Inner and outer characterization of the projection of polynomial equations using symmetries, quotients and intervals

Luc Jaulin

Lab-Sticc, ENSTA-Bretagne

Abstract

In this paper, we propose a new approach to compute the projection of a set defined by polynomial equations. It assumes that the polynomial equations have some nice symmetries and that a solution of the projection problem is already available in the case where the variables along which we project are all positive. A new interval-based algorithm which combines symmetry operators and set quotient is proposed. Symmetries are used to move from one part of the space to another. The set quotient is needed to avoid redundant symmetries. The projection procedure yields an inner and an outer approximations of the projected set. Two applications are considered. The first one corresponds to the characterization of the space occupied by a rotating polygon, and the second one deals with the estimation of the speed of a moving object observed by several robots with uncertain orientations.

1 Introduction

In this paper, we propose an original method which combines symmetries, set quotient and interval analysis [20] to compute an inner and an outer approximation of the set $\mathbb{P}$ which corresponds to the projection of another set $\mathbb{X}$ defined by polynomial equations. We will assume a solution of the projection problem is already available in the case where the variables along which we project are all positive. Here symmetries have to be understood as transformations from $\mathbb{R}^n$ to $\mathbb{R}^n$ such as axial symmetries or rotations.

The set to be projected has the form

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{f}(\mathbf{x}) = \mathbf{0}\} \quad (1)$$

where $\mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^\ell$ is a vector of polynomials. We want to characterize the projected set defined by

$$\mathbb{P}^{[\mathbf{q}]} = \{\mathbf{p} \in \mathbb{R}^m | \exists \mathbf{q} \in [\mathbf{q}], \mathbf{f}(\mathbf{p}, \mathbf{q}) = \mathbf{0}\} \quad (2)$$

where $\mathbf{x} = (\mathbf{p}, \mathbf{q})$ and $[\mathbf{q}]$ is a box of $\mathbb{R}^\ell$.

This problem has already been considered and solved using interval analysis in different fields such as in control theory [24], geodesy [26], state estimation [22] or robotics [10][25]. The idea is to benefit from a projection formalization so that we can focus on the variables of interest $p_1, p_2, \ldots$ without spending time to estimate variables $q_1, q_2, \ldots$ that we do not care of.

Due to the Tarski–Seidenberg theorem [19], we know that the set $\mathbb{P}^{[\mathbf{q}]}$ is semi-algebraic, *i.e.*, it can be defined by polynomial inequalities. The elimination can be performed symbolically [3]. For instance, if

$$\mathbb{P} = \left\{ \mathbf{p} \in \mathbb{R}^2 | \exists q \in [q], q^2 + p_1 q + p_2 = 0 \right\}, \tag{3}$$

then the quantifier elimination yields

$$\mathbb{P} = \left\{ \mathbf{p} \in \mathbb{R}^2 | p_1^2 - 4p_2 \geq 0 \right\}. \tag{4}$$

Although the original proof of the Tarski–Seidenberg theorem was constructive, the resulting algorithm has a computational complexity that is too high for using the method on a computer in a reasonable time. An effective symbolic algorithm, called *cylindrical algebraic decomposition* (CAD) has been proposed by Collins in 1975 [4][7] for the quantifier elimination. The principle is to decompose $\mathbb{R}^n$ into connected semi-algebraic sets called cells, on which each polynomial has constant sign. Unfortunately, Collins' algorithm has a computational complexity that is double exponential in n.

Searching to improve Collins' algorithm, or to provide algorithms that have a better complexity for subproblems of general interest, is an active field of research [3]. Interval methods have been combined with CAD to reduce the complexity of CAD [23]. In the specific case, where the quantifier to be eliminated is $\exists$, it is possible to build dedicated interval algorithms [11] [14] [27].

In this paper, we will consider the set of transformations $\boldsymbol{\varphi}$ of $\mathbb{R}^n$ that leave the set $\mathbb{X}$ invariant, *i.e.*, $\boldsymbol{\varphi}(\mathbb{X}) = \boldsymbol{\varphi}^{-1}(\mathbb{X}) = \mathbb{X}$. Such a transformation will be called a *symmetry* of $\mathbb{X}$. It can be shown that this set of symmetries is a group which is called the *symmetry group* of $\mathbb{X}$ [18]. We propose to take advantage of the symmetries to perform the quantifier elimination.

Our approach requires some specific properties on $\mathbf{f}$. The principle of the approach is to solve the projection problem assuming that the constraints are monotonic. Then using symmetries, we show that, for some specific cases we might be able to relax step by step all assumptions on the monotonicity to solve the problem for all conditions.

The paper is organized as follows. Section 2 recalls the mathematical notion that will be used in the paper and introduces the notion of the quotient of a symmetry group with respect to an equivalence relation in the context of polynomial projection. Section 3 provides the main theorem from which a projection algorithm is derived. Section 4 provides two applications: the first one is related to the computation of the workspace occupied by a rotating object and the second one is concerned with the estimation of the speed of an object observed by several robots with an uncertain orientation. Section 5 concludes the paper.

2 Theory

In this section, we will present the mathematical tools that are needed to present the projection algorithm. The principle is to decompose the $\mathbf{q}$-space into 2^m quadrants. If we assume that the projection problem is solved in one of the quadrant $[\mathbf{a}]$, then we will use symmetries as operators to move from the quadrant of interest to $[\mathbf{a}]$.

2.1 Introductory example

We propose a small example we call the *two circles example*, that will be referenced later in the paper to illustrate some notations and concepts.

Consider the set $\mathbb{X}$ defined by the equation:

$$f(x_1, x_2) = 0 \tag{5}$$

where

$$f(x_1, x_2) = x_1^4 + x_2^4 + 2x_1^2x_2^2 - 48x_1x_2 - 12x_2^2 + 8x_1^2 + 144 \tag{6}$$

We set

$$\mathbf{x} = (x_1, x_2) = (p, q). \tag{7}$$

Given the interval $[q] \subseteq \mathbb{R}$, the projected set, as defined by (2), is here

$$\mathbb{P}^{[q]} = \{p \in \mathbb{R} | \exists q \in [q], f(p, q) = 0\}, \tag{8}$$

as illustrated by Figure 1.

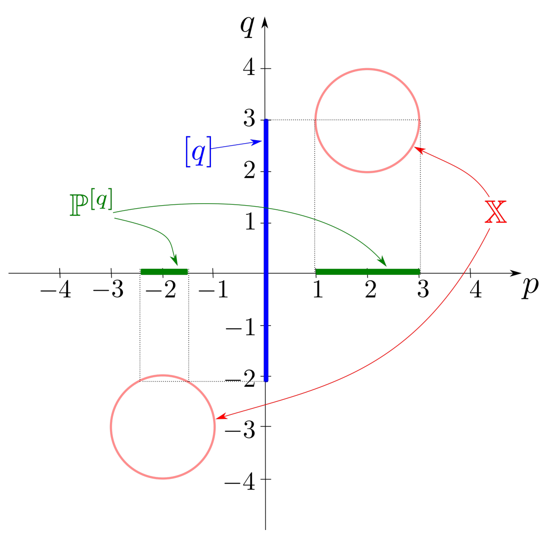


Fig. 1: The union of the two circles composes the set $\mathbb{X}$

We observe that for this example, the set $\mathbb{X}$ has a central symmetry with respect to $\mathbf{0}$. Now, since $[q]$ is not symmetric, the projection $\mathbb{P}^{[q]}$ has no reason to inherit the symmetry of $\mathbb{X}$.

2.2 Action of a symmetry on a set

Consider a symmetry σ of $\mathbb{R}^n$. We define the *action* [8] of σ on the set $\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{f}(\mathbf{x}) = \mathbf{0}\}$ as

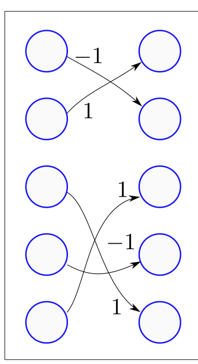
$$\begin{array}{rcl} \sigma\mathbb{X} & = & \{\mathbf{y} \,|\, \exists \mathbf{x}, \mathbf{y} = \sigma(\mathbf{x}) | \mathbf{f}(\mathbf{x}) = \mathbf{0}\} \\ & = & \left\{\mathbf{y} \,|\, \exists \mathbf{x}, \mathbf{x} = \sigma^{-1}(\mathbf{y}) | \mathbf{f}(\mathbf{x}) = \mathbf{0}\right\} \\ & = & \left\{\mathbf{y} \,|\, \mathbf{f}(\sigma^{-1}(\mathbf{y})) = \mathbf{0}\right\}. \end{array} \tag{9}$$

The set $\mathbb{X}$ is symmetric with respect to σ if $\sigma\mathbb{X} = \mathbb{X}$. As we will see later in the paper, the symmetries will be used as an operator to build complex sets from simple ones.

