Bisectable Abstract Domains for the Resolution of Equations Involving Complex Numbers*

L. Jaulin
ENSTA-Bretagne, Lab-STICC, 2 rue F. Verny, 29200 Brest, France
lucjaulin@gmail.com

B. Desrochers
Direction Générale de l'Armement, Techniques Navales Brest, BCRM de Brest, France
benoit.desrochers@intradef.gouv.fr

D. Massé
UBO, Lab-STICC, 20 avenue Le Gorgeu, 29238 Brest, France
damien.masse@univ-brest.fr

Abstract

The idea of interval arithmetic, proposed by Moore, is to enclose the exact value of a real number inside an interval. Then, computing with intervals will allow us to enclose the true value for a variable we want to compute. This paper emphases the importance of having a lattice structure for the set of intervals and shows that several interval algorithms could be adapted to other types of domains as soon as these domains have a lattice structure with respect to the inclusion and that we could bisect them. Such domains will be called *bisectable abstract domains* (or 'bad' for short). As an illustration, we introduce the *boxpies*, which correspond to the intersection between one box and one pie. We show that *boxpies* can be used efficiently to characterize the solution set of constraints involving complex numbers.

Keywords: interval analysis, bisectable abstract domains, abstract interpretation, contractors
AMS subject classifications: 65-00, 65G99

1 Introduction

This paper proposes to use interval analysis and contractor programming [1] with the objective of solving equations involving complex numbers. Our approach uses concepts

*Submitted: September 22, 2015; Revised: November 11, 2015; Accepted: November 17, 2015.

of interval analysis developed by Moore [25], but adapts and extends these concepts to be more efficient.

A problem involving equations with complex numbers always can be rewritten as real-valued equations through a Cartesian decomposition of the complex variables. As a consequence, classical interval based methods always can be used to solve this type of problem, without developing a new type of method. Unfortunately, such a scalar decomposition makes more complex the expressions of the equations, which amplifies the negative effect of the dependency problem. As a consequence, the resulting solvers are slow.

In his book, Moore [25] has shown that the family of all real intervals is closed under point-wise arithmetic operations $(+, -, \cdot)$ (*i.e.*, this family forms an arithmetic). This closedness is used to estimate efficiently the set of possible values for $y = f(x_1, \dots, x_n)$ from the known intervals of possible values for x_i when the x_i's are complex valued and it is desirable to find a similar closed family (arithmetic) of complex sets. Unfortunately, we now know from Nickel's paper [29] that, in contrast to the 1-D interval case, there is no finite-dimensional arithmetic for complex sets [22]. Several people have proposed to use different types of sets enclose an uncertain complex number, such as a disk [13], a pie [21], or a box. Further, they introduced corresponding unclosed arithmetics. Each type of domain has some advantages and drawbacks; to our knowledge, no method has been proposed to combine them.

In this paper, we propose to take advantage of the dual representation of complex numbers (Cartesian or polar form) and use the two different types of domains to enclose the solutions: Cartesian intervals (or boxes) and polar intervals (or pies) [4]. Another contribution of this paper is to show that to be able to find an inner and an outer approximation for the solution set of constraints involving complex numbers, it is fundamental to make these domains closed under bisection. For this, we will introduce the new notion of *bisectable abstract domains* (or *bad* for short).

The paper is organized as follows. Section 2 defines an interval of a set which is a metric lattice (such as $\mathbb{R}$ or $\mathbb{R}^n$), and Section 3 shows how the concept of *interval* can generalized to deal with the case where the variables to be enclosed do not belong to a lattice. As an illustration, Section 4 considers the set of angles for which no order relation exists. It shows that what is important is not that the variables take values inside a lattice, but that the domains used to enclose them belong to a lattice with respect to the set inclusion. Section 5 introduces the notion of *pie* which is an illustration of how vectors of variables with no order relation (such as angles) can be enclosed. Section 6 shows how different types of domains can be merged into a single type. This is illustrated by introducing the new notion of *boxpie*, which is the intersection between one box and one pie. Boxpies are particularly suited to deal with polynomial constraints involving complex numbers. Section 7 recalls the definition of a contractor in the general framework. Section 8 provides an illustrative example related to robot localization, which is formalized with polynomial equations involving complex variables. In this example, the solution set is approximated by an inner and an outer subpaving made with boxpies. A conclusion is given in Section 9.

2 Intervals

Most interval methods, introduced by Moore [24] in his Ph.D. thesis, can be applied as soon as the set of domains for the variables has a lattice structure [11] as shown in [27]. A *lattice* $(\mathcal{E}, \leq)$ is a partially ordered set, closed under least upper and greatest lower bounds [11]. The least upper bound of x and y is called the *join* and is denoted by $x \vee y$. The greatest lower bound is called the *meet* and is written as $x \wedge y$.

Example 1. The set $(\mathbb{R}^n, \leq)$ is a lattice with respect to the partial order relation given by $\mathbf{x} \leq \mathbf{y} \Leftrightarrow \forall i \in \{1, \ldots, n\}, x_i \leq y_i$. We have $\mathbf{x} \wedge \mathbf{y} = (x_1 \wedge y_1, \ldots, x_n \wedge y_n)$, and $\mathbf{x} \vee \mathbf{y} = (x_1 \vee y_1, \ldots, x_n \vee y_n)$, where $x_i \wedge y_i = \min(x_i, y_i)$ and $x_i \vee y_i = \max(x_i, y_i)$.

