contraintes : $\mathbf{x}(k) = \mathbf{f}(\mathbf{p}, \mathbf{q}(k)), k \leq 50$
- 4) Propagation.

j	α_j	d_j	θ_j	r_j
0	0	0	$\pi/2$	0.5
1	0.1	0	θ_1	0
2	$-\pi/2$	0	θ_2	0
3	0	0.5	θ_3	0
4	$\pi/2$	0	θ_4	0.5
5	$-\pi/2$	0	θ_5	0
6	$\pi/2$	0	θ_6	0.3
7	1.3	0.2	$\pi/2$	0.1

Ce problème est un CSP (constraint satisfaction problem) contenant 8526 variables et 8200 contraintes.

Résultats

	Domaines initiaux	20 points	50 points	Valeurs nominales
r_0	[0, 10]	[0.28, 0.726]	[0.497, 0.503]	0.5
d_3	[0, 10]	[0.25, 0.733]	[0.4970, 0.503]	0.5
r_4	[0, 10]	[0.33, 0.645]	[0.498, 0.502]	0.5
r_6	[0, 10]	[0.21, 0.401]	[0.299, 0.301]	0.3
d_7	[0, 10]	[0.135, 0.26]	[0.199, 0.201]	0.2
r_7	[0, 10]	[0.04, 0.153]	[0.099, 0.101]	0.1
	temps	12.3s	12.4s	

Résultats sur un pentium 2 avec 64mo de ram

Localisation d'un satellite

$$\left\{ \begin{array}{l} x_1(k+1) = x_1(k) + 0.5 x_3(k) + w_1(k) \\ x_2(k+1) = x_2(k) + 0.5 x_4(k) + w_2(k) \\ x_3(k+1) = x_3(k) - 0.5 \frac{x_1(k)}{\sqrt{x_1^2(k)+x_2^2(k)}} + w_3(k) \\ x_4(k+1) = x_4(k) - 0.5 \frac{x_2(k)}{\sqrt{x_1^2(k)+x_2^2(k)}} + w_4(k) \\ y_1(k) = x_2(k)/x_1(k) \\ y_2(k) = x_1(k)x_3(k) + x_2(k)x_4(k) \\ y_3(k) = x_1^2(k) + x_2^2(k) \end{array} \right.$$

Fichier en entrée dans le solveur (Realpaver)

```
x1_0 in [-oo,oo],  
x2_0 in [-oo,oo],  
x3_0 in [-oo,oo],  
x4_0 in [-oo,oo],  
w1_0 in [-0.01,0.01],  
w2_0 in [-0.01,0.01],  
w3_0 in [-0.01,0.01],  
w4_0 in [-0.01,0.01],  
y1_0 in [0.107,0.307],  
y2_0 in [-0.710,-0.510],  
y3_0 in [16.117,16.317],  
...
```

Constraints

```
x1_1=x1_0+0.5*x3_0+ w1_0,  
x2_1=x2_0+0.5*x4_0+ w2_0,  
x3_1=x3_0-0.5*x1_0/  
      (sqrt(x1_0^2+x2_0^2))^3+w3_0,  
...
```

Fichier en sortie du solveur

x1_0 in [3.89 , 4.01]

x2_0 in [0.41 , 0.95]

x3_0 in [-0.41 , -0.14]

x4_0 in [0.20 , 0.93]

y1_0 in [0.10 , 0.25]

y2_0 in [-0.71 , -0.51]

y3_0 in [16.11 , 16.31]

...

x1_200 in [4.38 , 4.88]

x2_200 in [-10.84 , -10.61]

x3_200 in [-0.14 , 0.54]

x4_200 in [-0.00 , 0.32]

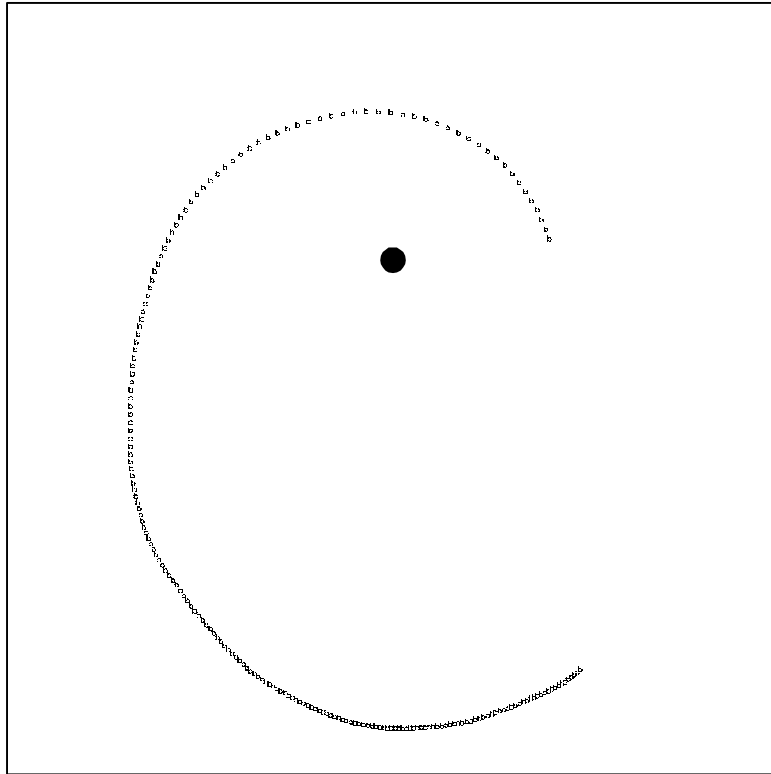
y1_200 in [-2.42 , -2.22]

