Chapter 1

Guaranteed numerical injectivity test via interval analysis

1.1. Introduction

The purpose of this paper is to present a new method based on guaranteed numerical computation able to verify that a function $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}^m$ satisfies

$$\forall x_1 \in \mathcal{X}, \forall x_2 \in \mathcal{X}, x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2). \tag{1.1}$$

To our knowledge, it does not exist any numerical method able to perform this injectivity test and moreover, the complexity of the algebraic manipulations involved often makes formal calculus in fault (especially when the function is not polynomial). Presently, in the context on structural identifiability, Braems and al. have presented in [BRA 01] an approximated method that verifies the injectivity around ε namely ε-injectivity. It consists in verifying the following condition

$$\forall x_1 \in \mathcal{X}, \forall x_2 \in \mathcal{X}, |x_1 - x_2| > \varepsilon \Rightarrow f(x_1) \neq f(x_2), \tag{1.2}$$

which can be view as an approximation of the condition (1.1).

Note that, many problems could be formulated as the injectivity verification of a specific function. For example, concerning the identification of parametric models, the problem of proving the *structural identifiability* amounts to check injectivity [E.W 90, BRA 01]. Other applications can be cited: For instance, consider the robotic arm with two degrees of freedom ($\theta_1 \in [0, \frac{\pi}{2}], \theta_2 \in [-\pi, \pi]$) represented in the Figure

---

Chapter written by Sébastien LAGRANGE, Nicolas DELANOUE and Luc JAULIN.

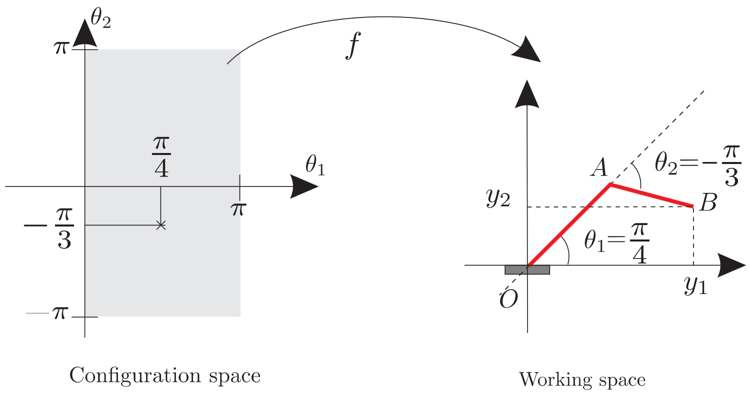

Figure 1.1. *A point in the configuration space and its corresponding robot configuration.*

1.1(right). Each point (θ_1, θ_2) of the *configuration space* is associated with a robot position (y_1, y_2) by the function

$$f : (\theta_1, \theta_2) \to \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 2cos(\theta_1) + 1.5cos(\theta_1 + \theta_2) \\ 2sin(\theta_1) + 1.5sin(\theta_1 + \theta_2) \end{pmatrix} \tag{1.3}$$

(See Figure 1.1). Now, a basic question is to known whether several pairs (θ_1, θ_2) lead to identical position (y_1, y_2) of the robot ending. This problem amounts to test the function f (defined in (1.3)) for injectivity.

This paper provides an efficient algorithm, based on interval analysis, able to check that a differentiable function is injective. The paper is organized as follows. Section 1.2 presents interval analysis that will be used. In Section 1.3, a new definition of partial injectivity makes possible the use of interval analysis techniques to test injectivity and to get a guaranteed answer. Section 1.4 presents an algorithm able to test a given differentiable function for injectivity. Finally, in order to show the efficiency of the algorithm, two illustrative examples are provided. A solver called ITVIA (Injectivity Test Via Interval Analysis) implemented in C++ is made available at `http://www.istia.univ-angers.fr/~lagrange/`.