Example 2. The set $(\mathbb{F}, \leq)$ of the functions which map $\mathbb{R}$ to $\mathbb{R}$ is a lattice with respect to the partial order relation given by $f \leq g \Leftrightarrow \forall t \in \mathbb{R}, f(t) \leq g(t)$. We have $f \wedge g : t \mapsto \min\{f(t), g(t)\}$, and $f \vee g : t \mapsto \max\{f(t), g(t)\}$.

Example 3. The set $\mathbb{IR}$ of closed intervals, as introduced by Moore [25], is a complete lattice with respect to the inclusion $\subset$. The meet corresponds to the intersection, and the join corresponds to the interval hull. For instance

$$[1,4] \wedge [2,\infty] = [2,4], \text{ and } [1,4] \vee [8,9] = [1,9]. \tag{1}$$

A lattice $\mathcal{E}$ is *complete* if for all (finite or infinite) subsets $\mathcal{A}$ of $\mathcal{E}$, the least upper bound $\wedge\mathcal{A}$ and the greatest lower bound $\vee\mathcal{A}$ belong to $\mathcal{E}$. When a lattice $\mathcal{E}$ is not complete, it is often possible to add two elements corresponding to $\wedge\mathcal{A}$ and $\vee\mathcal{A}$ to make it complete. For instance, the set $\mathbb{R}$ is not a complete lattice, whereas $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$ is. As a consequence, we have $\wedge\emptyset = \vee\mathcal{E}$ and $\vee\emptyset = \wedge\mathcal{E}$.

Intervals. A *closed interval* (or *interval* for short) $[x]$ of a complete lattice $\mathcal{E}$ is a subset of $\mathcal{E}$ which satisfies the equation $[x] = \{x \in \mathcal{E} \mid \wedge[x] \leq x \leq \vee[x]\}$. Both $\emptyset$ and $\mathcal{E}$ are intervals of $\mathcal{E}$. An interval is a sublattice of $\mathcal{E}$. If we denote by $\mathbb{I}\mathcal{E}$ the set of all intervals of a complete lattice $(\mathcal{E}, \leq)$, then $(\mathbb{I}\mathcal{E}, \subset)$ is also a lattice. For two elements $[x] = [x^-, x^+]$ and $[y] = [y^-, y^+]$ of $\mathbb{I}\mathcal{E}$, we have:

$$\begin{array}{rcl} [x] \wedge [y] & = & [x^- \vee y^-, x^+ \wedge y^+] \\ [x] \vee [y] & = & [x^- \wedge y^-, x^+ \vee y^+]. \end{array} \tag{2}$$

The meet $[x] \wedge [y]$ corresponds to the *intersection* and will denoted by $[x] \cap [y]$. The join $[x] \vee [y]$ is the *interval hull* and will be denoted by $[x] \sqcup [y]$.

Remark. In his book, Moore [25] considered intervals that are derived from the lattices $(\mathbb{R}^n, \leq)$. When $n > 1$, these intervals are named interval vectors. Moore also considered tubes, *i.e.*, intervals in the lattice of functions $(\mathbb{F}, \leq)$.

Width. The width function w associates to an interval $[x]$ a positive number. The width should satisfy the properties

$$\begin{array}{lll} \text{(i)} & [x] \subset [y] \Rightarrow w([x]) \leq w([y]) & \text{(monotonicity)} \\ \text{(ii)} & [x](k) \to a \Rightarrow w([x](k)) \to 0 & \text{(convergence).} \end{array} \tag{3}$$

The second property tells us that if a sequence of intervals $[x](k)$ converges to a point a (*i.e.*, a degenerated interval a which is a singleton), then the corresponding width converges to 0. This property requires that the sequence $[x](k)$ are intervals of a lattice $\mathcal{E}$, which is also a metric space. Moore defined the width of an interval of $\mathbb{R}$ as

$$w([x^-, x^+]) = x^+ - x^-, \tag{4}$$

which is consistent with this property.

Cartesian product. The Cartesian product of two lattices $(\mathcal{E}_1, \leq_1)$ and $(\mathcal{E}_2, \leq_2)$ is the lattice $(\mathcal{E}, \leq)$ defined as the set of all $(a_1, a_2) \in \mathcal{E}_1 \times \mathcal{E}_2$ with the order relation $(a_1, a_2) \leq (b_1, b_2) \Leftrightarrow ((a_1 \leq_1 b_1)$ and $(a_2 \leq_2 b_2))$. The intervals of $\mathcal{E}$ are made with the Cartesian product of the intervals of $\mathcal{E}_1$ and $\mathcal{E}_2$, *i.e.*, an interval $[x]$ of $\mathcal{E}$ can be written as

$$[x] = [x_1] \times [x_2] \text{ where } [x_1] \in \mathbb{I}\mathcal{E}_1 \text{ and } [x_2] \in \mathbb{I}\mathcal{E}_2. \tag{5}$$

Moreover, the width w in $\mathcal{E}$ can be derived from the width w_1 and w_2 in $\mathcal{E}_1$ and $\mathcal{E}_2$ as follows:

$$w\left([x_1] \times [x_2]\right) = \max\left(w_1\left([x_1]\right), w_2\left([x_2]\right)\right). \tag{6}$$

Moore's definition of the width of intervals of $\mathbb{R}^n$ is consistent with this definition.