y2_200 in [-0.92 , -0.72]

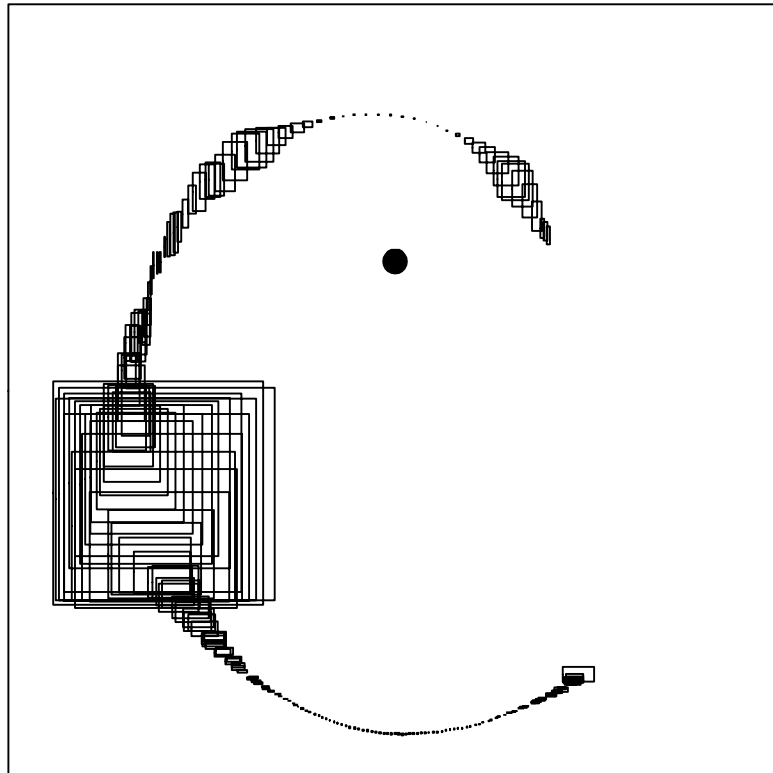
y3_200 in [136.61 , 136.81]

...

Elapsed time: 810 ms

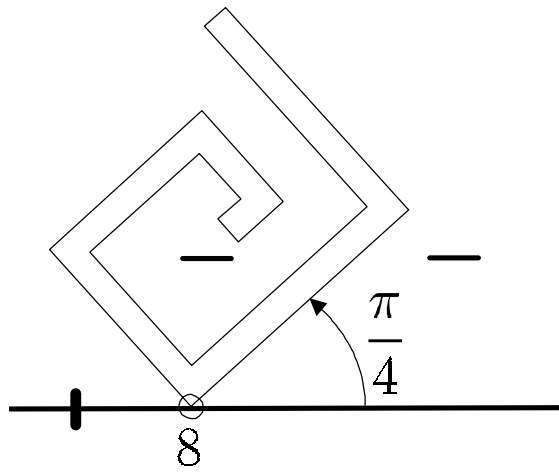


Positions du satellite générées par simulation
pour k allant de 0 à 200

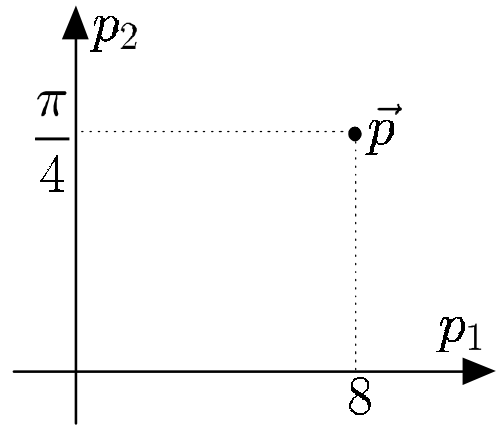


Pavés générés par Realpaver

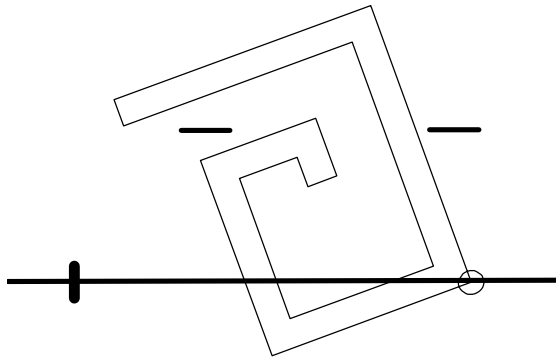
Ce problème est un CSP (constraint satisfaction problem) contenant 8815 variables et 5607 contraintes.

