Constraints (2010) 15:190–212
DOI 10.1007/s10601-009-9082-3

A new framework for sharp and efficient resolution of NCSP with manifolds of solutions

Alexandre Goldsztejn · Laurent Granvilliers

Published online: 11 November 2009

Abstract When numerical CSPs are used to solve systems of n equations with n variables, the preconditioned interval Newton operator plays two key roles: First it allows handling the n equations as a global constraint, hence achieving a powerful contraction. Second it can prove rigorously the existence of solutions. However, none of these advantages can be used for under-constrained systems of equations, which have manifolds of solutions. A new framework is proposed in this paper to extend the advantages of the preconditioned interval Newton to under-constrained systems of equations. This is achieved simply by allowing domains of the NCSP to be parallelepipeds, which generalize the boxes usually used as domains.

Keywords Interval analysis · Branch and prune algorithm · Preconditioned interval Newton · Global constraint for under-constrained systems of equations

1 Introduction

The paper presents a new framework for solving numerical CSPs (NCSPs) formed of under-constrained systems of equations:

$$\langle\ \mathbf{x}\ ,\ \mathbf{f}(\mathbf{x}) = \mathbf{0}\ ,\ [\mathbf{x}]\ \rangle, \tag{1}$$

where vectorial notations are used, vectors being denoted by boldface symbols, i.e. $\mathbf{x} = (x_1, \ldots, x_n)$ is a vector of variables, $\mathbf{f} = (f_1, \ldots, f_m)$, with $m < n$, is a vector of

This paper is an extended version of [10] presented at CP 2008 where it has been granted the best research paper award.

A. Goldsztejn (✉)
CNRS, Laboratoire d'Informatique de Nantes Atlantique, Nantes, France
e-mail: Alexandre.Goldsztejn@univ-nantes.fr

L. Granvilliers
Laboratoire d'Informatique de Nantes Atlantique, University of Nantes, Nantes, France
e-mail: Laurent.Granvilliers@univ-nantes.fr

functions and $[\mathbf{x}] = ([x_1], \ldots, [x_n])$ is a vector of interval domains. These NCSPs arise naturally in a wide range of applications, among which are surface intersection characterization and plotting [27], a particular case of implicit equation solving [17], global optimization [13] and robots kinematics [19].

Under-constrained systems of equations typically have either manifolds of solutions or no solution at all: A NCSP with n variables and m equations typically has a solution set which is a manifold of dimension $n - m$. This is illustrated by Fig. 1: The leftmost graphic represents the graph of the constraint $x_1^2 + x_2^2 - x_3^2 = 0$ whose solution set is a manifold of dimension $3 - 1 = 2$ (excepted for the origin which is called a singular solution). The second constraint $x_1^2 + x_2^2 + x_3^2 = 1$ is added to obtain the solution set of the central graphic, which is a manifold of dimension $3 - 2 = 1$ (the union of the two black circles). Finally, a linear constraint is furthermore added to obtain a well constrained NCSP with a finite set of solutions represented on the rightmost graphic. More formally, the implicit function theorem states that if $\mathbf{x}$ is a solution of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ and if the Jacobian of $\mathbf{f}$ has full rank (in other words, if the gradients of the functions f_i evaluated at $\mathbf{x}$ are linearly independent) then there exists an unique dimension $n - m$ manifold of solutions containing $\mathbf{x}$ in a neighborhood of $\mathbf{x}$. Solutions where the Jacobian has not full rank are called singular solutions. For example, a system of equations that would contain two times the same equation would have only singular solutions (since two lines of its Jacobian would be equal). The framework proposed in this paper does not apply to singular solutions (in this case the proposed algorithm just behaves as a standard branch and prune algorithm based on box domains). However, singular solutions are untypical in the sense that some arbitrarily small perturbation of the system can remove them, and specific techniques like formal rewriting has been used to handle them.

Algorithms designed for well constrained NCSPs [12, 28] (which have typically 0 dimensional manifolds of solutions) are not efficient for solving under-constrained NCSPs (1). Indeed, these algorithms are variations of the branch and prune algorithm where the preconditioned interval Newton operator [24] plays a key role: On the one hand, it acts like a global constraint that performs powerful contraction when the domains become small enough. On the other hand, it can prove rigorously the existence of a solution of a well constrained system of equations. However, these two key contributions of the interval Newton operator are not operating when dealing

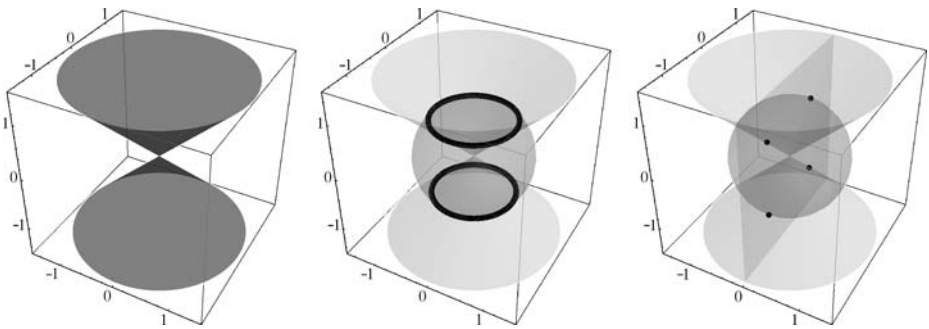

Fig. 1 Manifolds of different dimensions defined as the solutions of equality constraints

with under-constrained systems of equations, mainly because no preconditioning of these systems of equations has been proposed yet. The main contribution of this paper is to present a framework that extends these two advantages of the interval Newton operator to NCSPs formed of under-constrained systems of equations.

In the usual definition of a CSP, each variable is given a domain. As a matter of fact, it is equivalent to consider the Cartesian product of these variable domains (which is a box when variable domains are intervals) as a search space for a vector made of the CSP variables. Then, more complicated sets, which are no longer the Cartesian product of variable domains, can be used. The framework proposed in this paper shows that using parallelepiped domains instead of box domains can drastically improve the efficiency of branch and prune algorithms dedicated to (1). On the one hand, parallelepipeds offer a more flexible description of subsets of $\mathbb{R}^n$ than boxes, hence providing a more accurate enclosure of the NCSP solution set. On the other hand, using parallelepiped domains introduces an efficient preconditioning process for under-constrained systems of equations, hence allowing the preconditioned interval Newton operator to both work as a global constraint and prove the existence of solutions.