Bisections. A *bisector* [8] is an operator that takes an interval $[x]$ as an input and which returns two intervals $[a]$ and $[b]$ such that *i)* $[a]$ and $[b]$ do not overlap; *ii)* $[x] = [a] \cup [b]$; and *iii)* $\max\left(w\left([a]\right), w\left([b]\right)\right)$ is minimal. This is the choice made by several optimization algorithms [16, 20] such as the Moore-Skelboe algorithm [34] or by algorithms for solving non-linear equations [6].

3 Bisectable Abstract Domains

All interval methods initiated by Moore as well as contractor-based tools can be generalized easily in the case where the unknown variables do not belong to a lattice [1]. What is important [9, 15, 31, 33] is that the domains that are handled form a lattice [11] with respect to the inclusion $\subset$. More precisely, consider a Riemannian manifold $\mathbb{M}$ (such a $\mathbb{R}$, $\mathbb{R}^n$ or a sphere). Since $\mathbb{M}$ is Riemannian, we can define the distance $d(a,b)$ between two points a and b as the minimal length than can be reached by any path connecting a to b. For any subset $\mathbb{X} \subset \mathbb{M}$, we can define the *diameter* (or *width*) $w(\mathbb{X})$ of $\mathbb{X}$ as the maximal distance $d(a,b)$ that exists between two points a and $b \in \mathbb{X}$. Denote by $\mathcal{P}(\mathbb{M})$ the powerset of $\mathbb{M}$. We define a family of *bisectable abstract domains* (*bad* for short) $\mathbb{IM}$ as a subset of $\mathcal{P}(\mathbb{M})$ which satisfies the following properties.

- $\mathbb{IM}$ is a Moore family[1]. This means that the intersection (not necessary finite) is closed in $\mathbb{IM}$, *i.e.,*

$$[a](1) \in \mathbb{IM}, [a](2) \in \mathbb{IM}, \ldots \;\Rightarrow \bigcap_i [a](i) \in \mathbb{IM}. \tag{7}$$

  From this property, we can deduce that $(\mathbb{IM}, \subset)$ is a lattice. However, this lattice is not necessary a sublattice of $\mathcal{P}(\mathbb{M})$. Indeed, even if the meet operator $\cap$ is preserved, the join operator in $\mathbb{IM}$ (denoted by $\sqcup$) is different from that in $\mathcal{P}(\mathbb{M})$ (denoted by $\cup$). More precisely, instead of an equality, we have the inclusion:

$$\underbrace{[a] \cup [b]}_{\in \mathcal{P}(\mathbb{M})} \subset \underbrace{[a] \sqcup [b]}_{\in \mathbb{IM}}. \tag{8}$$

- $\mathbb{IM}$ is equipped with a *bisector, i.e.,* a function $\beta : \mathbb{IM} \to \mathbb{IM} \times \mathbb{IM}$, such that $\beta([x]) = \{[a], [b]\}$ with the following properties: *i)* $[a]$ and $[b]$ do not overlap, *ii)* $[a]$ and $[b]$ cover $[x]$ and no other bisection consistent with *i)* and *ii)* generates a lower value for $\max\{w([a]), w([b])\}$.

[1] The Moore who gave the name to the Moore family is not the Ramon Moore who built the theory of interval analysis, but Eliakim Hastings Moore (1862–1932), who studied closure operators.

Cartesian product. Let $(\mathbb{IM}_1, \beta_1)$ and $(\mathbb{IM}_2, \beta_2)$ be two bads associated with the manifolds $\mathbb{M}_1$ and $\mathbb{M}_2$. A bad associated with the Cartesian product $\mathbb{M} = \mathbb{M}_1 \times \mathbb{M}_2$ is $(\mathbb{IM}, \beta)$ where

$$\begin{array}{lll} \mathbb{IM} & = & \mathbb{IM}_1 \times \mathbb{IM}_2 \\ \beta([m_1] \times [m_2]) & = & \begin{cases} \beta_1([m_1]) \times [m_2] & \text{if } w_1([m_1]) \geq w_2([m_2]) \\ [m_1] \times \beta_2([m_2]) & \text{otherwise.} \end{cases} \end{array} \tag{9}$$

This defines what we call the *Cartesian product between two bads*. It is useful to enclose vectors of variables. As defined by Moore, a box $[\mathbf{x}] = [x_1] \times [x_2]$ of $\mathbb{R}^2$ is a Cartesian product of two intervals of $\mathbb{R}$, which is a bad. A bisection of $[\mathbf{x}]$ can be defined as in (9) from the bisection of its interval components $[x_1]$ and $[x_2]$.