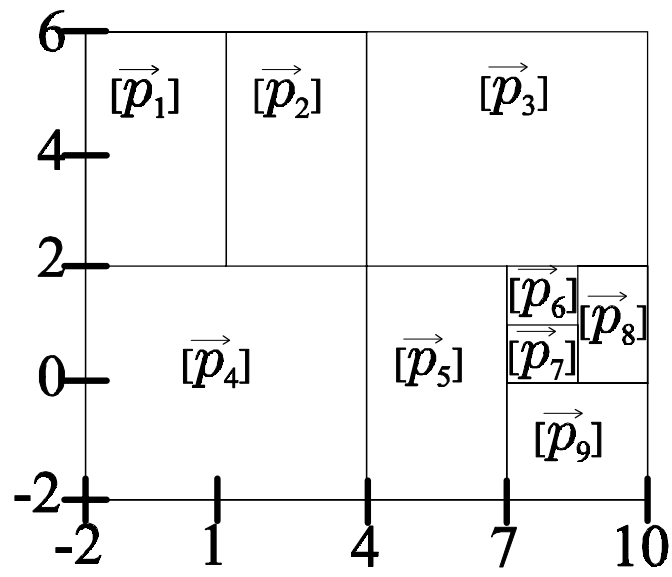


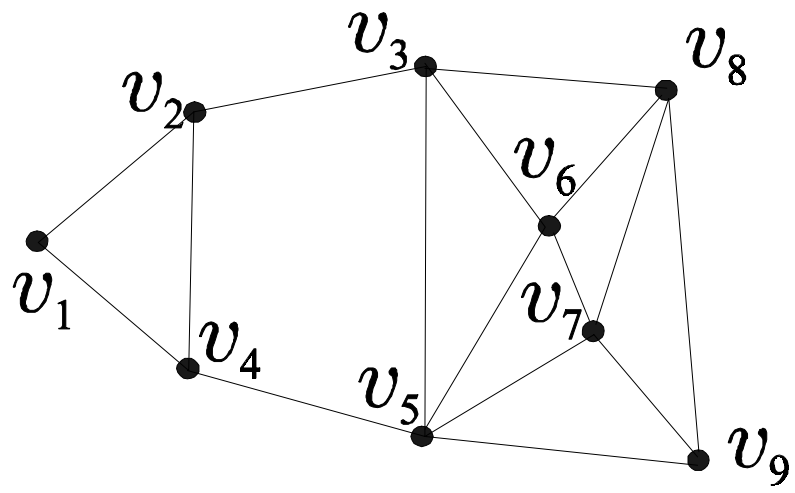
Room



Configuration space







FIN

Ce qui suit présente la partie faite au tableau

Méthode de calcul des trois projections de \mathbb{S} suivant x, y
et z

$$\begin{aligned} z = x + y \Rightarrow z \in [6, 10] \cap ([1, 5] + [2, 4]) \\ = [6, 10] \cap [3, 9] = [6, 9] \end{aligned}$$

$$\begin{aligned} x = z - y \Rightarrow x \in [1, 5] \cap ([6, 10] - [2, 4]) \\ = [1, 5] \cap [2, 8] = [2, 5] \end{aligned}$$

$$\begin{aligned} y = z - x \Rightarrow y \in [2, 4] \cap ([6, 10] - [1, 5]) \\ = [2, 4] \cap ([1, 9] - [1, 5]) = [2, 4] \end{aligned}$$

Calcul pour démontrer l'inconsistance du système des 3 équations à deux inconnues

$$(C_1) \Rightarrow y \in] - \infty, \infty[^2 = [0, \infty[$$

$$(C_2) \Rightarrow x \in 1/[0, \infty[= [0, \infty[$$

$$\begin{aligned}(C_3) \Rightarrow y &\in [0, \infty[\cap ((-2) \cdot [0, \infty[+ 1) \\ &= [0, \infty[\cap (] - \infty, 1]) = [0, 1] \\ x &\in [0, \infty[\cap (-[0, 1]/2 + 1/2) \\ &= [0, 1/2]\end{aligned}$$

$$(C_1) \Rightarrow y \in [0, 1] \cap [0, 1/2]^2 = [0, 1/4]$$

$$(C_2) \Rightarrow x \in [0, 1/2] \cap 1/[0, 1/4] = \emptyset$$

$$y \in [0, 1/4] \cap 1/\emptyset = \emptyset$$