1.2. Interval analysis

This section introduces some notions of interval analysis to be used in this paper. A vector interval or a *box* $[\mathbf{x}]$ of $\mathbb{R}^n$ is defined by

$$[x] = [\underline{x}, \overline{x}] = \{x \in \mathbb{R}^n \mid \underline{x} \le x \le \overline{x}\}, \tag{1.4}$$

where $\underline{x}$ and $\overline{x}$ are two elements of $\mathbb{R}^n$ and the partial order $\le$ is understood componentwise. The set of all bounded boxes of $\mathbb{R}^n$ is denoted by $\mathbb{IR}^n$ as in [JAU 01].

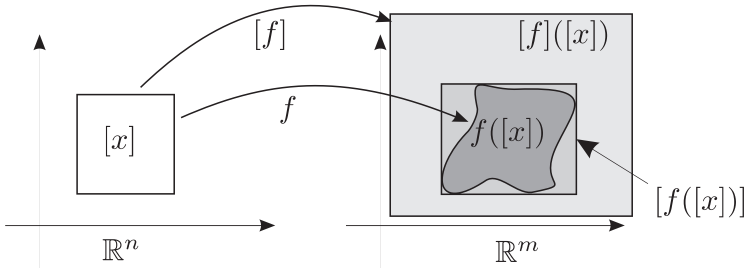

Figure 1.2. *Inclusion function $[f]$ of a function f.*

Remarque 1 *By extension, one defines an* interval matrix $[M] = [\underline{M}, \overline{M}]$ *as the set of the matrices of the form :*

$$[M] = \{M \in \mathbb{R}^{n\times m} \mid \underline{M} \leq M \leq \overline{M}\} \tag{1.5}$$

and $\mathbb{IR}^{n\times m}$ denoted the set of all interval matrices of $\mathbb{R}^{n\times m}$. The properties of punctual matrices can naturally be extended to interval matrices. For example, $[M]$ is full column rank if all the matrices $M \in [M]$ are full column rank.

To *bisect* a box $[x]$ means to cut it along a symmetry plane normal to a side of maximal length. The length of this side is the *width* of $[x]$. A bisection of $[x]$ generates two non overlapping boxes $[x_1]$ and $[x_2]$ such that $[x] = [x_1] \cup [x_2]$. The *hull box* $[\mathcal{X}]$ of a bounded subset $\mathcal{X} \in \mathbb{R}^n$ is the smallest box of $\mathbb{IR}^n$ that contains $\mathcal{X}$.

Interval arithmetic defined in [MOO 66] provides an effective method to extend all concepts of vector arithmetic to boxes.
Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a vector function; the set-valued function $[f] : \mathbb{IR}^n \to \mathbb{IR}^m$ is a *inclusion function* of f if, for any box $[x]$ of $\mathbb{IR}^n$, it satisfies $f([x]) \subset [f]([x])$ (see Figure 1.2). Note that $f([x])$ is usually not a box contrary to $[f]([x])$. Moreover, since $[f([x])]$ is the hull box of $f([x])$, one has

$$f([x]) \subset [f([x])] \subset [f]([x]). \tag{1.6}$$

The computation of an inclusion function $[f]$ for any analytical function f can be obtained by replacing each elementary operator and function by its interval counterpart [MOO 66, NEU 90].

Example 2 *An inclusion function for $f(x_1, x_2) = x_1^2 + \cos(x_1 x_2)$ is $[f]([x_1], [x_2]) = [x_1]^2 + \cos([x_1][x_2])$. For instance, if $[x] = ([-1, 1], [0, \frac{\pi}{2}])$ then the box $[f]([x])$ is computed as follows:*

$$\begin{array}{llll} [f]([-1,1],[0,\frac{\pi}{2}]) & = [-1,1]^2 + \cos([-1,1]\times[0,\frac{\pi}{2}]) & = [0,1] + \cos([-\frac{\pi}{2},\frac{\pi}{2}]) \\ & = [0,1] + [-1,1] & = [-1,2]. \end{array}$$

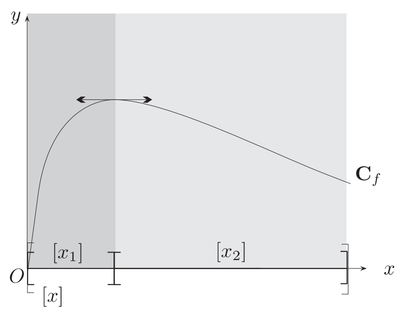

Figure 1.3. *Despite the fact that $f_{|[x_1]}$ and $f_{|[x_2]}$ are injection, f is not an injection.*