Reduced product [10]. Let $(\mathbb{IM}_1, \beta_1)$ and $(\mathbb{IM}_2, \beta_2)$ be two bads associated with the same manifold $\mathbb{M}$. We define the *reduced product* $(\mathbb{IM}, \beta) = (\mathbb{IM}_1, \beta_1) \otimes (\mathbb{IM}_2, \beta_2)$ as follows

$$\begin{array}{l} \mathbb{IM} = \{[m_1] \cap [m_2] \text{ such that } [m_1] \in \mathbb{IM}_1 \text{ and } [m_2] \in \mathbb{IM}_2\} \\ \beta([m_1] \cap [m_2]) = \begin{cases} \beta_1([m_1]) \cap [m_2] & \text{if } w([m_1]) \geq w([m_2]) \\ [m_1] \cap \beta_2([m_2]) & \text{otherwise.} \end{cases} \end{array} \tag{10}$$

The intersection is closed in $\mathbb{IM}$. Indeed, if $[a_1] \cap [a_2] \in \mathbb{IM}$ and $[b_1] \cap [b_2] \in \mathbb{IM}$, we have

$$[a_1] \cap [a_2] \cap [b_1] \cap [b_2] = \underbrace{[a_1] \cap [b_1]}_{\in \mathbb{IM}_1} \cap \underbrace{[a_2] \cap [b_2]}_{\in \mathbb{IM}_2}. \tag{11}$$

The idea of the reduced product, which is not well known by the interval community, is classically used in the community of abstract interpretation [10], where different types of domains are combined during the resolution. This is the case of an octagon [23], which corresponds to the intersection of a box with a rotated box.

4 Angles and Arcs

The notion of *bad* will now be illustrated in the case of angles, which do not have a lattice structure. Consider the equivalence relation on $\mathbb{R}$

$$\alpha \sim \beta \Leftrightarrow \frac{\beta - \alpha}{2\pi} \in \mathbb{Z}. \tag{12}$$

The set $\mathbb{A}$ of all angles corresponds to the quotient

$$\mathbb{A} = \frac{\mathbb{R}}{\sim} = \frac{\mathbb{R}}{2\pi\mathbb{N}}. \tag{13}$$

For simplicity, we will also write $\mathbb{A} = [-\pi, \pi[$. Note that the set $\mathbb{A}$ is a Riemannian manifold. Moreover, if α and β are angles and if $\rho \in \mathbb{R}$, we can define the operations $\alpha + \beta$, $\alpha - \beta$ and $\rho \cdot \alpha$. Due to its circular structure, the set of angles $\mathbb{A}$ is not a lattice. Thus, it is not possible to define intervals of angles to apply interval techniques [13]. Define an *arc* as a pair $\langle\alpha\rangle = \langle\overline{\alpha}, \widetilde{\alpha}\rangle$ such that $\overline{\alpha} \in \mathbb{A}$ and $\widetilde{\alpha} \in [0, \pi]$, where $\overline{\alpha}$ is called the *center* and $\widetilde{\alpha}$ is the *radius*. The set of all arcs is denoted by $\mathbb{IA}$. The intersection in $\mathbb{IA}$ is not closed, so $\mathbb{IA}$ is not a Moore family. To apply an interval approach on angles, it is necessary to take as a domains of angles: unions of arcs, which corresponds to the

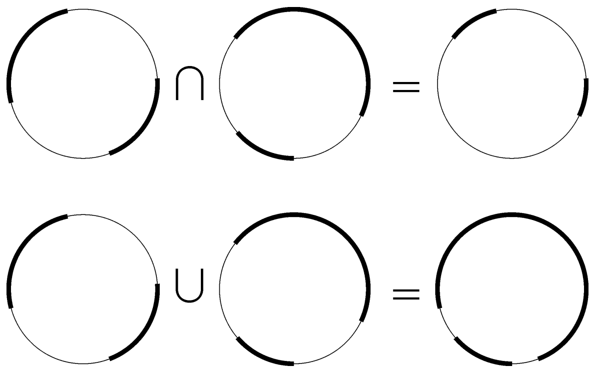

Figure 1: Intersection and union of two circular pavings

smallest Moore family which contains $\mathbb{IA}$. A union of non-overlapping arcs is called a *circular paving*. The set of circular pavings is denoted by $\mathbb{UA}$ and $(\mathbb{UA}, \subset)$. Figure 1 illustrates the intersection and the union of circular pavings.

Remark. In practice, we should limit the number of intervals inside a union. It is indeed possible to find examples where the number of intervals to represent a domain of $\mathbb{A}$ increases to infinity during the propagation. This phenomenon exists as soon as we work with union of intervals. In his thesis, Chabert proposes such an example (see [7], Example 6.5 page 152) for domains of $\mathbb{R}$. The constraints are

$$\begin{cases} y = x \\ 9\,(x-5)^2 = 16y, \end{cases} \tag{14}$$

and the initial domains are $[x] = [y] = [1, 9]$. This example can easily be adapted to illustrate an explosion with unions of arcs by setting $x = 3\alpha$ and $y = 3\beta$. We get

$$\begin{cases} \beta = \alpha \\ 9\,(3\alpha-5)^2 = 48\beta \end{cases} \tag{15}$$

with initial domains $[\alpha] = [\beta] = \left[\frac{1}{3}, 3\right]$. If we start a propagation with an interval of angles, we get the same type of explosion as the one observed by Chabert.