Parallelepipeds have already been used to advantageously replace boxes, for example to rigorously enclose the solutions of initial values problems (cf. the survey paper [22] and references therein). The relationship between parallelepipeds and preconditioning has already been investigated in [7, 23], where parallelepipeds are used to approximate the solution set of linear systems with interval uncertainties. However, the ways and means of the introduction of parallelepiped domains in the present work are totally different than in [7, 23]. Finally, parallelepipeds have been used in conjunction with existence theorems to build inner approximations of function ranges in [11], but again in the restricted case of well-constrained systems of equations.

Outline The framework proposed in this paper is presented on some motivating examples in Section 2. Then, the concepts of interval analysis used in this paper are given in Section 3. The technical description of the proposed branch and prune algorithm is given in Section 4. Finally, promising experiments are reported in Section 5.

2 Motivating examples

The usefulness of the usage of parallelepiped domains instead of box domains is now illustrated on two simple examples. The first shows how parallelepiped allow enclosing more precisely solution sets. The second illustrates the global constraint effect obtained using parallelepiped domains.

2.1 A two dimensional example

2.1.1 Contractions and bisections using box domains

Let us consider the very simple under-constrained NCSP

$$\langle\, \mathbf{x}\,,\ \{f(\mathbf{x}) = 0\}\,,\ [\mathbf{x}]\,\rangle, \tag{2}$$

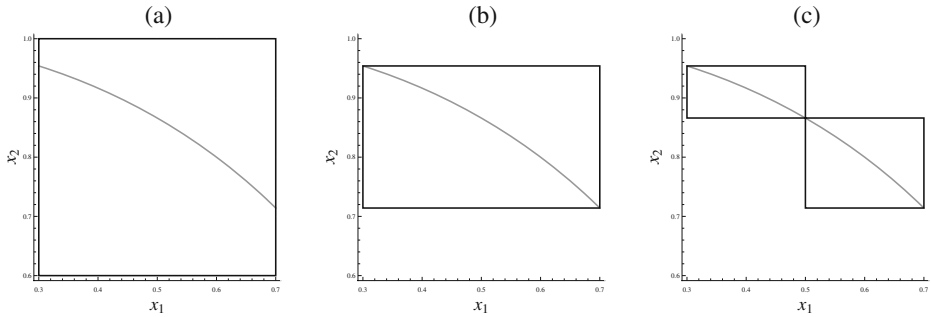

Fig. 2 **a** Solution set of the NCSP (2). **b** Domain after one contraction. **c** Domains after one bisection and two more contractions

with $\mathbf{x} = (x_1, x_2)$, $f(\mathbf{x}) = x_1^2 + x_2^2 - 1$ and $[\mathbf{x}] = ([0.3, 0.7], [0.6, 1.0])$. Its solution set is plotted in Fig. 2a.

The key point is to notice that the contraction and bisection processes of a branch and prune algorithm are not efficient in this situation: A contraction[1] is performed in Fig. 2b, and two more contractions are performed after a bisection in Fig. 2c. These plots clearly show that these contractions are not efficient because the solution set crosses the box domain in its diagonal. The contraction/bisection would be much more efficient if the solution set crossed the box domain along one of the axes. Reaching this situation is the goal of using parallelepiped domains instead of box domains.

2.1.2 From a box domain to a parallelepiped domain

To this end, a parallelepiped[2] is built whose two axes are respectively approximately parallel and perpendicular to the solution set. These directions are related to the gradient of the function f evaluated at the midpoint of the box domain. Once the parallelepiped axes are computed, the parallelepiped is chosen as small as possible under the constraint that it contains the original box domain (cf. Fig. 3a where the former box domain is represented using dashed lines).

Note that changing the box domain to an enclosing parallelepiped domain introduces new solutions on the border of the parallelepiped domain (the solutions that are inside the parallelepiped domain but outside the former box domain). In order to reject these additional solutions (which otherwise would be redundant with the neighbor domains), the former box domain is reintroduced as four inequality constraints which are added to the constraint store.

[1] In this introducing example, the optimal contractions are performed. In practice, contractions are not that efficient. Nevertheless, this illustrates the best that can be obtained from both methods.

[2] Note that in this motivating example parallelepipeds have perpendicular axes, but this is not the case in general for higher dimensions.

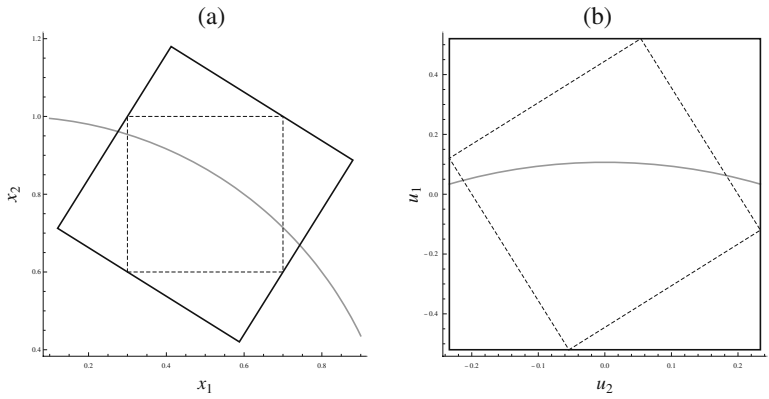

Fig. 3 **a** Enclosing parallelepiped domain for the NCSP (2). **b** The same NCSP expressed in the auxiliary basis formed of the parallelepiped axes

To see how a parallelepiped domain can improve the contraction/bisection process, we have to formalize its definition: A parallelepiped is the image of a box through an affine map $\mathbf{u} \mapsto C \cdot \mathbf{u} + \tilde{\mathbf{x}}$, i.e. a set $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}]\}$ where

- The matrix $C \in \mathbb{R}^{2\times 2}$ is a square matrix. This matrix gives its shape to the parallelepiped and is chosen so that the solution set is approximately parallel to one of its sides. This is done considering the gradient of the function evaluated at the midpoint of the box (cf. Section 4 for more details).
- The vector $\tilde{\mathbf{x}}$ is the midpoint of the box $[\mathbf{x}]$.
- The box $[\mathbf{u}]$ is computed in such a way that the parallelepiped encloses the box domain $[\mathbf{x}]$, i.e. $[\mathbf{u}] = C^{-1} \cdot ([\mathbf{x}] - \tilde{\mathbf{x}})$ where interval arithmetic is used (cf. Section 3).