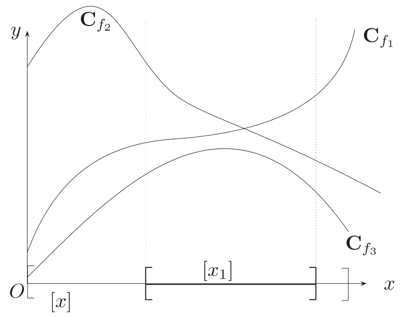

Figure 1.4. *Graphs of functions f_1, f_2 and f_3.*

1.3. Injectivity

Recall that this paper proposed to build a effective method to test differentiable function $f : \mathcal{X} \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ for injectivity. The main idea of the algorithm to be proposed is to divide $\mathcal{X}$ into subsets $\mathcal{X}_i$ where f restricted to $\mathcal{X}_i$ (denoted $f_{|\mathcal{X}_i}$) is an injection. However, as illustrated in Figure 1.3, the injectivity is not stable by union *i.e.*

$$\left(f_{|\mathcal{X}_1} \text{ is an injection and } f_{|\mathcal{X}_2} \text{ is an injection}\right) \nRightarrow f_{|\mathcal{X}_1 \cup \mathcal{X}_2} \text{ is an injection.}$$

Thus, the injectivity cannot directly be used. That why we are going to consider a concept akin to injectivity, namely *the partial injectivity*, that will be stable by union. First, we introduce the definition of the partial injectivity and give some illustrative examples. Then, we propose theorem which give a sufficient condition to test function for partial injectivity. This section presents the fundamental results that we will be used in the algorithm able to test function for injectivity.

1.3.1. *Partial Injectivity*

Let us introduce the definition of *partial injectivity* of a function.

Definition 1 *Consider a function $f : \mathcal{X} \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ and any set $\mathcal{X}_1 \subset \mathcal{X}$. The function f is a partial injection of $\mathcal{X}_1$ over $\mathcal{X}$, noted $(\mathcal{X}_1, \mathcal{X})$-injective, if $\forall x_1 \in \mathcal{X}_1, \forall x \in \mathcal{X}$,*

$$x_1 \neq x \Rightarrow f(x_1) \neq f(x). \tag{1.7}$$

f is said to be $\mathcal{X}$-injective if it is $(\mathcal{X}, \mathcal{X})$-injective.

Example 3 *Consider the three functions of Figure 1.4. The functions f_1 and f_2 are $([x_1], [x])$-injective (although f_2 is not $[x]$-injective) whereas f_3 is not.*

The following proposition will motivate the implementation of the algorithm presented in Section 1.4.

Proposition 4 *Consider a function $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}^m$ and $\mathcal{X}_1, \ldots, \mathcal{X}_p$ a collection of subsets of $\mathcal{X}$. We have*

$$\forall i, 1 \leq i \leq p, f \; is \; (\mathcal{X}_i, \mathcal{X}) - injective \; \Leftrightarrow f \; is \; (\bigcup_{i=1}^{p} \mathcal{X}_i, \mathcal{X}) - injective. \tag{1.8}$$

Proof. ($\Rightarrow$) One has $\forall x_i \in \mathcal{X}_i, \forall x \in \mathcal{X}$, $x_i \neq x \Rightarrow f(x_i) \neq f(x)$. Hence $\forall \check{x} \in (\cup_i \mathcal{X}_i), \forall x \in \mathcal{X}$, $\check{x} \neq x \Rightarrow f(\check{x}) \neq f(x)$. ($\Leftarrow$) Trivial. ∎

1.3.2. *Partial Injectivity Condition*

In this paragraph, a fundamental theorem, which gives a sufficient condition of partial injectivity, is presented. First, let us introduce a generalization of the Mean Value Theorem[1].