2.3 Hyperoctahedral group

The symmetries that will be used in this paper are limited to the *hyperoctahedral group* B_n [8] which is the group of symmetries of the hypercube $[-1, 1]^n$ of $\mathbb{R}^n$. We could also include other symmetries in our approach such as translations, or scaling, as long as it transforms a box into a box. Other types of symmetries could be included for other abstract domains such as octagons [21]. The group B_n corresponds to the group of $n \times n$ orthogonal matrices whose entries are integers. The group B_n contains $2^n \cdot n!$ elements. For instance, for $n = 2$, we have $2^2 \cdot 2! = 8$ elements. Each line and each column of a matrix should contain one and only one non zero entry which should be either 1 or -1. Figure 2 shows different notations usually considered to represent a symmetry σ of B_5. We will prefer the Cauchy one line notation [28] which is shorter. We should understand the symmetry σ of the figure as the function:

$$\sigma(x_1, x_2, x_3, x_4, x_5) = (-x_2, x_1, x_5, -x_4, x_3). \tag{10}$$



$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} -2 & 1 & 5 & -4 & 3 \end{pmatrix}$$

Fig. 2: Different representations of an element σ of B_5. Left: graph; Top right: Matrix notation; Bottom right: Cauchy one line notation

The subset of all symmetries preserving the set $\mathbb{X}$ is denoted by $B_n(\mathbb{X})$.

For our *two-disks* example, we have

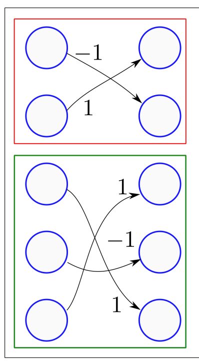
$$B_2(\mathbb{X}) = \{(1, 2), (-1, -2)\}. \tag{11}$$

Even if the matrix representation looks more intuitive, for efficiency reasons, we use the Cauchy one line representation to compose the symmetries. For instance, the multiplication of two vectors $\mathbf{u}, \mathbf{v}$ of B_n is given by

$$\mathbf{u} \cdot \mathbf{v} = (\mathrm{sign}(v_1) \cdot u_{|v_1|}, \ldots, \mathrm{sign}(v_n) \cdot u_{|v_n|}). \tag{12}$$

2.4 Separability

Consider the set $\mathbb{R}^n = \mathbb{R}^m \times \mathbb{R}^\ell$ containing elements $\mathbf{x} = (\mathbf{p}, \mathbf{q}) \in \mathbb{R}^n$. A symmetry $\sigma \in B_n$ is *separable* if $\sigma = (\sigma_\mathbf{p}, \sigma_\mathbf{q})$, where $\sigma_\mathbf{p}, \sigma_\mathbf{q}$ are symmetries in the $\mathbf{p}$-space and the $\mathbf{q}$-space respectively. Equivalently, the corresponding matrix is block diagonal. The set of separable symmetries of B_n is denoted by $sep(B_n)$. Is is easy to check that $sep(B_n)$ is a subgroup of B_n. The notion of separability is needed in this paper since we want to project the set $\mathbb{X} \subseteq \mathbb{R}^n$ onto the $\mathbf{p}$-space along the $\mathbf{q}$-space. This notion is illustrated by Figure 3.



$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} -2 & 1 & 5 & -4 & 3 \end{pmatrix}$$

Fig. 3: The symmetry is separable since $\sigma_\mathbf{p}$ (red) do not interact with $\sigma_\mathbf{q}$ (green)

2.5 Quadrants

The $\mathbf{q}$-space can be decomposed into 2^ℓ *quadrants* that may be seen as the cell decomposition needed by our projection algorithm.

Define the set of signs: $\mathcal{E} = \{-, +\}$ equipped with the relation order $\leq$ with $- \leq +$. A *sign vector* is a vector of signs. The set of all sign vectors of dimension ℓ is $\mathcal{E}^\ell = \{-, +\}^\ell$. By convention, the sign of 0 will be taken as $+$. For each vector $\mathbf{q}$, we define the vector sign $sgn(\mathbf{q})$ componentwise. For each element ε of $\mathcal{E}^\ell$, we can associate a quadrant

$$\mathbb{R}^\varepsilon = sgn^{-1}(\varepsilon). \tag{13}$$

For instance the positive quadrant of $\mathbb{R}^2$ is $[\mathbf{a}] = [0,\infty)^2$. We can define an order relation for $\mathcal{E}^\ell$ componentwise. For instance in $\mathcal{E}^4$ we have

$$(-+-+) \leq (++-+).$$

We can also define the order relation for the set of quadrants induced by sgn:

$$\mathbb{R}^{\varepsilon_1} \leq \mathbb{R}^{\varepsilon_2} \Leftrightarrow \varepsilon_1 \leq \varepsilon_2.$$

The two partially ordered sets (the set of quadrants and $\mathcal{E}^\ell$) are order isomorphic and sgn is an order isomorphism [2]. Thus, we will make no distinction between the sign vector and the quadrant. For instance, we will write $[\mathbf{a}] = [0,\infty)^2 = (++)$. Since $(\mathcal{E}^\ell, \leq)$ has a lattice structure, we can define intervals of $\mathcal{E}^\ell$. The notion of a quadrant is illustrated by Figure 4. The interval sign vector in magenta is $[-,+] \times [+,+]$. It corresponds to the union of the two quadrants yellow + green.

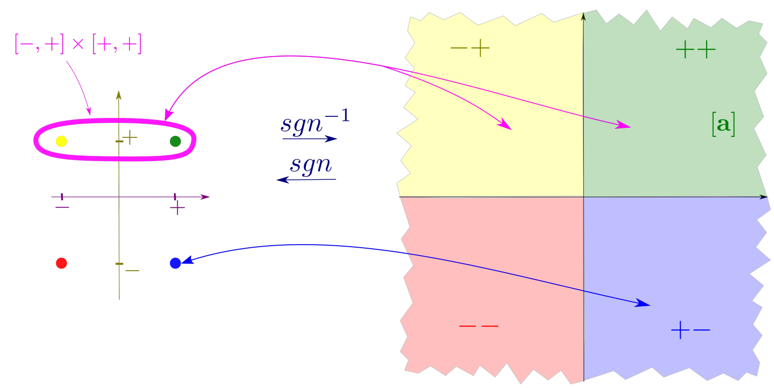


Fig. 4: Equivalence between the quadrants of $\mathbb{R}^2$ and the set of sign vectors $\mathcal{E}^2$

Consider a box $[\mathbf{q}]$. We define the interval extension of the sign vector as

$$sgn([\mathbf{q}]) = \left\{\varepsilon \in \mathcal{E}^\ell | \exists \mathbf{q} \in [\mathbf{q}], \varepsilon = sgn(\mathbf{q})\right\}. \tag{14}$$

Note that $sgn([\mathbf{q}])$ is an interval of $\mathcal{E}^\ell$. For instance, if $[\mathbf{q}] = [-1,2] \times [1,2] \times [-2,2]$, we have

$$\begin{array}{rcl} sgn([\mathbf{q}]) & = & [-,+] \times [+,+] \times [-,+] \\ & = & \{(-+-),(-++),(++-),(+++])\} \\ & = & \left\{ \begin{pmatrix} (-\infty,0] \\ [0,\infty) \\ (-\infty,0] \end{pmatrix}, \begin{pmatrix} (-\infty,0] \\ [0,\infty) \\ [0,\infty) \end{pmatrix}, \begin{pmatrix} [0,\infty) \\ [0,\infty) \\ (-\infty,0] \end{pmatrix}, \begin{pmatrix} [0,\infty) \\ [0,\infty) \\ [0,\infty) \end{pmatrix} \right\} \end{array} \tag{15}$$

2.6 Quotient

The group $sep(B_n)$ allows us to move from one quadrant to another. Now, many elements of $sep(B_n)$ are equivalent. They transport the projection problem from one quadrant to the same quadrant. The goal of the quotient operator to be defined now is to select a minimal number of representatives to avoid redundant symmetries.

Define the function $\varphi(\sigma_{\mathbf{p}}, \sigma_{\mathbf{q}}) = \mathrm{sgn}(\sigma_{\mathbf{q}})$. For instance, for $m = 2$, we have

$$\varphi(2, 1, -4, 3, 6, -5) = \mathrm{sgn}(-4, 3, 6, -5) = (- + + -). \tag{16}$$

We define the equivalence relation $\sim$ as

$$\sigma_1 \sim \sigma_2 \Leftrightarrow \varphi(\sigma_1) = \varphi(\sigma_2) \tag{17}$$

and the quotient set a

$$\mathcal{Q} = \frac{sep(B_n(\mathbb{X}))}{\sim}. \tag{18}$$

In practice, the quotient set $\mathcal{Q}$ gives us the symmetries which allows us to move from one quadrant of the $\mathbf{q}$ space to the positive quadrant. Most of the time, for a given $\varepsilon \in \mathcal{E}^{\ell}$, $\varphi^{-1}(\varepsilon)$ is not unique, as illustrated by Figure 5. Therefore, we take one of them, called the *representative,* and denoted by $\psi(\varepsilon)$. The function ψ is a *choice function* as defined [29]. Note that $\mathcal{Q}$ is a quotient set and not a quotient group. This is due to the fact that φ is not a group isomorphism (neither the set $\mathcal{E}^{\ell}$ is a group). Now, this stronger algebraic structure will not be needed further and only the quotient set will be used.