5 Pies

In the previous section, we were able to define a family of domains (the circular paving) for angles which is a bad. Since the Cartesian product of bads is a bad, we can thus easily define a bad associated to a finite set of variables. This is what it is done when Moore has defined boxes of $\mathbb{R}^n$ as Cartesian products of intervals. We now illustrate this by considering an angle variable α and a scalar variable $\rho > 0$. If α belongs to the circular paving $< \alpha >$ and ρ belongs to the scalar interval $[\rho]$ then the pair (α, ρ) belongs to $< \alpha > \times [\rho]$ which is called a *pie*. More formally, a *pie* is an element of $\mathbb{UA} \times \mathbb{IR}$. A pie can also be interpreted as a subset of $\mathbb{R}^2$ as illustrated by Figure 2 right, which shows a pie $\langle \overline{\alpha}, \widetilde{\alpha} \rangle \times \left[\rho^-, \rho^+\right]$ with a single connected component. A pie will often be denoted in its polar form as $[\rho]\, e^{i<\alpha>}$. Note that, because the set of pies is a bad, the intersection between pies is always a pie. Indeed, if $[\rho_1]\, e^{i<\theta_1>}$ and $[\rho_2]\, e^{i<\theta_2>}$ are two pies, we have

$$[\rho_1]\, e^{i<\theta_1>} \cap [\rho_2]\, e^{i<\theta_2>} \;=\; ([\rho_1] \cap [\rho_2])\, e^{i(<\theta_1> \cap <\theta_2>)}. \tag{16}$$

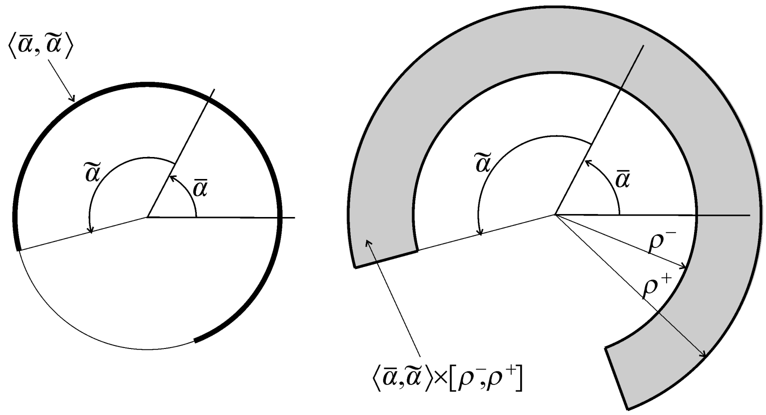

Figure 2: Left: an arc; Right: a pie

6 Boxpies

Consider the set $\mathbb{C}$ of complex numbers. Two bads could be considered: boxes of $\mathbb{C}$ of the form $[x] + i\,[y]$ and pies of $\mathbb{C}$ of the form $[\rho]\, e^{i<\theta>}$. Both $\mathbb{IC}$ (the boxes) and $\mathbb{UA} \times \mathbb{IR}$ (the pies) are Moore families in $\mathcal{P}(\mathbb{C})$. The union $\mathbb{IC}$ with $\mathbb{UA} \times \mathbb{IR}$ is not a Moore family anymore, but we can define the smallest Moore family $\mathbb{BP}$ of $\mathcal{P}(\mathbb{C})$ which contains both $\mathbb{IC}$ and $\mathbb{UA} \times \mathbb{IR}$. This corresponds to the reduced product $\otimes$ [10] presented in Section 3. Therefore, we can write $\mathbb{BP} = \mathbb{IC} \otimes\ \mathbb{UA} \times \mathbb{IR}$. The family $\mathbb{BP}$ contains boxes and pies, but it also contains all intersections between one box and one pie. An element of $\mathbb{BP}$ is called a *boxpie*. Thus, a boxpie can be written as

$$[x] + i\,[y] \;\;\cap\;\; [\rho]\, e^{i<\theta>}. \tag{17}$$

The intersection between two boxpies is also a boxpie. Indeed:

$$\begin{aligned} & [x_1] + i\,[y_1] \;\;\cap\;\; [\rho_1]\, e^{i<\theta_1>} \;\;\cap\;\; [x_2] + i\,[y_2] \cap [\rho_2]\, e^{i<\theta_2>} \\ = \;\; & [x_1] \cap [x_2] + i\,([y_1] \cap [y_2]) \cap ([\rho_1] \cap [\rho_2])\, e^{i(<\theta_1> \cap <\theta_2>)}. \end{aligned} \tag{18}$$

An arithmetic on boxpies inherits not only the good properties of interval arithmetic for the addition, but also the good properties of pie arithmetic [30] for the multiplication.

Self-consistency. The expression for a boxpie may not be unique. For instance, the boxpie

$$[0, 1] + i\,[1, 2] \cap [1, 2] \cdot e^{i[0, \frac{\pi}{4}]} = [1, 1] + i\,[1, 1] \cap [\sqrt{2}, \sqrt{2}] e^{i\left[\frac{\pi}{4}, \frac{\pi}{4}\right]} \tag{19}$$

is a singleton which contains as a single element, the complex number $1 + i = \sqrt{2} e^{i\frac{\pi}{4}}$. The representation which is minimal with respect to the inclusion of intervals is said to be *self-consistent*.