Theorem 5 (Generalized Mean Value Theorem) *Consider a differentiable function $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}^m$. Let ∇f be its Jacobian matrix and $[x] \subset \mathcal{X}$. One has*

$$\forall x_1, x_2 \in [x], \exists J_f \in [\nabla f([x])] \; such \; that \; f(x_2) - f(x_1) = J_f \cdot (x_2 - x_1), \tag{1.9}$$

where $[\nabla f([x])]$ denotes the hull box of $\nabla f([x])$.

Proof. According to Mean-Value Theorem applied on each components $f_i : \mathbb{R}^n \to \mathbb{R}$ of f ($1 \leq i \leq m$) and since the segment $seg(x_1, x_2)$ belongs to $[x]$, we have

$$\exists \xi_i \in [x] \; such \; that \; f_i(x_2) - f_i(x_1) = \nabla f_i(\xi_i) \cdot (x_2 - x_1). \tag{1.10}$$

Taking $J_{f_i} = \nabla f_i(\xi_i)$, we get

$$\exists J_{f_i} \in \nabla f_i([x]) \; such \; that \; f_i(x_2) - f_i(x_1) = J_{f_i} \cdot (x_2 - x_1). \tag{1.11}$$

1. Let $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$, $f \in \mathcal{C}^1$. If $x_1, x_2 \in \mathcal{X}$ such that the segment between x_1 and x_2, noted $seg(x_1, x_2)$, is included in $\mathcal{X}$. Then, there exists $\xi \in seg(x_1, x_2)$ such that

$$f(x_2) - f(x_1) = \nabla f(\xi) \cdot (x_2 - x_1).$$

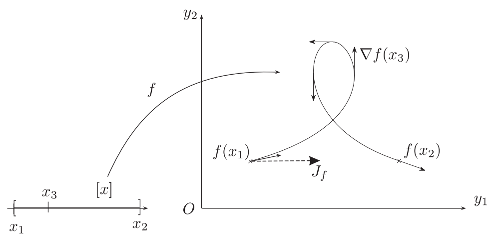

Figure 1.5. *Graph of* $f : \mathbb{R} \to \mathbb{R}^2$.

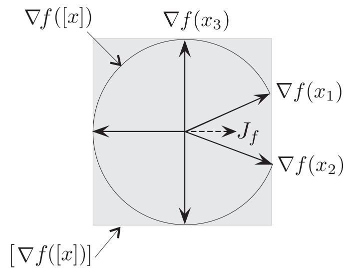

Figure 1.6. *Illustration of the set* $[\nabla f([x])]$.

Thus

$$\exists J_f \in (\nabla f_1\left([x]\right), \ldots, \nabla f_m\left([x]\right))^T \ \textit{such that}\ f(x_2) - f(x_1) = J_f \cdot (x_2 - x_1). \tag{1.12}$$

i.e., since $(\nabla f_1\left([x]\right), \ldots, \nabla f_m\left([x]\right))^T \subset [\nabla f([x])]$ (see (1.6)),

$$\exists J_f \in [\nabla f([x])] \ \textit{such that}\ f(x_2) - f(x_1) = J_f \cdot (x_2 - x_1). \ \blacksquare$$

Example 6 *Consider the function*

$$f : \left\{ \begin{array}{ccc} \mathbb{R} & \to & \mathbb{R}^2 \\ x & \to & (y_1, y_2)^T \end{array} \right. . \tag{1.13}$$

depicted in Figure 1.5. Figure 1.6 represents the set $\nabla f([x])$ *of all derivatives of* f *(drawn as vectors) and its hull box* $[\nabla f([x])]$*. One can see that the vector* J_f *defined in (1.9) belongs to* $[\nabla f\left([x]\right)]$ *(but* $J_f \notin \nabla f\left([x]\right)$*) as forecasted by Theorem 5.*

Now, the following theorem introduces a sufficient condition of partial injectivity. This condition will be exploited in next section in order to design a suitable algorithm that test injectivity.