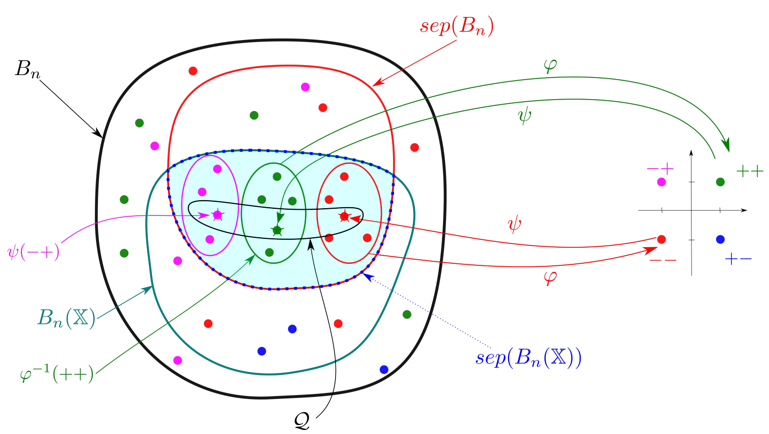


Fig. 5: Construction of the choice function ψ. The elements $\psi(\varepsilon)$, $\varepsilon \in \{(++), (+-), (--)\}$ are represented by the stars

For our two-disks example, we have

$$\begin{array}{lcc}
B_2 & = & \{(-1,-2),(-1,2),(1,-2),(1,2),(-2,-1),(-2,1),(2,-1),(2,1)\} \\
sep(B_2) & = & \{(-1,-2),(-1,2),(1,-2),(1,2)\} \\
sep(B_2(\mathbb{X})) & = & \{(-1,-2),(1,2)\} \\
\mathcal{Q} & = & \{(-1,-2),(1,2)\} \\
\psi(-) & = & (-1,-2) \\
\psi(+) & = & (1,2)
\end{array}$$

We are in an simple situation where $\mathcal{Q} = sep(B_2(\mathbb{X}))$, *i.e.*, φ is injective.

3 Method

This section proposes a theorem and an algorithm that will be used for computing the projection of set defined by (1), *i.e.*,

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{f}(\mathbf{x}) = \mathbf{0}\} \tag{19}$$

where $\mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^\ell$. More precisely, we want to characterize the projected set (2) :

$$\mathbb{P}^{[\mathbf{q}]} = \{\mathbf{p} \in \mathbb{R}^m | \exists \mathbf{q} \in [\mathbf{q}], \mathbf{f}(\mathbf{p}, \mathbf{q}) = \mathbf{0}\} \tag{20}$$

where $\mathbf{x} = (\mathbf{p}, \mathbf{q})$ and $[\mathbf{q}]$ is a box of $\mathbb{R}^\ell$.

3.1 Decomposition theorem

The decomposition theorem will allow us to build the projected set using simpler sets transported by symmetries. It it based on the following proposition.

Proposition 1. *If* $\sigma = \sigma_{\mathbf{p}} \times \sigma_{\mathbf{q}} \in sep(B_n(\mathbb{X}))$, *we have*

$$\mathbb{P}^{[\mathbf{q}]} = \sigma_{\mathbf{p}} \mathbb{P}^{\sigma_{\mathbf{q}}[\mathbf{q}]}. \tag{21}$$

Proof. We have

$$\begin{array}{rcl}
\sigma_{\mathbf{p}} \mathbb{P}^{\sigma_{\mathbf{q}}[\mathbf{q}]} & = & \sigma_{\mathbf{p}} \{\mathbf{p}\, | \exists \mathbf{q} \in \sigma_{\mathbf{q}}[\mathbf{q}], \mathbf{f}(\mathbf{p}, \mathbf{q}) = \mathbf{0}\} \\
& = & \sigma_{\mathbf{p}} \left\{\mathbf{p}\, | \exists \mathbf{q} \in \sigma_{\mathbf{q}}[\mathbf{q}], \mathbf{f}(\sigma_{\mathbf{p}}^{-1}\mathbf{p}, \sigma_{\mathbf{q}}^{-1}\mathbf{q}) = \mathbf{0}\right\} \\
& = & \sigma_{\mathbf{p}} \left\{\mathbf{p}\, | \exists \mathbf{q}' \in [\mathbf{q}], \mathbf{f}(\sigma_{\mathbf{p}}^{-1}\mathbf{p}, \mathbf{q}') = \mathbf{0}\right\} \\
& = & \left\{\sigma_{\mathbf{p}}\mathbf{p} | \exists \mathbf{q}' \in [\mathbf{q}], \mathbf{f}(\sigma_{\mathbf{p}}^{-1}\mathbf{p}, \mathbf{q}') = \mathbf{0}\right\} \\
& = & \{\mathbf{p}' | \exists \mathbf{q}' \in [\mathbf{q}], \mathbf{f}(\mathbf{p}', \mathbf{q}') = \mathbf{0}\} \\
& = & \mathbb{P}^{[\mathbf{q}]}
\end{array}$$

□

An illustration related to our Example 1 is provided by Figure 6. We have $\sigma = (-1, -2)$ which corresponds to the central symmetry with respect to the origin. We have $[q] = [-3 + \frac{\sqrt{3}}{2}, 3]$ and $\sigma_q[q] = [-3, 3 - \frac{\sqrt{3}}{2}]$.

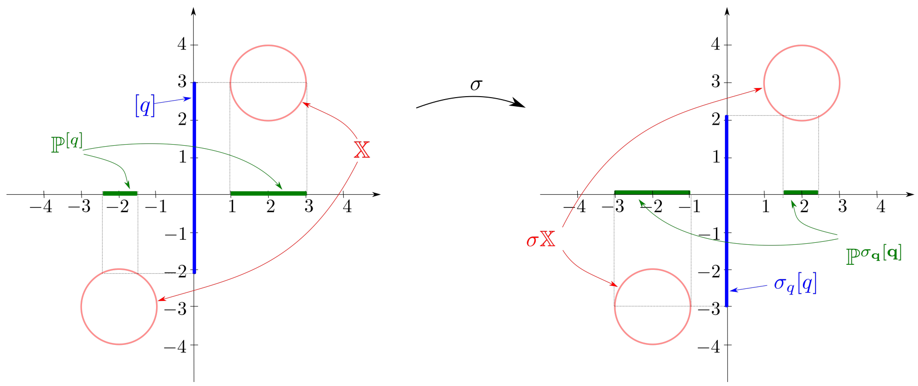


Fig. 6: Illustration of Proposition 1

Theorem 2. *(decomposition Theorem) Assume that the choice function ψ for the quotient set $\mathcal{Q} = \frac{sep(B_n(\mathbb{X}))}{\sim}$ satisfies $dom(\psi) = \mathcal{E}^\ell$. Define the positive quadrant $[\mathbf{a}] = (\mathbb{R}^+)^\ell$. The projected set is given by*

$$\mathbb{P}^{[\mathbf{q}]} = \bigcup_{\sigma \in \psi(sgn([\mathbf{q}]))} \sigma_{\mathbf{p}} \mathbb{P}^{[\mathbf{a}] \cap \sigma_{\mathbf{q}}[\mathbf{q}]}. \tag{22}$$