7 Contractors

Many problems of estimation, control or robotics can be represented by *constraint networks* [18]. A constraint networks (see, *e.g.,* [36, 37]) is composed of a set of variables $\{x_1, \ldots, x_n\}$, a set of constraints $\{c_1, \ldots, c_m\}$ and a set of domains $\{\mathbb{X}_1, \ldots, \mathbb{X}_n\}$. The domains $\mathbb{X}_i$ should belong to a complete lattice $(\mathcal{L}_i, \subset)$. In the interval literature derived from Moore's work, the domains for the variables of a constraint networks are intervals. It is not the case when dealing with finite domains. The interval nature is not needed as soon as the set of domains has a structure of lattice. In the context of this paper, the sets $\mathcal{L}_i$ will correspond to the set of boxpies $\mathbb{BP}$. Denote by $\mathcal{L}$ the Cartesian product of all $\mathcal{L}_i$'s, *i.e.,* $\mathcal{L} = \mathcal{L}_1 \times \cdots \times \mathcal{L}_n$. An element $\mathbb{X}$ of $\mathcal{L}$ is the Cartesian product of n elements of $\mathcal{L}_i$, (*i.e.,* it satisfies $\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_n$). A *contractor* (see *e.g.,* [3]) is an operator

$$\mathcal{C} : \begin{array}{ccc} \mathcal{L} & \rightarrow & \mathcal{L} \\ \mathbb{X} & \mapsto & \mathcal{C}(\mathbb{X}), \end{array} \tag{20}$$

which satisfies

$$\begin{array}{ll} \mathbb{X} \subset \mathbb{Y} \Rightarrow \mathcal{C}(\mathbb{X}) \subset \mathcal{C}(\mathbb{Y}) & \text{(monotonicity)} \\ \mathcal{C}(\mathbb{X}) \subset \mathbb{X} & \text{(contractance).} \end{array} \tag{21}$$

The set of contractors forms also a complete lattice. As a consequence, the meet (or intersection) and join (or union) can also be defined. This leads us to the contractor algebra [8]. When all variables of the constraint networks belong to $\mathbb{R}$, contractor techniques have been shown to be very powerful [2, 35].

Remark. The interval Newton operator developed by Moore [25] also aims at contracting boxes without removing any point from the solution set. Now, this operator is not monotonic. For a counter-example, take $f(x) = e^x - 1$. The associated Newton operator is

$$\mathcal{N}([x]) = \left(x_0 - \frac{f(x_0)}{f'([x])} \right) \cap [x] = \left(x_0 - \frac{e^{x_0} - 1}{e^{[x]}} \right) \cap [x], \tag{22}$$

where x_0 is the center of $[x]$. Take $[a] = [0, 1]$ and $[b] = [-1, 1]$. We have

$$\mathcal{N}([a]) = \mathcal{N}([0, 1]) = \left(\frac{1}{2} - \frac{e^{\frac{1}{2}} - 1}{e^{[0,1]}} \right) \cap [0, 1] \simeq [0, 0.26135] \tag{23}$$

$$\mathcal{N}([b]) = \mathcal{N}([-1, 1]) = \left(0 - \frac{0}{e^{[-1,1]}} \right) \cap [-1, 1] = 0. \tag{24}$$

Thus,

$$[a] \subset [b] \text{ does not imply that } \mathcal{N}([a]) \subset \mathcal{N}([b]). \tag{25}$$

Due to this non-monotonicity, the Newton operator does not satisfy the definition of a contractor.

Constraint propagation. In principle, we associate to each constraint $c_j \in \{c_1, \ldots, c_m\}$ of a constraint network, a contractor $\mathcal{C}_j(\mathbb{X})$ which does not remove any $(x_1, \ldots, x_n)$, and which is consistent with c_j. Then, we build the contractor $\mathcal{C} = \mathcal{C}_1 \circ \cdots \circ \mathcal{C}_m$. We apply the contractor $\mathcal{C}$ until no more contraction can be observed. From Tarski's Theorem, we conclude that the process converges toward the largest subdomain $\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_n$ of the initial domain which cannot be contracted by any $\mathcal{C}_i$.

Contractors. Most contractors are built on an arithmetic of domains (which corresponds to the interval arithmetic if these domains are intervals). If $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$

are pies containing the three complex numbers a, b and c, using the arithmetic proposed in [5], it is possible to define efficient contractors associated with the constraints

$$a + b = c \text{ and } c = a \cdot b. \tag{26}$$

These contractors can be used for solving polynomial equations in $\mathbb{C}$. Moreover, due to the non-unicity of the expression of a boxpie, it is important to add a self-consistent contractor to have better contractions.

Separators. A separator [17] is a pair complementary contractors. Combined with a paver, separators make it possible to compute an inner and an outer characterization of the solution set. The principle is similar to what has been proposed by Moore [26] and successors (see, *e.g.*, [14, 19, 28] to characterize an inner and an outer approximations of a set defined by inequalities. The main difference is that Moore used inclusion tests, whereas here, we use separators for efficiency. As shown in [17], from a contractor, it often is possible to get the corresponding separator automatically.