Then, the NCSP (2) can be expressed in the auxiliary basis formed of the characteristic axes of the parallelepiped, giving rise to the auxiliary NCSP

$$\langle\ \mathbf{u}\ ,\ \{g(\mathbf{u}) = 0\}\ ,\ [\mathbf{u}]\ \rangle, \tag{3}$$

with $g(\mathbf{u}) = f(C \cdot \mathbf{u} + \tilde{\mathbf{x}})$, that is explicitly

$$g(u_1, u_2) = f(C_{11}u_1 + C_{12}u_2 + \tilde{x}_1, C_{21}u_1 + C_{22}u_2 + \tilde{x}_2). \tag{4}$$

The solution sets of (3) is represented by Fig. 3b. As mentioned previously, four linear inequalities are added to the constraints of (3) which represent the belonging to the original box domain (they are not given explicitly in (3) for clarity). These inequalities are represented using dashed lines in Fig. 3b. Note that the solution sets of (2) and (3) are closely related: The former is exactly the image of the latter through the affine transformation $\mathbf{u} \mapsto C \cdot \mathbf{u} + \tilde{\mathbf{x}}$.

The next two subsections show how to contract and bisect this parallelepiped domain.

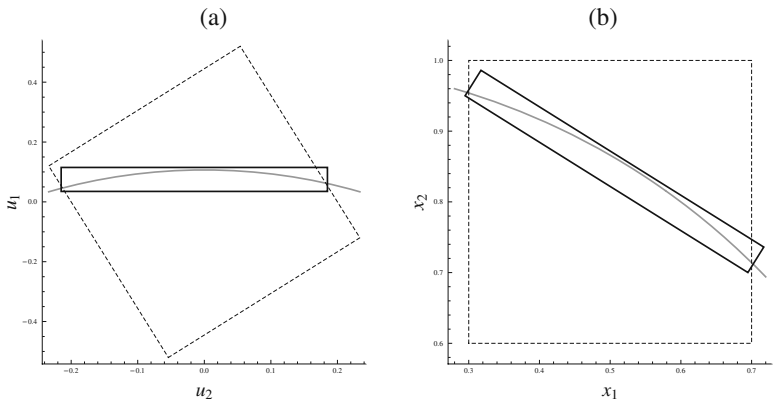

Fig. 4 Contracting parallelepiped domains (**a**, **b**)

2.1.3 Contracting parallelepiped domains

Contracting the parallelepiped $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}]\}$ consists in contracting its characteristic domain $[\mathbf{u}]$. The aim is to keep all the solutions of the original NCSP (2) within the contracted parallelepiped. This is obviously equivalent to contracting $[\mathbf{u}]$ without losing any solution of the auxiliary NCSP (3). As this later NCSP has a box domain, one can use the usual techniques dedicated to NCSPs. Note that since the solution set crosses the parallelepiped in the direction u_2, the constraint $g(\mathbf{u}) = 0$ will contract efficiently the domain $[u_1]$ but will certainly be useless for the domain $[u_2]$. On the other hand, the inequality constraints coming from the box domain will help contracting the domain $[u_2]$. The contraction of $[\mathbf{u}]$ obtained for this introducing example is shown in Fig. 4. Figure 4a shows how $[\mathbf{u}]$ is contracted using the auxiliary NCSP (3) while Fig. 4b shows the corresponding contraction for the parallelepiped domain of the original NCSP (2). Comparing Fig. 4b to Fig. 2b shows how much more efficient the contraction of the parallelepiped domain is compared to the contraction of the original box domain.

The auxiliary NCSP (3) is actually more complicated than the original NCSP (2): The function $g(\mathbf{u}) = f(C \cdot \mathbf{u} + \tilde{\mathbf{x}})$ contains more occurrences of each variables (cf. Eq. 4), which is well known to decrease the efficiency of interval based methods. However, this situation is very similar to the preconditioning of the interval Newton operator. And indeed, this acts like a *right-preconditioning*[3] where the interval Newton operator should be very efficient. Actually, Section 4 shows that this right-preconditioning process allows the interval Newton to both act like a global constraint, and rigorously prove the existence of the manifold of solutions. In the context of this introducing example, the interval Newton operator proves that for all $u_2 \in [u_2]$ there exists an unique $u_1 \in [u_1]$ such that $g(\mathbf{u}) = 0$, hence proving that the auxiliary solution set (3) crosses $[\mathbf{u}]$ along u_2. Therefore, the original (2) is proved to cross the parallelepiped in this direction.

[3]I.e. a preconditioning where the change of basis is applied before the function, see e.g. [7, 11, 23].

2.1.4 Bisecting parallelepiped domains

Bisecting a parallelepiped $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}]\}$ into two smaller parallelepipeds is naturally done by bisecting its characteristic domain $[\mathbf{u}]$ (cf. Fig. 5a), thus obtaining two new parallelepipeds $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}']\}$ and $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}'']\}$ (cf. Fig. 5b). Note that it obviously makes no sense to bisect $[\mathbf{u}]$ to $([u_1'], [u_2])$ and $([u_1''], [u_2])$, which would not preserve the solution set transverse crossing. Instead, bisecting $[\mathbf{u}]$ to $([u_1], [u_2'])$ and $([u_1], [u_2''])$ does preserve this transversality, and is therefore a very efficient bisection heuristic. This useful bisection heuristic will be easily extended to the general case of arbitrary dimensions.