Proof. The assumption $dom(\psi) = \mathcal{E}^\ell$ means that there is no ε such that $\psi(\varepsilon)$ is undefined. Equivalently, it means that we got enough symmetries to move from any quadrant to the positive quadrant $[\mathbf{a}]$. Since $\mathcal{E}^\ell$ has 2^ℓ elements, we can write: $\mathcal{E}^\ell = \{\varepsilon_1, \varepsilon_2, \varepsilon_3, \dots, \varepsilon_{2^\ell}\}$. The element $\sigma^i = \psi(\varepsilon_i), i \in \{1, \dots, 2^n\}$ corresponds to the symmetry which moves the ith quadrant $\mathbb{R}^{\varepsilon_i}$ to the positive quadrant. Since $\sigma^i \in sep(B_n)$, it can be decomposed as $\sigma^i = (\sigma^i_{\mathbf{p}}, \sigma^i_{\mathbf{q}})$. Define the box

$$[\mathbf{q}_i] = (\sigma^i_{\mathbf{q}})^{-1}([\mathbf{a}] \cap \sigma^i_{\mathbf{q}}([\mathbf{q}]))) \tag{23}$$

which corresponds to the part of $[\mathbf{q}]$ which is in the quadrant $\mathbb{R}^{\varepsilon_i}$. The set $\mathcal{Q} = \{[\mathbf{q}_1], [\mathbf{q}_2], \dots\}$ is thus a partition of $[\mathbf{q}]$ among all quadrants. For each $[\mathbf{q}_i]$, the vector sign $\varepsilon_i = sgn([\mathbf{q}_i])$ is such that $[\mathbf{q}_i] \subseteq \mathbb{R}^{\varepsilon_i}$. We have

$$\begin{aligned} \mathbb{P}^{[\mathbf{q}]} &= \bigcup_{i \in \{1,2,\dots\}} \mathbb{P}^{[\mathbf{q}_i]} \\ &\overset{(21)}{=} \bigcup_{i \in \{1,2,\dots\}} \sigma^i_{\mathbf{p}} \mathbb{P}^{\sigma^i_{\mathbf{q}}[\mathbf{q}_i]} \\ &\overset{(23)}{=} \bigcup_{i \in \{1,2,\dots\}} \sigma^i_{\mathbf{p}} \mathbb{P}^{[\mathbf{a}] \cap \sigma^i_{\mathbf{q}}[\mathbf{q}]} \end{aligned} \tag{24}$$

Now, $[\mathbf{a}] \cap \sigma^i_{\mathbf{q}}[\mathbf{q}]$ is empty if $\varepsilon_i \notin sgn([\mathbf{q}])$, *i.e.*, if $\sigma^i \notin \psi(sgn([\mathbf{q}]))$, we conclude that

$$\mathbb{P}^{[\mathbf{q}]} = \bigcup_{(\sigma_{\mathbf{p}}, \sigma_{\mathbf{q}}) \in \psi(sgn([\mathbf{q}]))} \sigma_{\mathbf{p}} \mathbb{P}^{[\mathbf{a}] \cap \sigma_{\mathbf{q}}[\mathbf{q}]}. \tag{25}$$

□

3.2 Computing $sep(B_n(\mathbb{X}))$

To build the quotient $\mathcal{Q} = \frac{sep(B_n(\mathbb{X}))}{\sim}$, we need the symmetries of $sep(B_n(\mathbb{X}))$. For this, we will use the following rules

$$\begin{array}{rcll} \mathbf{x} \in \mathbb{X}, \sigma(\mathbf{x}) \notin \mathbb{X} & \Rightarrow & \sigma \notin B_n(\mathbb{X}) & (i) \\ \sigma_1 \in B_n(\mathbb{X}), \sigma_2 \in B_n(\mathbb{X}) & \Rightarrow & \sigma_1\sigma_2 \in B_n(\mathbb{X}) & (ii) \end{array} \quad (26)$$

where the multiplication in $B_n(\mathbb{X})$ corresponds to composition, *i.e.*, $\sigma_1\sigma_2 = \sigma_1 \circ \sigma_2$. When it does not fail, Algorithm 1 (SepBnX) computes $sep(B_n(\mathbb{X}))$. It requires the knowledge of a list of points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots\}$ of $\mathbb{X}$, and a list of symmetries $\mathcal{S}$ of $\mathbb{X}$.

Algorithm 1 SepBnX generates the group $sep(B_n(\mathbb{X}))$

	Input: $\mathbf{f}(\mathbf{p}, \mathbf{q})$, $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots\}$, $\mathcal{S} = \{\sigma_1, \sigma_2, \dots\}$
1	$m = \dim \mathbf{p}$; $\ell = \dim \mathbf{q}$; $n = m + \ell$
2	Generate the list $\mathcal{B}$ of all symmetries in B_n
3	Remove from $\mathcal{B}$ all σ that are not separable
4	Remove from $\mathcal{B}$ all σ such that $\sigma(\mathbf{x}_i) \notin \mathbb{X}$ for at least one $\mathbf{x}_i \in \mathcal{X}$
5	Compute the group $\langle \mathcal{S} \rangle$ generated by $\mathcal{S}$
6	If $\mathcal{B} = \langle \mathcal{S} \rangle$ then return $\mathcal{B}$
7	Return "Failure: symmetries should be added to $\mathcal{S}$ or solutions should be added to $\mathcal{X}$".

The input of Algorithm 1 (SepBnX) are

- a function defining the set $\mathbb{X}$ where $\mathbf{x} = (\mathbf{p}, \mathbf{q})$,
- a list $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots\}$ of elements of $\mathbb{X}$ and
- a list $\mathcal{S}$ of symmetries of $\mathbb{X}$.

Algorithm 1 uses $\mathcal{X}$ to compute an over-approximation of the set of symmetries and $\mathcal{S}$ to compute an under-approximation, and hence equality of the both implies a correct result.

Step 2 generates a list of $2^n \cdot 2!$ vectors σ of $\mathbb{Z}^n$. The vectors σ correspond to symmetries of B_n. The components of σ should all have different absolute values and be inside $\{-n, \dots, -1, 1, \dots, n\}$.

Step 3 removes from $\mathcal{B}$ all vectors σ such that the m first components σ are not inside the set $\{-m, \dots, -1, 1, \dots, m\}$.

Step 4 eliminates symmetries σ that are not inside $B_n(\mathbb{X})$ from the list of solutions $\mathcal{X}$. This is done by Rule *(i)* of (26). The entries for the solutions $\mathbf{x}_i$ should be rational with radicals so that we can check the membership property $\mathbf{f}(\mathbf{x}_i) = \mathbf{0}$ symbolically.

Step 5 computes the group $\langle \mathcal{S} \rangle$ generated by the symmetries given as an input list $\mathcal{S}$ of the algorithm. The group is generated by successive compositions of the symmetries in $\mathcal{S}$ using Rule *(ii)* of (26).

Step 6. Since $\langle \mathcal{S} \rangle \subseteq sep(B_n(\mathbb{X}))$ and since $sep(B_n(\mathbb{X})) \subseteq \mathcal{B}$, if $\mathcal{B} = \langle \mathcal{S} \rangle$ then $\mathcal{B}$ corresponds to $sep(B_n(\mathbb{X}))$ and the algorithm returns $\mathcal{B}$.

Step 7. The algorithm fails. We can return an interval of symmetries $[\langle \mathcal{S} \rangle, \mathcal{B}]$ which can be used to propose other symmetries as an input of the algorithm in $\mathcal{S}$ in order to inflate $\langle \mathcal{S} \rangle$. The user may also decide to add other solutions in $\mathcal{X}$ in order to reduce $\mathcal{B}$.

3.3 Build the choice function ψ

Once $\mathcal{Q} = sep(B_n(\mathbb{X}))$ has been computed, we need to select a minimal number of them. This corresponds to the construction of the choice function ψ which is performed by Algorithm 2, named GenePsi.

Algorithm 2 GenePsi to build the choice function ψ associated to $sep(B_n(\mathbb{X}))$

	Input: $\mathcal{Q}$
1	For each pair (σ_1, σ_2) of $\mathcal{S}$ such that $\varphi(\sigma_1) = \varphi(\sigma_2)$, remove σ_2 from $\mathcal{Q}$.
2	For each $\varepsilon \in \mathcal{E}^\ell$, define $\psi(\varepsilon) = \varphi_{\mid \mathcal{Q}}^{-1}$.
3	Return ψ

We feed the algorithm GenePsi with $\mathcal{Q} = sep(B_n(\mathbb{X}))$ computed by Algorithm 1 (SepBnX).

Step 1 selects a representative for each element of the quotient $\mathcal{Q}$. It means that for each quadrant ε of $\mathbb{R}^\ell$, the set $\varphi^{-1}(\varepsilon)$ will be replaced by a unique representative $\sigma = \psi(\varepsilon)$. This correspond to a quotient of a set by the equivalence relation $\sim$.

Step 2 generates the choice function $\psi(\varepsilon)$ under the form of a dictionary (or table). Note that the set $\varphi_{\mid \mathcal{Q}}^{-1} = \{\sigma \in \mathcal{Q} \mid \varphi(\sigma) = \varepsilon\}$ is always a singleton, since Step 1 made φ injective.

3.4 Separators

This section recalls the basic notions on intervals, contractors and separators that are needed to understand how the paver will approximate the solution set. A *contractor* $\mathcal{C}$ for the set $\mathbb{X} \subset \mathbb{R}^n$ is an operator $\mathbb{IR}^n \mapsto \mathbb{IR}^n$ which satisfies

$$\begin{array}{ll} \mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}] & \text{(contractance)} \\ [\mathbf{x}] \subset [\mathbf{y}] \;\Rightarrow\; \mathcal{C}([\mathbf{x}]) \subset \mathcal{C}([\mathbf{y}]). & \text{(monotonicity)} \\ \mathcal{C}([\mathbf{x}]) \cap \mathbb{X} = [\mathbf{x}] \cap \mathbb{X} & \text{(consistency)} \end{array} \tag{27}$$

where $\mathbb{IR}^n$ is the set of axis-aligned boxes in $\mathbb{R}^n$. Figure 7(a) illustrates the notion of contractor. A contractor for $\mathbb{X}$ can be used inside a paver (an algorithm which bisects boxes and uses $\mathcal{C}$ to eliminate parts of the search space that are outside the solution set) to provide an outer approximation of $\mathbb{X}$. Figure 7(b) shows the paving generated by the paver. It corresponds to an outer approximation of $\mathbb{X}$. The blue boxes are outside $\mathbb{X}$.