Theorem 7 *Let* $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}^m$ *be a differentiable function and* $[x_1] \subset [x] \subset \mathcal{X}$*. Set* $[\tilde{x}] = \left[f^{-1}\left(f\left([x_1]\right)\right) \cap [x]\right]$*. If the interval matrix* $[\nabla f\left([\tilde{x}]\right)]$ *is full column rank then* f *is* $([x_1], [x])$*-injective.*

Proof. The proof is by contradiction. Assume that f is not $([x_1],[x])$-injective then

$$\exists x_1 \in [x_1], \exists x_2 \in [x] \ \textit{such that}\ x_1 \neq x_2 \ \textit{and}\ f\left(x_1\right) = f\left(x_2\right). \tag{1.14}$$

Now, since $f(x_1) = f(x_2)$, one has $x_2 \in f^{-1}\left(f\left([x_1]\right)\right) \cap [x]$ and trivially $x_1 \in f^{-1}\left(f\left([x_1]\right)\right) \cap [x]$. Therefore, since $\left(f^{-1}\left(f\left([x_1]\right)\right) \cap [x]\right) \subset \left[f^{-1}\left(f\left([x_1]\right)\right) \cap [x]\right] =$

$[\tilde{x}]$ (see Equation (1.6)), one has $x_1, x_2 \in [\tilde{x}]$.
Hence, (1.14) implies

$$\exists x_1, x_2 \in [\tilde{x}]\,,\ such\ that\ x_2 \neq x_1\ and\ f\left(x_1\right) = f\left(x_2\right). \tag{1.15}$$

To conclude, according to Theorem 5, $\exists x_1, x_2 \in [\tilde{x}]$, $\exists J_f \in [\nabla f\left([\tilde{x}]\right)]$ such that

$$x_1 \neq x_2\ and\ 0 = f(x_2) - f(x_1) = J_f \cdot (x_2 - x_1), \tag{1.16}$$

i.e. $\exists J_f \in [\nabla f\left([\tilde{x}]\right)]$ such that J_f is not full column rank and therefore the (interval) matrix $[\nabla f\left([\tilde{x}]\right)]$ is not full column rank. ■

1.4. ITVIA Algorithm

This section presents the Injectivity Test Via Interval Analysis (ITVIA) algorithm designed from Propposition 4 and Theorem 7. ITVIA uses the divide and conquer strategy to check a given differentiable function $f : [x] \subset \mathbb{R}^n \to \mathbb{R}^m$ for injectivity. It can be decomposed in two distinct sub-algorithms :
- Algorithm 1 checks if the interval matrix $\left[\nabla f\left(\left[f^{-1}\left(f\left([x_1]\right)\right) \cap [x]\right]\right)\right]$ is full column rank. In the positive case, according to Theorem 7, the function f is $([x_1], [x])$-injective. Therefore, Algorithm 1 can be viewed as a test for partial injectivity.
- Algorithm 2 divides the initial box $[x]$ into a paving $\{[x_i]\}_i$ such that, for all i, the function f is $([x_i], [x])$-injective. Then, since $[x] = (\cup_i [x_i])$ and according to Proposition 4, f is $[x]$-injective.

In Algorithm 1, a set inversion technique [GOL 05, JAU 01] is first exploited to characterize a box $[\tilde{x}]$ that contains $[f^{-1}(f([x_1])) \cap [x]]$. Secondly, an evaluation of $[\nabla f]([\tilde{x}])$ is performed in order to test its column rank[2]. Thus, since $[\nabla f([\tilde{x}])] \subset [\nabla f]([\tilde{x}])$ and according to Theorem 7, one can test whether f is $([x_1], [x])$-injective.
Algorithm 2 creates a paving of the initial box $[x]$ such that, for all i, the function f is $\left([x_i]\,, [x]\right)$-injective. Therefore, if the algorithm terminates, then f is an injection.
By combination of these two algorithms, we can prove that a function is injective over a box $[x]$. A solver, called ITVIA, developed in C++ is made available and tests the injectivity of a given function $f : \mathbb{R}^2 \to \mathbb{R}^2$ (or $f : \mathbb{R} \to \mathbb{R}^2$) over a given box $[x]$.