Once bisected, their characteristic matrices are updated using the same process as described previously, but based on the gradient vector of f evaluated at the center of each new parallelepiped. This allows the new parallelepipeds adapting their shape more accurately to the shape of the solution set (cf. Fig. 5c). Furthermore, as done with the original box domain, each former parallelepiped domain is expressed as four additional inequality constraints in its NCSP (represented in dashed lines in Fig. 5c), which will be used to reduce the overlapping introduced when updating the characteristic matrices of the new parallelepipeds. Finally, a contraction is performed on these two new parallelepipeds, leading to Fig. 5d. Comparing this figure to Fig. 2c

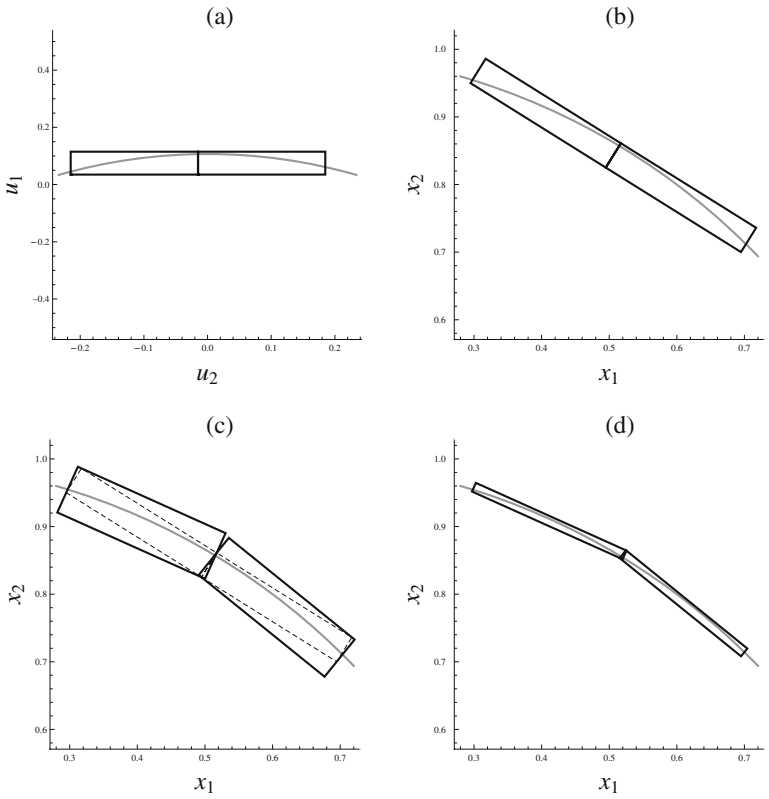

Fig. 5 Bisecting parallelepiped domains (**a**–**d**)

shows how important is the improvement obtained using parallelepipeds domains instead of box domains.

The gains in contracting efficiency and existence proving illustrated on this motivating example are even more important when dealing with more complicated NCSPs of higher dimension (c.f. Subsections 5.3 and 5.4).

2.2 A three dimensional example

The first example has shown that using parallelepipeds can drastically improve the representation and the enclosure of solution sets. Using parallelepipeds provides another key improvement: This gives rise to a global constraint algorithm. This could not been seen on the previous example since there were only one constraint, while the need of global constraint algorithms occurs only when several constraints are involved. Hence, we consider here the simplest situation where global constraint algorithms are useful, i.e. two constraints and three variables. To simplify the discourse, we consider linear constraints. Nonlinear constraints show the same behavior when domains are small enough.

Let us consider the NCSP

$$\langle\, \mathbf{x}\,,\ \{f_1(\mathbf{x}) = 0,\ f_2(\mathbf{x}) = 0\}\,,\ [\mathbf{x}]\,\rangle, \tag{5}$$

where $f_1(\mathbf{x}) = x_1 + x_2 - \frac{1}{2}x_3$, $f_2(\mathbf{x}) = x_1 - x_2 + \frac{1}{2}x_3$ and $[\mathbf{x}] = ([-1, 1], [-1, 1], [-1, 1])$. The solution set of this NCSP is depicted on the left hand side graphic of Fig. 6. It shows that both constraints have some solutions on each side of the domain. Therefore, none of the constraints can be used to contract the domain hence the need of a global constraint.

Let us choose a parallelepiped domain $\{C \cdot \mathbf{u} : \mathbf{u} \in [\mathbf{u}]\}$ whose side are parallel to the solution sets of the two constraints, and which contain the initial box domain.

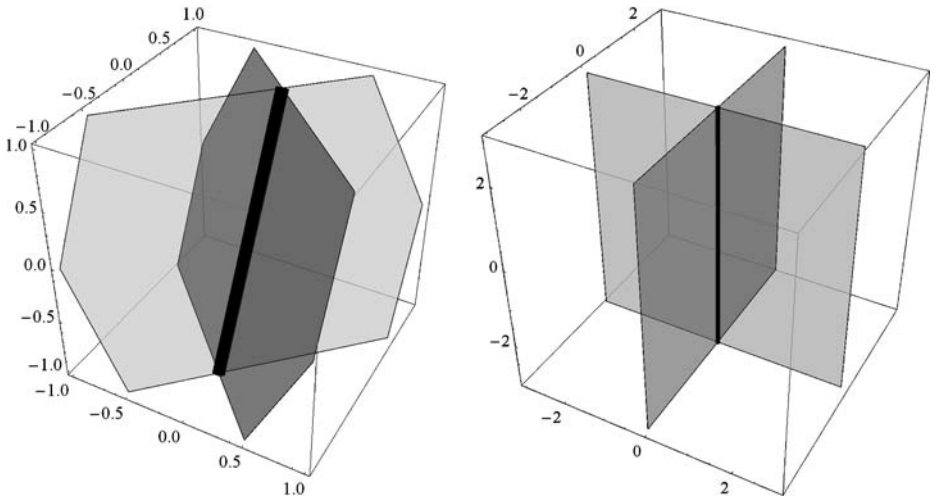

Fig. 6 A NCSP formed of two linear equalities. The original NCSP is represented on the *left hand side graphic*, the NCSP expressed in the auxiliary basis of the parallelepiped on the *right hand side graphic*

Such a parallelepiped domain is obtained defining C as the inverse of the matrix whose first two lines are the gradients of the constraints evaluated at the center of the box domain, and the third line is a vector perpendicular to the first two:

$$C = \begin{pmatrix} 1 & 1 & -\frac{1}{2} \\ 1 & -1 & \frac{1}{2} \\ \sqrt{2} & 0 & -\sqrt{2} \end{pmatrix}^{-1} = \begin{pmatrix} 2 & 2 & 0 \\ 3 & -1 & -\sqrt{2} \\ 2 & 2 & -2\sqrt{2} \end{pmatrix}. \tag{6}$$

Then, $[\mathbf{u}]$ is chosen as the smallest box that makes $\{C \cdot \mathbf{u} : \mathbf{u} \in [\mathbf{u}]\}$ a superset of $[\mathbf{x}]$, i.e. $[\mathbf{u}] = C^{-1}[\mathbf{x}]$ (cf. Proposition 1). Finally, the following auxiliary NCSP is considered:

$$\langle\, \mathbf{u}\,,\ \{f_1(C\mathbf{u}) = 0,\ f_2(C\mathbf{u}) = 0\}\,,\ [\mathbf{u}]\,\rangle. \tag{7}$$

Remark 1 As in Section 2.1, we can add to the auxiliary NCSP (7) some linear inequalities that represent the box domain $[\mathbf{x}]$ in the new basis. These inequalities are not considered here for simplicity.