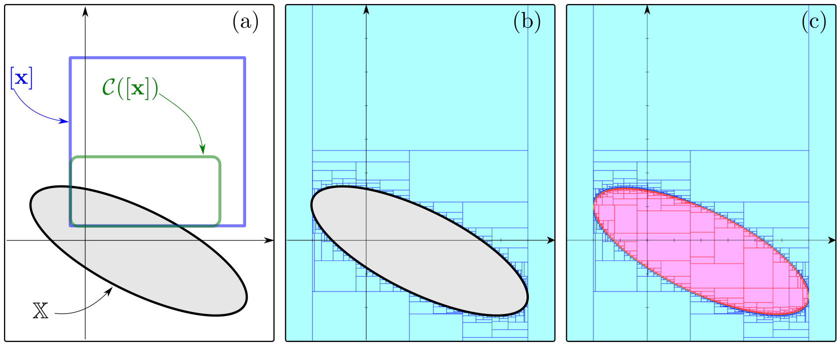


Fig. 7: (a) The box $[\mathbf{x}]$ is contracted by the contractor $\mathcal{C}$; (b) A paver uses the contractor $\mathcal{C}$ to get an outer approximation of $\mathbb{X}$; (c) A paver uses the separator $\mathcal{S}$ to get an outer and an inner approximations of $\mathbb{X}$

The contractor $\mathcal{C}$ is *minimal* if $\mathcal{C}([\mathbf{x}])$ corresponds exactly to the smallest box that can be obtained by a contraction of $[\mathbf{x}]$ without removing a single point of $\mathbb{X}$.

If $\mathcal{C}_1$ and $\mathcal{C}_2$ are two contractors, we define the following operations on contractors [5]:

$$\begin{aligned} (\mathcal{C}_1 \cap \mathcal{C}_2)([\mathbf{x}]) &= \mathcal{C}_1([\mathbf{x}]) \cap \mathcal{C}_2([\mathbf{x}]) && (28)\\ (\mathcal{C}_1 \cup \mathcal{C}_2)([\mathbf{x}]) &= \mathcal{C}_1([\mathbf{x}]) \sqcup \mathcal{C}_2([\mathbf{x}]) && (29)\\ (\mathcal{C}_1 \circ \mathcal{C}_2)([\mathbf{x}]) &= \mathcal{C}_1 (\mathcal{C}_2([\mathbf{x}])) && (30) \end{aligned}$$

where $\sqcup$ is the interval union hull, *i.e.*, $[\mathbf{a}] \sqcup [\mathbf{b}]$ corresponds to the smallest box which encloses the two boxes $[\mathbf{a}]$ and $[\mathbf{b}]$. If $\sigma : \mathbb{R}^n \to \mathbb{R}^n$ is a symmetry, we define the *action* by σ of a contractor $\mathcal{C}$ for $\mathbb{X}$ as [9]:

$$\sigma\mathcal{C} = \sigma \circ \mathcal{C} \circ \sigma^{-1}. \tag{31}$$

It can be shown that $\sigma\mathcal{C}$ is a contractor for $\sigma\mathbb{X}$.

In order to characterize both an inner and outer approximation of the set $\mathbb{X}$, we need the notion of *separator*, as illustrated by Figure 7(c). A *separator* $\mathcal{S}$ for $\mathbb{X}$ is a pair of contractors $\{\mathcal{S}^{\text{in}}, \mathcal{S}^{\text{out}}\}$ such that, for all $[\mathbf{x}] \in \mathbb{IR}^n$, we have

$$\begin{array}{ll} \mathcal{S}^{\text{in}}([\mathbf{x}]) \cup \mathcal{S}^{\text{out}}([\mathbf{x}]) = [\mathbf{x}] & \text{(complementarity)} \\ \mathcal{S}^{\text{out}}([\mathbf{x}]) \cap \mathbb{X} = [\mathbf{x}] \cap \mathbb{X} & \text{(outer consistency)} \\ \mathcal{S}^{\text{in}}([\mathbf{x}]) \cap \mathbb{X}^C = [\mathbf{x}] \cap \mathbb{X}^C & \text{(inner consistency)} \end{array} \tag{32}$$

where $\mathbb{X}^C = \{\mathbf{x} \mid \mathbf{x} \notin \mathbb{X}\}$. We write $\mathcal{S} \sim \mathbb{X}$ if $\mathcal{S}$ is a separator for $\mathbb{X}$.

A separator $\mathcal{S}$ is *minimal* if its two contractors $\mathcal{S}^{\text{in}}$ and $\mathcal{S}^{\text{out}}$ are both minimal. We define the following operations

$$\begin{array}{cccc} \mathcal{S}_1 \cap \mathcal{S}_2 & = & \left\{\mathcal{S}_1^{\text{in}} \cup \mathcal{S}_2^{\text{in}}, \mathcal{S}_1^{\text{out}} \cap \mathcal{S}_2^{\text{out}}\right\} & \text{(intersection)} \\ \mathcal{S}_1 \cup \mathcal{S}_2 & = & \left\{\mathcal{S}_1^{\text{in}} \cap \mathcal{S}_2^{\text{in}}, \mathcal{S}_1^{\text{out}} \cup \mathcal{S}_2^{\text{out}}\right\} & \text{(union)} \\ \mathcal{S}^C & = & \{\mathcal{S}^{\text{out}}, \mathcal{S}^{\text{in}}\} & \text{(complement)} \\ \sigma\mathcal{S} & = & \left\{\sigma \circ \mathcal{S}_{\mathbb{X}}^{\text{in}} \circ \sigma^{-1}, \sigma \circ \mathcal{S}_{\mathbb{X}}^{\text{out}} \circ \sigma^{-1}\right\} & \text{(action)} \end{array} \tag{33}$$

We have [15]:

$$\left\{\begin{array}{ccc}\mathcal{S}_1 & \sim & \mathbb{X}_1\\ \mathcal{S}_2 & \sim & \mathbb{X}_2\\ \mathcal{S} & \sim & \mathbb{X}\end{array}\right. \Rightarrow \left\{\begin{array}{ccc}\mathcal{S}_1\cap\mathcal{S}_2 & \sim & \mathbb{X}_1\cap\mathbb{X}_2\\ \mathcal{S}_1\cup\mathcal{S}_2 & \sim & \mathbb{X}_1\cup\mathbb{X}_2\\ \mathcal{S}^C & \sim & \mathbb{X}^C\\ \sigma\mathcal{S} & \sim & \sigma\mathbb{X}\end{array}\right. \tag{34}$$

As an illustration, consider the set

$$\mathbb{X} = \left\{\mathbf{x}\in\mathbb{R}^2, (x_1-2)^2+(3x_2+x_1-1)^2\in[0,4]\right\} \tag{35}$$

using a paver with a separator $\mathcal{S}$ for $\mathbb{X}$, we get Figure 8(a). The blue part has been eliminated by the outer contractor $\mathcal{S}^{out}$ of $\mathcal{S}$. Whereas the magenta part has been eliminated by the inner contractor $\mathcal{S}^{in}$. Figure 8(b) has been computed using the separator $\sigma\mathcal{S}$ where $\sigma=(2\,1)$ corresponds to the axial symmetry with respect the axis $x_1-x_2=0$. We therefore get an approximation for $\sigma\mathbb{X}$. Figure 8(c) is obtained using the separator $\sigma\mathcal{S}\cup\mathcal{S}$ and corresponds to an approximation for $\sigma\mathbb{X}\cup\mathbb{X}$.