1.5. Examples

In this section, two examples are provide in order to illustrate the efficiency of the solver ITVIA presented in previous section. We are going to check the injectivity of two functions $f : \mathbb{R}^2 \to \mathbb{R}^2$ over a given box $[x]$.

---

2. Several techniques exist to test an interval matrix for full column ranking. If it is square, the simplest way consists in verifying that the determinant (which is an interval) not contains zero. Otherwise (*i.e.* $f : \mathbb{R}^n \to \mathbb{R}^m$), the Interval Gauss Algorithm could be used [NEU 90].

Algorithm 1 Partial_Injectivity_Test

Require: $f \in \mathcal{C}^1$, $[x]$ the initial box and $[x_1] \subset [x]$.
Ensure: A boolean :
  - true : f is $([x_1], [x])$-injective,
  - false : f may or not be partially injective.

Initialization : $\mathcal{L}_{stack} := \{[x]\}$, $[\tilde{x}] := \emptyset$.
while $\mathcal{L}_{stack} \neq \emptyset$ **do**
  Pop $\mathcal{L}_{stack}$ into $[w]$.
  if $[f]([w]) \cap [f]([x_1]) \neq \emptyset$ **then**
    if width$([w]) >$ width$([x_1])$ \\ To avoid useless splitting of $[w]$ *ad infinitum* **then**
      Bisect $[w]$ into $[w_1]$ and $[w_2]$.
      Stack $[w_1]$ and $[w_2]$ in $\mathcal{L}_{stack}$.
    else
      $[\tilde{x}] = [[\tilde{x}] \cup [w]]$.
    end if
  end if
end while
if $[\nabla f]([\tilde{x}])$ is full column rank **then**
  Return true \\ "f is $([x_1], [x])$-injective"
else
  Return False \\ "Failure"
end if

Algorithm 2 Injectivity_Test_Via_Interval_Analysis

Require: f a $\mathcal{C}^1$ function and $[x]$ the initial box.

Initialization : $\mathcal{L} := \{[x]\}$.
while $\mathcal{L} \neq \emptyset$ **do**
  Pull $[w]$ in $\mathcal{L}$.
  if Partial_Injectivity_Test$(f, [x], [w]) =$ False **then**
    Bisect $[w]$ into $[w_1]$ and $[w_2]$.
    Push $[w_1]$ and $[w_2]$ in $\mathcal{L}$.
  end if
end while
Return "f is injective over $[x]$".

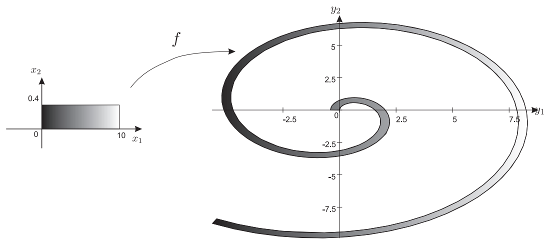

Figure 1.7. *Graph of the function f defined in (1.17)*

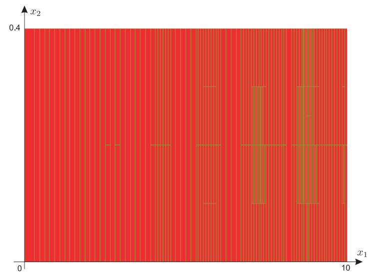

Figure 1.8. *Bisection of $[x]$ obtained by ITVIA for the function f defined in (1.17). All the grey boxes have been proved partially injective.*

1.5.1. *Spiral function*

Consider the function f, depicted in Figure 1.7, defined by

$$f : \begin{array}{ccccc} \mathbb{R}^2 & \rightarrow & \mathbb{R}^2 & & \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & \rightarrow & \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} & = & \begin{pmatrix} x_1 \sin(x_1) + x_2 \frac{x_1 \sin(x_1) - \cos(x_1)}{\sqrt{x_1^2+1}} \\ x_1 \cos(x_1) + x_2 \frac{\sin(x_1) + x_1 \cos(x_1)}{\sqrt{x_1^2+1}} \end{pmatrix} \end{array} \tag{1.17}$$

and test its injectivity over the box $[x] = \left([0, 10], [0, \frac{4}{10}]\right)^T$. After less than $0.1\,\mathrm{sec}$ on a Pentium $1.7GHz$, ITVIA proved that f is injective over $[x]$. The initial box $[x]$ has been divided in a set of sub-boxes where f is partially injective. Figure 1.8 shows the successive bisections of $[x]$ made by ITVIA.