As previously, this auxiliary NCSP is solved using classical techniques aiming strong filtering and rigorous proof of solution existence. The key point is to note that the auxiliary basis defined by C has been chosen to greatly simplify the NCSP (7). Indeed, expanding the constraints expressions in (7) gives rise to $f_1(C\mathbf{u}) = \mathbf{u}_1$ and $f_2(C\mathbf{u}) = \mathbf{u}_2$, whose solution sets are represented on the right hand side graphic of Fig. 6. The two constraints are independent in the auxiliary basis, and each of them allows a very strong contraction on the box auxiliary domain $[\mathbf{u}]$, leading to very accurate parallelepiped enclosure of the solution set of the original NCSP. This illustrates the global constraint effect of the usage of parallelepiped domains which allows contracting NCSPs domains where local filtering at single constraints level is inefficient.

Note that this motivating example involve linear constraints which idealize nonlinear constraints restricted to small domains around nonsingular solutions (i.e. where the Jacobian has full rank). Hence, we can expect to meet a similar effect on nonlinear constraints as soon as domains are small enough for linearization to make sense. The additional bisection process implemented in a branch and prune algorithm will force the convergence to such small domains where the parallelepiped domains will eventually show their usefulness.

3 Interval analysis for NCSP resolution

The modern interval analysis was born in the 60's with [20] (see also [14, 16, 24]). Since, it has been widely developed and is today one central tool in the resolution of constraints acting over continuous domains (see [3, 15] and extensive references).

3.1 Interval arithmetic

Intervals, interval vectors and interval matrices are denoted using brackets. The sets of closed intervals, interval vectors and interval matrices are denoted respectively

by $\mathbb{IR}$, $\mathbb{IR}^n$ and $\mathbb{IR}^{n\times m}$. The elementary function are extended to intervals in the following way: let $\circ \in \{+, -, \times, /\}$ then $[x] \circ [y] = \{x \circ y : x \in [x], y \in [y]\}$ (division is defined only for non zero containing interval denominators). Some formal expressions are easily obtained from this definition, e.g. $[a, b] + [c, d] = [a + c, b + d]$. Also, continuous one variable functions $f(x)$ are extended to intervals using the same definition: $f([x]) = \{f(x) : x \in [x]\}$, which is an interval because f is continuous. When one represents numbers using a finite precision (usually using floating point numbers, cf. [6]), the previous operations cannot be computed exactly in general. Therefore, the outward rounding is then used so as to keep valid the interpretations. For example, $[1, 2] + [2, 3]$ would be equal to $[2.999, 5.001]$ if outward rounding is used with a three decimal precision.

Then, an expression compound of these elementary operations can be evaluated for interval arguments using this interval arithmetic. The main property of interval analysis is that such an interval evaluation gives rise to a superset of the image through the function of the interval arguments: For example,

$$[x] \times ([y] - [x]) \supseteq \{x(y - x) : x \in [x], y \in [y]\}. \tag{8}$$

In some cases (e.g. when the expression contains only one occurrence of each variable), this enclosure is optimal. In particular, the computation $C \cdot [\mathbf{u}] + \tilde{\mathbf{x}}$ is the smallest box that contains the parallelepiped $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : u \in [\mathbf{u}]\}$ (see Fig. 7).

3.2 Interval contractors

Given an n-ary constraint c and a box $[\mathbf{x}] \in \mathbb{R}^n$, a contractor for c will contract the box $[\mathbf{x}]$ without losing any solution of c. Some widely used contractors are based on the 2B-consistency (also called hull-consistency) or the box consistency [1, 2, 18], which are adaptations of the arc-consistency to continuous domains. They are both applied to one constraint at a time, hence suffering of the usual drawbacks of the locality of their application. On the other hand, the preconditioned interval Newton (see [24] and extensive references) can be applied to a set of n equations and n variables and is able to treat this set of constraints as a global constraint. The usage of hull-consistency for linear inequalities is detailed in Subsection 3.2.1

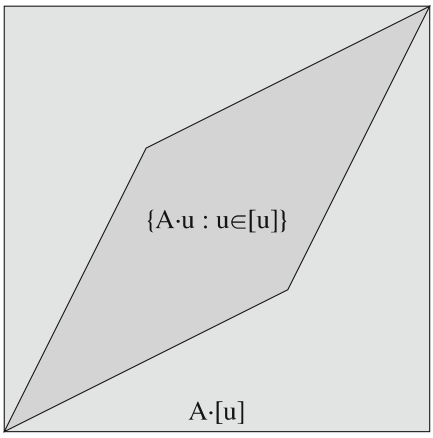

Fig. 7 A parallelepiped $\{A \cdot \mathbf{u} : \mathbf{u} \in [\mathbf{u}]\}$ and its interval hull $A \cdot [\mathbf{u}]$

while the preconditioned interval Newton operator is presented in Subsection 3.2.2. Subsection 3.2.3 finally shows how the preconditioned interval Newton can be applied to a parametric system of equations.

3.2.1 Hull consistency for linear inequalities

Although box consistency is stronger than hull-consistency based algorithms the later is optimal when applied to a single constraint $f(\mathbf{x}) = 0$ or $f(\mathbf{x}) \leq 0$ if the expression of f has only one occurrence of each variables (cf. [5]). It is furthermore computed with much less computational work than the box-consistency. Therefore, hull-consistency based algorithms are used to contract boxes under linear inequality constraints that will be met in the sequel.