8 Application to Robot Localization

A robot, moving in a plane, is able to see a landmark $\mathbf{m}$ with coordinates $(10, 12)$. More precisely, a sensor in the robot is able to measure the distance d and the azimuth α of $\mathbf{m}$ with a known accuracy. Assume for instance that we collected $\alpha \in [\frac{\pi}{12}, \frac{\pi}{6}]$ and $d \in [4, 6]$. Assume that the position for the robot is known to belong to the box $[3, 8] \times [6, 13]$. Let us represent the position of the robot by a complex number $p \in \mathbb{C}$. We have to solve:

$$10 + 12i - p = de^{i\alpha}, p \in [3, 8] \times [6, 13],\ \alpha \in \left[\frac{\pi}{12}, \frac{\pi}{6}\right],\ d \in [4, 6]. \tag{27}$$

The first contraction yields the boxpie represented in bold in Figure 3, left. In this figure, the black triangle which corresponds to the unknown true position for the robot in $(6, 10)$. A paver is able to give the inner and the outer characterization represented in Figure 3, right.

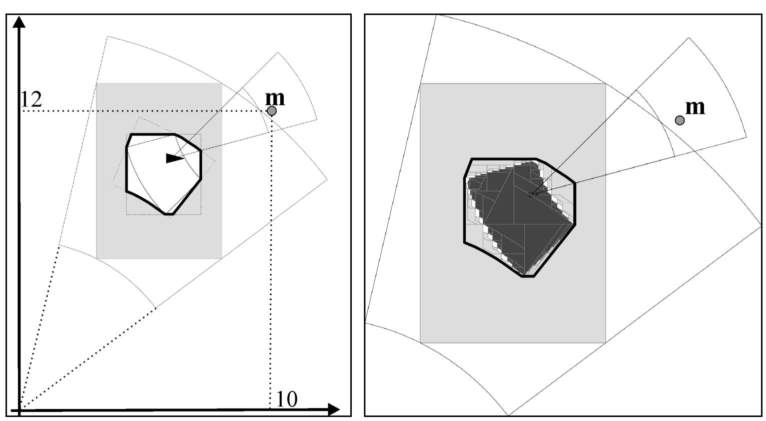

Figure 3: Left: boxpie enclosing the true position of the robot, represented by the black triangle; Right: Inner and outer approximation of the solution set

As a comparison, Figure 4 provides the pavings obtained using boxes and pies as domains, but in a separate way. To get these figures, we used specific minimal separators for the projection of the set

$$\{(x, y, \rho, \theta) \mid x = \rho \cos\theta \text{ and } y = \rho \sin\theta\} \tag{28}$$

with respect to the (x, y) and (ρ, θ) space [12].

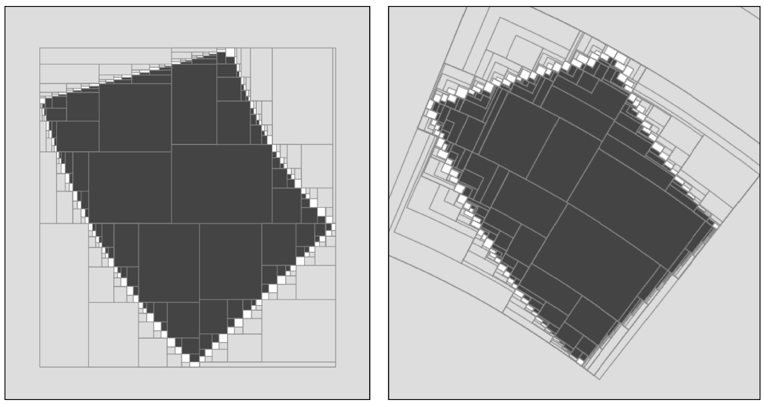

Figure 4: Left: paving obtained using boxes only; Right: paving obtained using pies only

Boxpies belong to the class of redundant computation methods [36] and cannot be seen as a new branch-and-prune framework. Contrary to what is done when we add redundant constraints for more efficiency, we combine here the different types of domains to reduce the wrapping effect. As illustrated by Figures 3 and 4, even when minimal contractors are available (or equivalently, when we are able to compute the global consistency), the best approximation we can get by a single domain (*i.e.*, without bisections) is always affected by wrapping effect. With boxpies, this wrapping effect still exists (see Figure 3) ,but its influence is smaller than with boxes only (Figure 4, left) or pies only (Figure 4, right).

9 Conclusion

This paper shows that the interval arithmetic introduced by Moore can be generalized to other types of domains as soon as these domains form a lattice with respect to the inclusion and that we could bisect them. This allowed us to introduce a new category of domains, named *bisectable abstract domain* (bad for short). The bisectable property makes a difference with the domains classically considered in domain theory [32] where the bisection is not considered. An example of *bad* is the boxpie which corresponds to the intersection between one box and one pie. Most of interval-based algorithms can be extended easily to this type of domains, since we are able to contract boxpies with respect to some constraints and to bisect them. Boxpies are particularly interesting when we deal with equations in $\mathbb{C}$ since they inherit the accuracy of the Cartesian representation for the addition and the accuracy of the polar representation for the multiplication.

References

[1] K. Apt. The essence of constraint propagation. *Theoretical Computer Science*, 221(1-2):179–210, 1998.