1.5.2. *Ribbon function*

Consider the ribbon function f (depicted in Figure 1.9) defined by

$$f : \begin{array}{ccccc} \mathbb{R}^2 & \rightarrow & \mathbb{R}^2 & & \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & \rightarrow & \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} & = & \begin{pmatrix} \frac{x_1}{2} + (1 - x_2)\cos(x_1) \\ (1 - x_2)\sin(x_1) \end{pmatrix} \end{array} \tag{1.18}$$

and get interest with its injectivity over the box $[x] = ([-1, 4], [0, \frac{1}{10}])^T$. Since the ribbon overlapping, one can see that f is not injective over $[x]$. After 3 seconds, the solver ITVIA is stopped (before going to end). It returns the solution presented in Figure 1.10. The function f has been proved to be a partial injection on the gray domain over $[x]$, whereas the white domain corresponds to the indeterminate domain where ITVIA was not able to prove the partial injectivity. Indeed, the indeterminate domain corresponds to the non injective zone of f where all points are mapped in the overlapping zone of the ribbon.

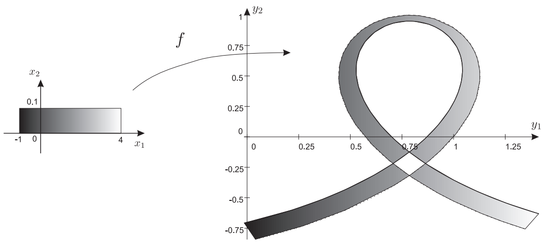

Figure 1.9. *Graph of the function f defined in (1.18).*

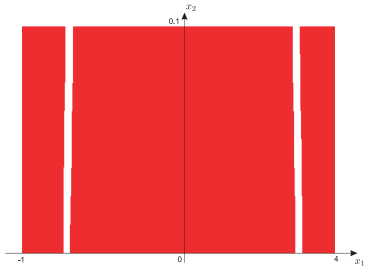

Figure 1.10. *Partition of the box $[x]$ obtained by ITVIA for the function f defined in (1.18). In gray, the partial injectivity domain and, in white, the domain where the f is not proved partially injective.*

1.6. Conclusion

In this paper, we have presented a new algorithm, based on interval analysis, able to test differentiable functions for injectivity. In case of functions $f : \mathbb{R} \to \mathbb{R}^2$ and $f : \mathbb{R}^2 \to \mathbb{R}^2$, a C++ solver is available. From a given function f and a given box $[x]$, the solver divides $[x]$ in two domains : a partially injective domain and a indeterminate domain (where the function may or not be injective). Of course, when the indeterminate domain is empty, the function is proved injective over $[x]$.

1.7. Bibliography

[BRA 01] BRAEMS I., JAULIN L., KIEFFER M., WALTER E., "Guaranteed numerical alternatives to structural identifiability testing", *Conference on Decision and Control.*, 2001.

[E.W 90] E.WALTER, PRONZATO L., "Qualitative and quantitative experiment design for phenomenological models, a survey.", *Automatica*, vol. 26, p. 195-213, 1990.

[GOL 05] GOLDSZTEJN A., "A Right-Preconditioning Process for the Formal-Algebraic Approach to Inner and Outer Estimation of AE-solution Set.", *Reliable Computing*, 2005.

[JAU 01] JAULIN J., KIEFFER M., DIDRIT D., WALTER E., *Applied Interval Analysis*, Springer, 2001.

[MOO 66] MOORE R., *Interval Analysis*, Prentice-Hall, Englewood Cliffs, 1966.

[NEU 90] NEUMAIER A., *Interval Methods for Systems of Equations*, Camb. Univ. Pres, 1990.