3.2.2 Interval Newton for $n \times n$ systems of equations

When a NCSP is a system of n equations with n unknowns the interval Newton can be used to efficiently contract domains. There are several versions of the interval Newton operator and we will use here the Hansen-Sengupta operator (the other widely used interval Newton operator being the Krawczyk operator, cf. [24] for details). The interval Gauss-Seidel is first defined as follows: In dimension one, $\Gamma\big([a], [b], [x]\big) := \square\big([x] \cap \{x | (\exists a \in [a])(\exists b \in [b])(ax = b)\}\big)$. In the case where $0 \notin [a]$, one obtains the expression $\Gamma\big([a], [b], [x]\big) = ([b]/[a]) \cap [x]$ (cf. [24] for the expression in the case $0 \in [a]$). Then, in arbitrary dimension, $\Gamma\big([A], [\mathbf{b}], [\mathbf{x}]\big) := [\mathbf{y}]$ with

$$[y_i] := \Gamma\Big([a_{ii}]\,,\ [b_i] - \sum_{j<i} [a_{ij}][y_j] - \sum_{j>i} [a_{ij}][x_j]\,,\ [x_i]\Big). \tag{9}$$

Finally, the Hansen-Sengupta operator is defined as follows:

$$\mathrm{HS}(\mathbf{f}, [\mathbf{x}]) = \tilde{\mathbf{x}} + \Gamma\big(J_{\mathbf{f}}([\mathbf{x}]), -\mathbf{f}(\tilde{\mathbf{x}}), [\mathbf{x}] - \tilde{\mathbf{x}}\big), \tag{10}$$

where $\tilde{\mathbf{x}} = \mathrm{mid}[\mathbf{x}]$ and $J_{\mathbf{f}}$ is the Jacobian of $\mathbf{f}$ ($J_{\mathbf{f}}([\mathbf{x}])$ being its interval evaluation). The interval Newton operator can both contract a box without loosing any solution and prove the existence of solutions. Formally it verifies:

- $\mathbf{x} \in [\mathbf{x}] \wedge \mathbf{f}(\mathbf{x}) = \mathbf{0} \implies \mathbf{x} \in \mathrm{HS}(\mathbf{f}, [\mathbf{x}])$,
- if $\emptyset \neq \mathrm{HS}(\mathbf{f}, [\mathbf{x}]) \subseteq \mathrm{int}[\mathbf{x}]$ then there exists an unique solution inside $[\mathbf{x}]$.

The interval Newton becomes efficient when domains are small enough. Then it is able to treat a $n \times n$ system of equations as a global constraints thanks to preconditioning: Preconditioning a system of equations $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ consists of solving instead the auxiliary system of equations $\mathbf{g}(\mathbf{x}) = C \cdot \mathbf{f}(\mathbf{x}) = \mathbf{0}$, where $C \in \mathbb{R}^{n \times n}$ is nonsingular. Both the original system and the preconditioned system have the same solutions because C is nonsingular. The Jacobian of this auxiliary system is easily computed: $J_{\mathbf{g}}(\mathbf{x}) = C \cdot J_{\mathbf{f}}(\mathbf{x})$. If C is well chosen (usually the the midpoint-inverse preconditioning, i.e. $C = (\mathrm{mid} J_{\mathbf{f}}([\mathbf{x}]))^{-1}$, so that $J_{\mathbf{g}}(\mathbf{x})$ is close to the identity matrix) and if the domains are small enough then the preconditioned system will be much better suited to interval contractors like the interval Newton. The following example illustrates the power of treating a system of equations as a global constraint using the preconditioned interval Newton.

Example 1 Consider the 2×2 system of equations

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} 3(x_1 - x_2) + x_1 x_2 - 1 \\ 2(x_1 + x_2) + x_1 x_2 + 1 \end{pmatrix} = \mathbf{0}, \tag{11}$$

and the domain $[\mathbf{x}] = ([-2, 2], [-2, 2])$. The graphs of the two constraints are depicted on the left hand side graphic of Fig. 8, the unique solution being represented by a point. The non-preconditioned Hansen-Sengupta operator does not contract at all the domain while enforcing hull consistency to each constraint locally leads to $([-\frac{5}{4}, \frac{7}{5}], [-\frac{19}{17}, 2])$ which is represented in dashed line on the left hand side graphic of Fig. 8. This reduced domain cannot be contracted anymore using any of two constraints independently.

Let us now consider the auxiliary system $\mathbf{g}(\mathbf{x}) = C \cdot \mathbf{f}(\mathbf{x})$, with midpoint-inverse preconditioner. The graphs of these two constraints are shown on the right hand side graphic of Fig. 8. We can see that the two constraints solution sets are almost parallel to axes, which will allow contractors based on each constraints to effectively contract the domains. The interval evaluation of the Jacobian of $\mathbf{g}$ is

$$J_{\mathbf{g}}([\mathbf{x}]) = C \cdot J_{\mathbf{f}}([\mathbf{x}]) = \begin{pmatrix} [\frac{3}{4}, \frac{5}{4}] & [-\frac{1}{4}, \frac{1}{4}] \\ [-\frac{1}{4}, \frac{1}{4}] & [\frac{3}{4}, \frac{5}{4}] \end{pmatrix}. \tag{12}$$

Thanks to the midpoint inverse preconditioning, the Jacobian of the auxiliary system is centered on the identity matrix. On the other hand, as interval domains are small enough the interval matrix (12) is strictly diagonally dominant. These two properties make the Hansen-Sengupta operator extremely efficient: After 7 applications of the operator, we obtain an enclosure of the solution of width 10^{-16} (the midpoint-inverse preconditioner being updated at each application). The first three contracted boxes are represented in dashed lines on the right hand side graphic of Fig. 8.