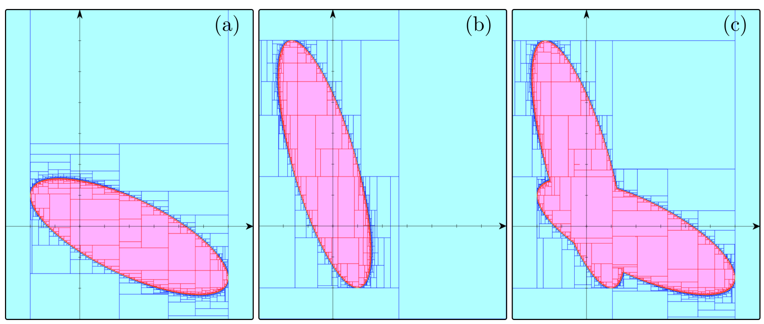


Fig. 8: A paver is used to compute an inner and an outer approximations of 3 sets: (a) an ellipse $\mathbb{X}$, (b) its symmetric $\sigma\mathbb{X}$ by σ, (c) the union of the two sets $\sigma\mathbb{X}\cup\mathbb{X}$. The frame box is $[-3,7]\times[-3,7]$

3.5 Symmetries and monotonicity to build contractors

Before considering projections with separators, let us spend some times to understand how symmetries and monotonicity can be combined to build contractors for sets defined by non-monotonic constraints. For this purpose, let us consider the set

$$\mathbb{X} = \{(x_1,x_2)\in\mathbb{R}^2 \,|\, x_2=\sin x_1\}. \tag{36}$$

In order to build a contractor for $\mathbb{X}$, we first take the box $[\mathbf{a}]=[a_1]\times[a_2]=[-\frac{\pi}{2},\frac{\pi}{2}]\times[-1,1]$ and we define the seed set:

$$\mathbb{X}_a = \mathbb{X}\cap[\mathbf{a}]. \tag{37}$$

Since inside $[\mathbf{a}]$, the sine function is monotonic, a contractor for $\mathbb{X}_a$ is

$$\mathcal{C}_a \begin{pmatrix} [x_1] \\ [x_2] \end{pmatrix} = \begin{pmatrix} [x_1] \cap \arcsin([x_2] \cap [a_2]) \\ [x_2] \cap \sin([x_1] \cap [a_1]) \end{pmatrix}. \tag{38}$$

The seed contractor that will be used as a brick to build the contractor for $\mathbb{X}$. A paver with the seed contractor $\mathcal{C}_a$ yields the approximation of $\mathbb{X}_a$ depicted in Figure 9(a). If we consider the axial symmetry $\sigma_{\mathcal{D}}$ of $\mathbb{X}$ with respect to the line $x_1 = \frac{\pi}{2}$, we can get that a contractor for

$$\mathbb{X}_b = \mathbb{X} \cap [\mathbf{b}] = \mathbb{X}_a \cup \sigma_{\mathcal{D}} \mathbb{X}_a \tag{39}$$

where $[\mathbf{b}] = [b_1] \times [b_2] = [-\frac{\pi}{2}, \frac{3\pi}{2}] \times [-1, 1]$. It is given by

$$\mathcal{C}_b = \mathcal{C}_a \cup \sigma_{\mathcal{D}} \mathcal{C}_a \tag{40}$$

as illustrated by Figure 9(b). If we take into account the symmetry $\sigma_{\mathbf{v}}$ with respect to a translation of $\mathbf{v} = (2\pi, 0)$, we are able to build a contractor for the whole set $\mathbb{X}$. Now, in this example, the sine is not a polynomial, so we cannot use the Algorithm 1 to find the symmetries. It is thus our responsibility to provide all symmetries that are needed to reconstruct $\mathbb{X}$ from the set $\mathbb{X}_a$. If we miss one symmetry, we may loose solutions. In two dimensions, we rarely miss symmetries, but in larger dimension, it is different. Algorithm 1 protects us from any omission in the polynomial case. If it fails, we need to provide more inputs to the algorithm.

On this example, we can understand that the monotonicity of the sine function was useful to build the seed contractor $\mathcal{C}_a$, but the monotonicity is not mandatory. What is required is a seed contractor (not necessarily based on the monotonicity) which is efficient and reliable in one box $[\mathbf{a}]$ of the search space and the symmetries that can be used to build the solution set.

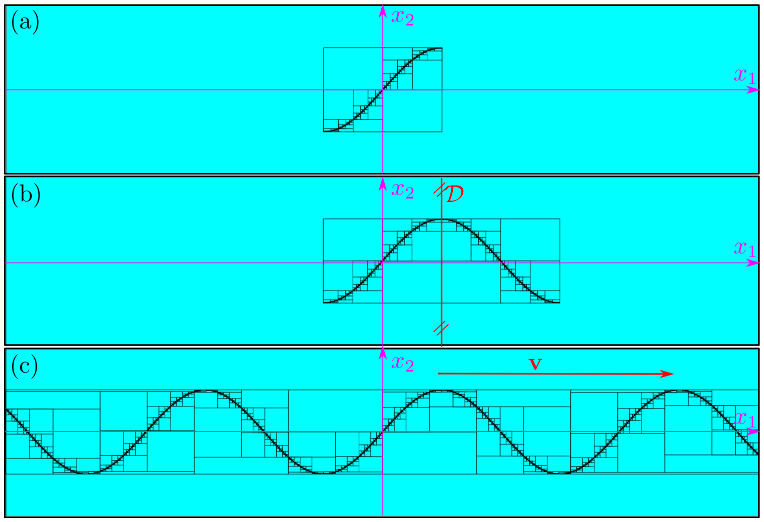


Fig. 9: A paver is used to compute an outer approximations of 3 sets: (a) the seed set $\mathbb{X}_a$, (b) The set $\mathbb{X}_b$ built using the symmetry $\sigma_{\mathcal{D}}$, (c) The set $\mathbb{X}$ using the two symmetries $\sigma_{\mathcal{D}}$ and $\sigma_{\mathbf{v}}$

3.6 Compute the projection set

There is no general method to compute inner and outer approximations of the projection of a set defined by constraints, as for the set $\mathbb{P}^{[\mathbf{q}]}$. In the polynomial case this can be done, using symbolic methods [7], but this is far from trivial for arbitrary polynomial functions, and is quite difficult to do in practice. Interval algorithms have been proposed [23] to compute such projections. The principle is to build contractors (or separators [15]) for projected sets, but this operation requires bisections in the **q**-space (see [14], Section 4.2 or [5] Section 3.2) which make them slow. The problem is much easier when the sign of $\mathbf{q}$ is known and we can even hope to get the minimal separator analytically or using a simple dichotomy. This has been shown in [1] and [6] is the case where constraints are monotonic.

To compute an inner and outer approximation of $\mathbb{P}^{[\mathbf{q}]}$, we first propose to build a separator $\mathcal{S}_0^{[\mathbf{q}]}$ for $\mathbb{P}^{[\mathbf{q}]}$ which works for $[\mathbf{q}] \subseteq [\mathbf{a}] = [0, \infty)^{\ell}$. Then, we build the separator

$$\mathcal{S}^{[\mathbf{q}]} = \bigcup_{(\sigma_{\mathbf{p}}, \sigma_{\mathbf{q}}) \in \psi(sgn([\mathbf{q}]))} \sigma_{\mathbf{p}} \mathcal{S}_0^{[\mathbf{a}] \cap \sigma_{\mathbf{q}}[\mathbf{q}]} \tag{41}$$

which will be valid for an arbitrary $[\mathbf{q}]$ and not only when $[\mathbf{q}] \subseteq [\mathbf{a}]$, as shown by Theorem 2. We can now use a paver to generate an inner and an outer approximation of the solution set $\mathbb{P}^{[\mathbf{q}]}$. This paver performs the following operations. *(i)* It uses the separator $\mathcal{S}^{[\mathbf{q}]}$ to contract and classify zones of the search space that are inside or outside the solution set; *(ii)* It bisects boxes that it cannot contract and *(iii)* It returns as unclassified boxes that are deemed to small to be bisected.

3.7 Example: The rotate constraint

Consider the *rotate* constraint:

$$\begin{cases} q_1 p_1 - q_2 p_2 & = & q_3 \\ q_2 p_1 + q_1 p_2 & = & q_4 \\ q_1^2 + q_2^2 - 1 & = & 0 \end{cases} \tag{42}$$

As illustrated by Figure 10, the red vector corresponds to the angle θ on the trigonometric circle, with $\cos\theta = q_1$ and $\sin\theta = q_2$. If we rotate the vector (p_1, p_2) by the angle θ, we get the vector (q_3, q_4).