[2] I. Araya, G. Trombettoni, and B. Neveu. A contractor based on convex interval Taylor. In *Proc. of CPAIOR*, pages 1–16, LNCS 7298, Springer, 2012.

[3] F. Benhamou, F. Goualard, L. Granvilliers, and J-F. Puget. Revising Hull and Box Consistency. In *ICLP*, pages 230–244, 1999.

[4] R. E. Boche. Complex interval arithmetic with some applications. *Technical Report Report LMSC4-22-66-1, Lockheed Missiles and Space*, 1965.

[5] Y. Candau, T. Raissi, N. Ramdani, and L. Ibos. Complex interval arithmetic using polar form. *Reliable Computing, Springer Verlag (Germany)*, 20(1):1–20, 2006.

[6] M. Ceberio and L. Granvilliers. Solving nonlinear systems by constraint inversion and interval arithmetic. In *Artificial Intelligence and Symbolic Computation*, volume 1930, pages 127–141, LNCS 5202, 2001.

[7] G. Chabert. *Techniques d'intervalles pour la résolution de systèmes d'équations.* PhD dissertation, University of Nice–Sophia Antipolis, Nice, France, 2007.

[8] G. Chabert and L. Jaulin. Contractor programming. *Artificial Intelligence*, 173:1079–1100, 2009.

[9] P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages*, pages 238–252, Los Angeles, California, 1977.

[10] P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Conference Record of the Sixth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 269–282, San Antonio, Texas, 1979. ACM Press, New York, NY.

[11] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order.* Cambridge University Press, (ISBN 0521784514), 2002.

[12] B. Desrochers and L. Jaulin. A minimal contractor for the polar equation; application to robot localization. *Engineering Applications of Artificial Intelligence (submitted)*, 2015.

[13] I. Gargantini and P. Henrici. Circular arithmetic and the determination of polynomial zeros. *Numer. Math.*, 18:305–320, 1972.

[14] A. Goldsztejn and L. Jaulin. Inner approximation of the range of vector-valued functions. *Reliable Computing*, pages 1–23, 2010.

[15] E. Goubault and S. Putot. Static analysis of numerical algorithms. In *Proceedings of SAS 2006, LNCS 4134*, pages 18–34. Springer-Verlag, 2006.

[16] E. R. Hansen. *Global Optimization using Interval Analysis.* Marcel Dekker, New York, NY, 1992.

[17] 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.

[18] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.

[19] L. Jaulin and E. Walter. Guaranteed nonlinear parameter estimation via interval computations. *Interval Computation*, pages 61–75, 1993.

[20] R. B. Kearfott and V. Kreinovich, editors. *Applications of Interval Computations*. Kluwer, Dordrecht, the Netherlands, 1996.

[21] R. Klatte and C. Ullrich. Complex sector arithmetic. *Computing*, 24(2):139–148, 1980.

[22] V. Kreinovich and J. Wolfe von Gudenberg. Arithmetic of complex sets: Nickel's classical paper revisited from a geometric viewpoint. *Geombinatorics*, 9(1):21–26, 1999.

[23] A. Miné. The octagon abstract domain. *Higher Order Symbol. Comput.*, 19(1):31–100, 2006.

[24] R. E. Moore. *Interval Arithmetic and Automatic Error Analysis in Digital Computing*. PhD dissertation, Department of Mathematics, Stanford University, Stanford, CA, 1962.

[25] R. E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966.

[26] R. E. Moore. Parameter sets for bounded-error data. *Mathematics and Computers in Simulation*, 34(2):113–119, 1992.

[27] R. E. Moore and M. J. Cloud. *Computational Functional Analysis*. Horwood Publishing, London, 2006.

[28] O. Mullier, E. Goubault, M. Kieffer, and S. Putot. General inner approximation of vector-valued functions. *Reliable Computing*, 18(0):117–143, 2013.

[29] K. Nickel. Arithmetic of complex sets. *Computing*, 24(0):97–105, 1980.

[30] M. Petkovic and L. Petkovic. *Complex Interval Arithmetic and Its Applications*. John Wiley & Sons, Mathematics, 1998.

[31] E. S. Petrov and T. M. Yakhno. Subdefinite models and logic programming: Implementation of constraints. *Programming and Computer Software*, 27(2):95–100, 2001.

[32] D. S. Scott. Data types as lattices. In Lecture Notes in Mathematics, editor, *Proceedings of the International Summer Institute and Logic Colloquium*, pages 579–651. Springer-Verlag, 1975.

[33] I. Shvetsov, A. Semenov, and V. Telerman. Application of subdefinite models in engineering. *Artificial Intelligence in Engineering*, 11(1):15–24, 1997.

[34] S. Skelboe. Computation of rational interval functions. *BIT*, 14:87–95, 1974.

[35] G. Trombettoni and G. Chabert. Constructive interval disjunction. In *Proc. CP, Constraint Programming*, pages 635–650, LNCS 4741, 2007.

[36] M. van Emden. Algorithmic power from declarative use of redundant constraints. *Constraints*, 4(4):363–381, 1999.

[37] P. van Hentenryck, L. Michel, and Y. Deville. *Numerica - A Modelling Language for Global Optimization*. MIT Press, Cambridge, Massachusetts, 1997.