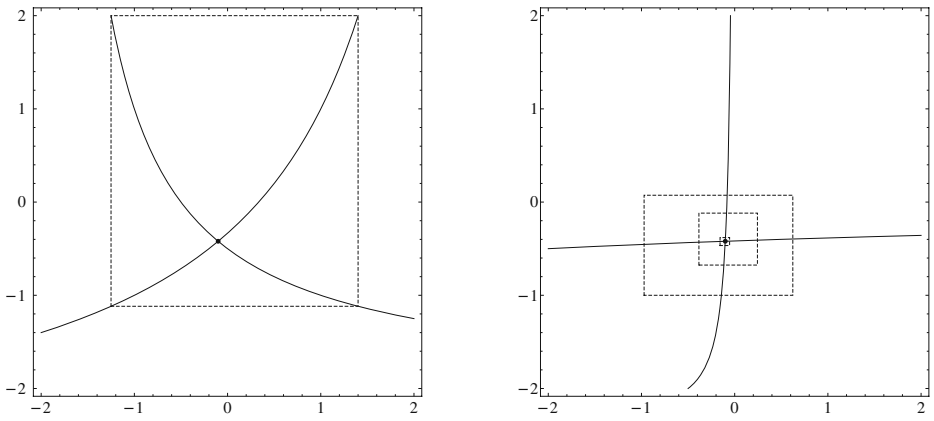

Fig. 8 Two constraints solution set before preconditioning (*left*) and after midpoint-inverse preconditioning (*right*)

3.2.3 Parametric systems of equations

By the implicit function theorem (see e.g. [25]), an underconstrained system of equations with n equations and $p > n$ variables typically gives rise to a solution set of dimension $m := p - n$ which can be locally parametrized in the form $\mathbf{f}(\mathbf{x}, \mathbf{a}) = \mathbf{0}$ with $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n$ (where $\mathbf{x}$ is the vector of variables and $\mathbf{a}$ is the vector of parameters). Let us now consider such a parametric square system of equations. Let us denote the Jacobian w.r.t. variables (respectively parameters) $J^{\mathbf{x}}_{\mathbf{f}}$ (respectively $J^{\mathbf{a}}_{\mathbf{f}}$). The parametric systems met in the sequel are ideal in the sense that:

- $J^{\mathbf{a}}_{\mathbf{f}}$ is close to zero, meaning that the system of equation is not very sensitive to parameters
- $J^{\mathbf{x}}_{\mathbf{f}}$ is close to identity, meaning that nonlinear equations are nearly independent and the interval Newton operator will apply efficiently.

In this situation, the interval Newton operator can be applied to the square system of equations handling rigorously the dependency w.r.t. parameters. The parametric Hansen-Sengupta operator [8] is defined as follows:

$$\mathrm{HS}(\mathbf{f}, [\mathbf{x}], [\mathbf{a}]) = \tilde{\mathbf{x}} + \Gamma\big(J^{\mathbf{x}}_{\mathbf{f}}([\mathbf{x}], [\mathbf{a}]), -(\mathbf{f}(\tilde{\mathbf{x}}, \tilde{\mathbf{a}}) + [\mathbf{b}]), [\mathbf{x}] - \tilde{\mathbf{x}}\big), \tag{13}$$

where $\tilde{\mathbf{x}} = \mathrm{mid}[\mathbf{x}]$, $\tilde{\mathbf{a}} = \mathrm{mid}[\mathbf{a}]$ and $\mathbf{b} = J^{\mathbf{a}}_{\mathbf{f}}([\mathbf{x}], [\mathbf{a}]) \cdot ([\mathbf{a}] - \tilde{\mathbf{a}})$ is the correction to apply for handling rigorously the dependency w.r.t. parameters. A parametric Krawczyk operator was proposed in [26] (see [9] for a comparison of these two parametric Newton like operators). The parametric Newton operators enjoy the same properties as their non parametric counterparts:

- $\mathbf{x} \in [\mathbf{x}] \wedge \mathbf{a} \in [\mathbf{a}] \wedge \mathbf{f}(\mathbf{x}, \mathbf{a}) = \mathbf{0} \implies \mathbf{x} \in \mathrm{HS}(\mathbf{f}, [\mathbf{x}], [\mathbf{a}])$,
- if $\emptyset \neq \mathrm{HS}(\mathbf{f}, [\mathbf{x}], [\mathbf{a}]) \subseteq \mathrm{int}[\mathbf{x}]$ then there exists an unique solution inside $[\mathbf{x}]$ for all $\mathbf{a} \in [\mathbf{a}]$.

4 Description of the algorithm

The branch and prune Algorithm 1 used here is classical except for line 5 which will be explained in the rest of the paper: The input is a NCSP $\mathcal{P}$ and the output a set of NCSPs $\mathcal{L} = \{\mathcal{P}_1, \ldots, \mathcal{P}_s\}$ whose disjunction is equivalent to the original NCSP. Formally,

$$\mathrm{Sol}(\mathcal{P}) = \bigcup_{\mathcal{Q} \in \mathcal{L}} \mathrm{Sol}(\mathcal{Q}). \tag{14}$$

Normally, the set of NCSPs $\mathcal{L}$ is a more accurate description of the solution set that the original NCSP: First, the union of their domains is much smaller than the original domain, hence providing a sharper enclosure. Second, it is often possible to prove that these NCSPs actually contains some solutions, which is a crucial information. The solution existence proof is not explicitly described in Algorithm 1 for clarity. Informally, some existence proof can result of the contraction performed at Line 6 when it is computed using the interval Newton operator. When the existence of solutions is proved, this information is attached to the NCSP.

When the algorithm starts, the domain of the NCSP is a box. The function UpdateDomainShape at Line 5 allows changing the shape of the domain (changing a

box to a parallelepiped when first successfully applied, or a parallelepiped to another parallelepiped more suited to the shape of the solution set). The first change from a box to a parallelepiped is performed only when the shape of the solution set can be foreseen from the evaluation of the constraints derivatives (cf. Subsection 4.1), which implies that the box domain is small enough. As a consequence, the algorithm will use box domains in a first phase, and parallelepiped domains as soon as box domains are small enough to identify the directions of the solution manifold. While the NCSP domain is a box, usual contractors are used to prune its domain (2B-consistency based contractors and non-preconditioned interval Newton operator in our implementation of Algorithm 1, whose collaboration is known to be efficient). When the domain is changed to a parallelepiped, the interval Newton operator is adapted and allows powerful contractions and existence proof of solutions (cf. Subsection 4.2). Finally, parallelepiped domains are bisected similarly to box domains (cf. Subsection 4.3).

Algorithm 1: Branch and prune algorithm using parallelepiped domains.