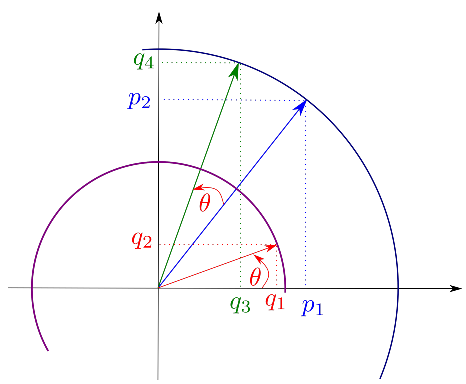


Fig. 10: Illustration of the rotate constraint

We get 46080 elements for B_6. To find $B_6(\mathbb{X})$, we have provided the following symmetries

$$\mathcal{S} = \{(2,-1,-4,3,5,6),(1,-2,3,-4,5,-6),(-1,-2,-4,3,6,-5)\}\,. \tag{43}$$

Here, all elements of S are separable, but it is not needed for the generation of the group $B_6(\mathbb{X})$. We also provided an element of $\mathbb{X}$ given by

$$\mathbf{x}_1 = \left(2, 4, \frac{1}{2}, \frac{\sqrt{3}}{2}, 1 - 2\sqrt{3}, \sqrt{3} + 2\right). \tag{44}$$

The value for $\mathbf{x}_1$ has been chosen to be non trivial in order to eliminate as many unfeasible symmetries as possible. Finally, we got a list of 32 elements for the group $sep(B_6(\mathbb{X}))$. The quotient $\mathcal{Q}$ is obtained by removing symmetries with the same value for φ. We obtained a set $\mathcal{Q}$ of 16 elements, which is consistent with the number of quadrants of $\mathbb{R}^\ell, \ell = 4$. The resulting choice function ψ is

$$\psi : \begin{cases} (++++) & \mapsto & (1,2,3,4,5,6) \\ (+++-) & \mapsto & (2,-1,3,4,6,-5) \\ (++-+) & \mapsto & (-2,1,3,4,-6,5) \\ (++--) & \mapsto & (-1,-2,3,4,-5,-6) \\ (+-++) & \mapsto & (2,1,3,-4,6,5) \\ (+-+-) & \mapsto & (1,-2,3,-4,5,-6) \\ (+--+) & \mapsto & (-1,2,3,-4,-5,6) \\ (+---) & \mapsto & (-2,-1,3,-4,-6,-5) \\ (-+++) & \mapsto & (-2,-1,-3,4,6,5) \\ (-++-) & \mapsto & (-1,2,-3,4,5,-6) \\ (-+-+) & \mapsto & (1,-2,-3,4,-5,6) \\ (-+--) & \mapsto & (2,1,-3,4,-6,-5) \\ (--++) & \mapsto & (-1,-2,-3,-4,5,6) \\ (--+-) & \mapsto & (-2,1,-3,-4,6,-5) \\ (---+) & \mapsto & 2,-1,-3,-4,-6,5 \\ (----) & \mapsto & (1,2,-3,-4,-5,-6) \end{cases}$$

Recall that the fact that $\sigma = \psi\,(+--+) = (-1,2,3,-4,-5,6)$, (7th line in the previous formula) corresponds to the following equivalence

$$\begin{pmatrix} q_1p_1 - q_2p_2 - q_4 \\ q_2p_1 + q_1p_2 - q_5 \\ q_1^2 + q_2^2 - 1 \end{pmatrix} = \mathbf{0} \Leftrightarrow \begin{pmatrix} q_1(-p_1) - (-q_2)p_2 - (-q_4) \\ (-q_2)(-p_1) + q_1p_2 - q_5 \\ q_1^2 + (-q_2)^2 - 1 \end{pmatrix} = \mathbf{0}$$

This symmetry σ can be interpreted as a way to transport the quadrant $(+--+)$ in the $\mathbf{q}$-space to the quadrant $(++++)$.

For $[\mathbf{q}] = (\cos([\theta]), \sin([\theta]), [-10,10], [5,12])$ with $[\theta] = [3,4]$, we get the approximation for $\mathbb{P}$ given by Figure 11, Left in less than 0.3 sec on a standard laptop [13]. The frame box corresponds to $[\mathbf{p}] = [-20,20]^2$. Figure 11, Right, represents the box $[q_3] \times [q_4]$ in blue and a sampling of rotated rectangles with different $\theta \in [\theta]$.

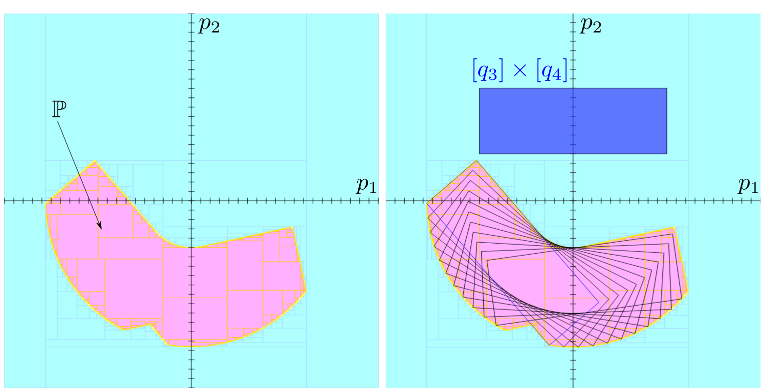


Fig. 11: Rotating rectangle

4 Applications

This section provides two different test-cases to illustrate the efficiency of our projection algorithm.

4.1 Workspace

We consider an object (blue in Figure 12, Right) which corresponds to a polygon $\mathcal{M}$ with coordinates

$$\begin{pmatrix} 5 & 5 & -4 & -4 & 11 & 11 & 10 & 10 \\ -8 & 8 & 8 & 10 & 10 & 8 & 8 & -8 \end{pmatrix} \tag{45}$$

which can be obtained by the union of two boxes:

$$\mathcal{M} = \underbrace{([5,10] \times [-8,8])}_{=[\mathbf{m}_1]} \cup \underbrace{([-4,11] \times [8,10])}_{=[\mathbf{m}_2]}. \tag{46}$$

The object has one degree of freedom: it can rotate around the origin with an angle $\theta \in [\theta] = [0.5, 1.5]$. The workspace [17] corresponds to all space that can be occupied by the object. It is given by

$$\mathbb{P} \quad = \quad \bigcup_{i \in \{1,2\}} \mathbb{P}_i$$

where

$$\mathbb{P}_i = \left\{ \mathbf{p} \in \mathbb{R}^2 \mid \exists q_1 \in \cos([\theta]), \exists q_2 \in \sin([\theta]), \exists (q_3, q_4) \in [\mathbf{m}_i], \, rotate(\mathbf{p}, \mathbf{q}) \right\}.$$

Since we have a separator for the $\mathbb{P}_i$, we are able to get in less than 1 sec an inner and an outer approximations for $\mathbb{P}$. The frame box corresponds to $[\mathbf{p}] = [-20, 20]^2$.

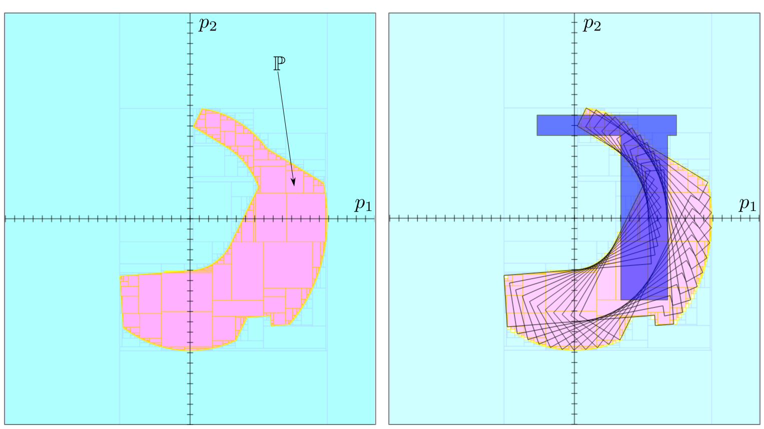


Fig. 12: Workspace of the blue object in rotation around 0

4.2 Speed estimation

We have one object moving with an unknown speed $\mathbf{v}$. The speed is measured by 6 robots. In their own frame, they are able to give a box enclosing the speed they measure. Now, the orientation of the ith robot θ_i is known with a large uncertainty ($\pm 1 rad$), as shown in the following table:

i	1	2	3	4	5	6
θ^i	$[1, 2]$	$[2, 3]$	$[3, 4]$	$[-0, 1]$	$[-2, -1]$	$[-3, -2]$
y_1^i	$[12, 14]$	$[-2, 0]$	$[-10, -8]$	$[10, 12]$	$[-8, -6]$	$[-14, -12]$
y_2^i	$[-6, -4]$	$[-16, -14]$	$[-12, -10]$	$[8, 10]$	$[12, 14]$	$[2, 4]$

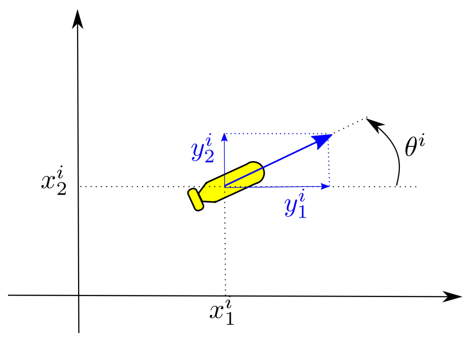


Fig. 13: The ith robot has a position (x_1^i, x_1^2), an orientation θ^i and a speed (y_1^i, y_2^i)

The set of all feasible speed vectors is

$$\mathbb{V} = \bigcap_{i \in \{1,\dots,6\}} \mathbb{V}_i \tag{47}$$

where

$$\mathbb{V}_i = \left\{ \mathbf{v} \in \mathbb{R}^2 \mid \exists q_1 \in \cos([\theta^i]), \exists q_2 \in \sin([\theta^i]), \exists q_3 \in [y_1^i], \exists q_4 \in [y_2^i], \, rotate(\mathbf{v}, \mathbf{q}) \right\}.$$

We get the first Subfigure of Figure 14. Since we have separators for the sets $\mathbb{V}_i$, we can compute more complex combinations such as the relaxed intersection, which allow us to be robust with respect to some outliers. As defined in [16], the relaxed intersection

$$\mathbb{V}^{\{k\}} = \bigcap_{i \in \{1,\dots,6\}}^{\{k\}} \mathbb{V}_i \ , \tag{48}$$

is the set of all points that belong to all $\mathbb{V}_i$ except k of them. For $k = 1, 2, 3$, in less than 1 sec, we get the approximations provided by Figure 14.

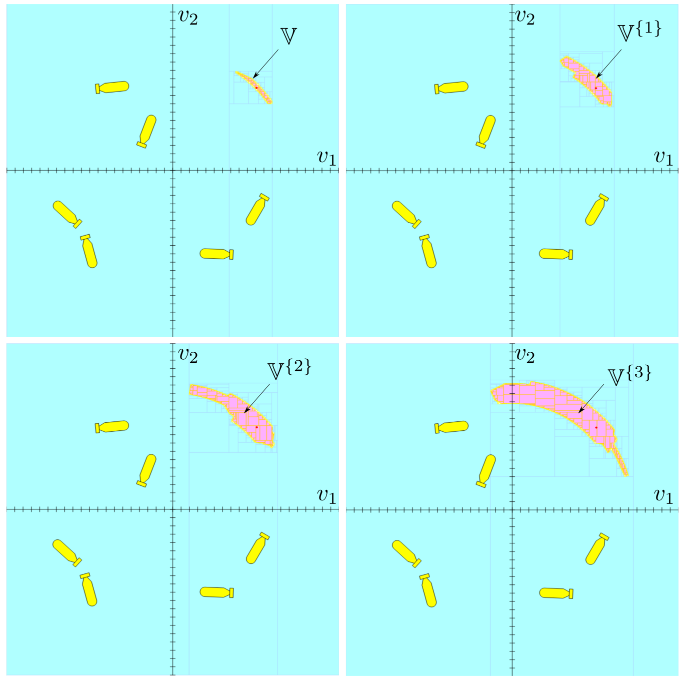


Fig. 14: Speed estimation for different degrees of relaxed intersection

5 Conclusion

In this paper, a new method for computing an inner and an outer approximations for a set defined as a projection of polynomial equations has been proposed. To avoid bisections with respect to variables $\mathbf{q}$ along which the projection is defined, we propose to solve the problem in one quadrant of the $\mathbf{q}$-space and then to use symmetries to extend the zone over which the projection works.

To develop the algorithm, we had to use different concepts that are not common in the domain of interval analysis such as the symmetries of the unit cube, the quotient by an equivalence relation, and the choice function ψ. Note that following the results presented in [12], the separators we obtain are minimal.

As a result, we were able to treat two important applications efficiently. The first one is the approximation of the workspace of one object with one degree of freedom. The second application is the estimation of the speed of one object with several observers the orientation of which is uncertain.

Our approach is limited by the fact that we need to be able to find seed separator on the positive quadrant, which is not a trivial task, even if the monotonicity can be very helpful as shown in the applications. An other limitation is that we need enough symmetries so that the visibility of the set $\mathbb{X}$ on the positive quadrant can be sufficient to reconstruct the set $\mathbb{X}$ globally.

Note. The `Python` programs associated with all examples are given in [13].

References

[1] I. Araya, G. Trombettoni, and B. Neveu. Exploiting monotonicity in interval constraint propagation. In M. Fox and D. Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010. 3.6

[2] E. Bloch. *Proofs and fundamentals: a first course in abstract mathematics*. Springer, London, 2011. 2.5

[3] J. Bochnak, M. Coste, and M.F. Roy. *Real algebraic geometry*. Springer Science and Business Media, London, 2013. 1, 1

[4] B. Caviness and J. Johnson. *Quantifier Elimination and Cylindrical Algebraic Decomposition*. Springer Science & Business Media, December 2012. 1

[5] G. Chabert and L. Jaulin. Contractor Programming. *Artificial Intelligence*, 173:1079–1100, 2009. 3.4, 3.6

[6] G. Chabert and L. Jaulin. Hull consistency under monotonicity. In Ian P. Gent, editor, *Principles and Practice of Constraint Programming - CP 2009, 15th International Conference, CP 2009, Lisbon, Portugal, September 20-24, 2009, Proceedings*, volume 5732 of *Lecture Notes in Computer Science*, pages 188–195. Springer, 2009. 3.6

[7] G. Collins. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In *Second GI Conf. Automata Theory and Formal Languages*, volume 33. Springer LNCS 33, 1975. 1, 3.6

[8] H. Coxeter. *The Beauty of Geometry: Twelve Essays*. Dover Books on Mathematics, 1999. 2.2, 2.3

[9] B. Desrochers and L. Jaulin. A minimal contractor for the polar equation; application to robot localization. *Engineering Applications of Artificial Intelligence*, 55:83–92, 2016. 3.4

[10] A. Ehambram, R. Voges, C. Brenner, and B. Wagner. Interval-based visual-inertial lidar SLAM with anchoring poses. In *2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022*, pages 7589–7596. IEEE, 2022. 1

[11] A. Goldsztejn and L. Jaulin. Inner and outer approximations of existentially quantified equality constraints. In *Proceedings of the Twelfth International Conference on Principles and Practice of Constraint Programming, (CP 2006)*, Nantes (France), 2006. 1

[12] L. Jaulin. Actions of the hyperoctahedral group to compute minimal contractors. *Artif. Intell.*, 313:103790, 2022. 5

[13] L. Jaulin. Codes associated with the paper entitled: Inner and outer characterization of the projection of polynomial equations using symmetries, quotients and intervals. `www.ensta-bretagne.fr/jaulin/quotient.html`, 2022. 3.7, 5

[14] L. Jaulin, I. Braems, and E. Walter. Interval methods for nonlinear identification and robust control. In *In Proceedings of the 41st IEEE Conference on Decision and Control (CDC)*, 9-13 decembre 2002, Las Vegas, 2002. 1, 3.6

[15] L. Jaulin and B. Desrochers. Introduction to the algebra of separators with application to path planning. *Engineering Applications of Artificial Intelligence*, 33:141–147, 2014. 3.4, 3.6

[16] L. Jaulin and E. Walter. Guaranteed robust nonlinear minimax estimation. *IEEE Transaction on Automatic Control*, 47(11):1857–1864, 2002. 4.2

[17] J.P. Merlet and D. Daney. Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, April 18-22, 2005, Barcelona, Spain*, pages 942–947. IEEE, 2005. 4.1

[18] W. Miller. *Symmetry Groups and Their Applications*. Academic Press, New York, 1972. 1

[19] B. Mishra. *Algorithmic Algebra*. Springer, New York, 1993. 1

[20] R.E. Moore, R.B. Kearfott, and M.J. Cloud. *Introduction to Interval Analysis*. SIAM, Philadelphia, PA, 2009. 1

[21] M. Pelleau, C. Truchet, and F. Benhamou. The octagon abstract domain for continuous constraints. *Constraints An Int. J.*, 19(3):309–337, 2014. 2.3

[22] N. Ramdani, L. Travé-Massuyès, and C. Jauberthie. Mode discernibility and bounded-error state estimation for nonlinear hybrid systems. *Autom.*, 91:118–125, 2018. 1

[23] S. Ratschan. Approximate quantified constraint solving by cylindrical box decomposition. *Reliable Computing*, 8(1):21–42, 2002. 1, 3.6

[24] A. Rauh. *Sensitivity Methods for Analysis and Design of Dynamic Systems with Applications in Control Engineering*. Shaker–Verlag, 2017. 1

[25] S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, and S. Veres. *Reliable Robot Localization*. Wiley, dec 2019. 1

[26] S. Schön and H. Kutterer. Using zonotopes for overestimation-free interval least-squares-some geodetic applications. *Reliab. Comput.*, 11(2):137–155, 2005. 1

[27] P. van Hentenryck, Y. Deville, and L. Michel. *Numerica: A Modeling Language for Global Optimization*. MIT Press, Boston, MA, 1997. 1

[28] H. Wussing. *The Genesis of the Abstract Group Concept: A Contribution to the History of the Origin of Abstract Group Theory*. Dover Publications, 2007. 2.3

[29] E. Zermelo. Beweis, dass jede Menge wohlgeordnet werden kann. *Mathematische Annalen*, 59(4), 1904. 2.6