Input: $\mathcal{P} = \langle \mathbf{x}, \mathcal{C}, [\mathbf{x}] \rangle$, ϵ
Output: $\mathcal{L} = \{\mathcal{P}_1, \ldots, \mathcal{P}_s\}$
$\mathcal{T} \leftarrow \{\mathcal{P}\}$; $\mathcal{L} \leftarrow \emptyset$;
while (not $\mathcal{T} = \emptyset$) **do**
  $\mathcal{P} \leftarrow$ Extract($\mathcal{T}$);
  if (Measure(Domain($\mathcal{P}$)) $> \epsilon$) **then**
    $\mathcal{P}' \leftarrow$ UpdateDomainShape($\mathcal{P}$);
    $\mathcal{P}'' \leftarrow$ Contract($\mathcal{P}'$);
    if (Domain($\mathcal{P}''$) $\neq \emptyset$) **then**
      $(\mathcal{Q}, \mathcal{Q}') \leftarrow$ Bisect($\mathcal{P}''$);
      $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathcal{Q}, \mathcal{Q}'\}$;
    end
  else
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathcal{P}\}$;
  end
end
return ($\mathcal{L}$);

4.1 Changing the shape of a parallelepiped

The function UpdateDomainShape($\mathcal{P}$) attempts to find a new parallelepiped domain $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}]\}$ that will be more suited to the solution set of $\mathcal{P}$. The former domain of $\mathcal{P}$ is the parallelepiped $\{D \cdot \mathbf{v} + \tilde{\mathbf{x}} : \mathbf{v} \in [\mathbf{v}]\}$, which is possibly a box in which case $D = \mathrm{id}$ (note that the former and the new parallelepiped domains share the same characteristic vector $\tilde{\mathbf{x}}$). This process is done in two steps: First a candidate

new parallelepiped domain $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}]\}$ is computed. Then, the efficiency of this candidate parallelepiped domain is verified a posteriori. If not useful, the former parallelepiped is kept. If the candidate parallelepiped domain is chosen, then some inequality constraints are added to the NCSP in order to prevent some parasite solutions to appear due to the enclosure of the former parallelepiped domain inside a new parallelepiped (cf. Section 2).

4.1.1 Computation of the candidate parallelepiped domain

As illustrated in Section 2, the aim of using a new parallelepiped domain $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}]\}$ is to apply the interval Newton operator in the auxiliary basis of the parallelepiped to reduce directly the box domain $[\mathbf{u}]$. Hence, the parallelepiped characteristic matrix C is chosen aiming an efficient application of the interval Newton operator. The box domain $[\mathbf{u}]$ will be reduced using the constraint $\mathbf{g}(\mathbf{u}) = \mathbf{0}$ where $\mathbf{g}(\mathbf{u}) = \mathbf{f}(C \cdot \mathbf{u} + \tilde{\mathbf{x}})$ (cf. Section 2). To obtain an efficient application of the interval Newton operator, the under-constrained system $\mathbf{g}(\mathbf{u}) = \mathbf{0}$ needs to be interpreted as a system of m equations and m variables, where the remaining $n - m$ variables are considered as parameters (cf. Subsection 3.2.3). Thus, to be efficient the interval Newton operator requires the Jacobian $J_{\mathbf{g}}$ of $\mathbf{g}$ to be close to the matrix $\left(I_{m\times m} \middle| 0_{m\times(n-m)} \right)$, where $I_{m\times m}$ is the identity matrix of size $m \times m$ and $0_{m\times(n-m)}$ is the null matrix of size $m \times (n - m)$. As $J_{\mathbf{g}} = J_{\mathbf{f}} \cdot C$, it is natural to choose C such that

$$J_{\mathbf{f}}(\tilde{\mathbf{x}}) \cdot C \approx \left(I_{m\times m} \middle| 0_{m\times(n-m)} \right). \tag{15}$$

To this end, C is constructed based on the evaluation of the Jacobian of $\mathbf{f}$ at $\tilde{\mathbf{x}}$. The matrix C^{-1} is first constructed as follows, and C will be obtained inverting C^{-1}:

- The m first lines of C^{-1} are $J_{\mathbf{f}}(\tilde{\mathbf{x}})$.
- The i^{th} line for $m < i$ is chosen orthogonal to all previous lines (these lines are not uniquely determined, but a set of such vectors is easy to obtain using a Gram-Schmidt orthogonalization).

Then, this matrix is inverted to obtain C (note that C needs only to be computed approximately and thus standard double precision computations can be used at this step).

Remark 2 The computation of C requires C^{-1} to be nonsingular. If not, $C = D$ is chosen, which has to be the identity matrix (that is the domain was a box). Indeed, C^{-1} cannot be singular if D is different than the identity matrix since the test performed in Subsection 4.1.2 for the validation of the candidate parallelepiped domain implies that all matrices in $J_{\mathbf{f}}([\mathbf{x}])$ have full rank, this property being preserved for all subdomains subsequently met during the search.

The matrix C thus satisfies $J_{\mathbf{f}}(\tilde{\mathbf{x}}) \cdot C = \left(I_{m\times m} \middle| 0_{m\times(n-m)} \right)$, up to rounding errors (indeed by construction, $J_{\mathbf{f}}(\tilde{\mathbf{x}}) \cdot C$ is made of the m first rows of the $n \times n$ identity matrix).

Finally, the next proposition shows that the characteristic domain $[\mathbf{u}]$ should be $(C^{-1} \cdot D) \cdot [\mathbf{v}]$ so that the new parallelepiped domain is the smallest parallelepiped with characteristic matrix C and vector $\tilde{\mathbf{x}}$ that encloses the initial parallelepiped domain.

Proposition 1 *The nonsingular matrices* $C, D \in \mathbb{R}^{n\times n}$*, the vector* $\tilde{\mathbf{x}} \in \mathbb{R}^n$ *and the interval vector* $[\mathbf{v}] \in \mathbb{IR}^n$ *being fixed,* $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}]\}$ *with* $[\mathbf{u}] = (C^{-1} \cdot D) \cdot [\mathbf{v}]$ *is the smallest parallelepiped that contains* $\{D \cdot \mathbf{v} + \tilde{\mathbf{x}} : \mathbf{v} \in [\mathbf{v}]\}$.

Proof It is well known that $(C^{-1} \cdot D) \cdot [\mathbf{v}]$ is the interval hull of (i.e. the smallest box that contains) $\{C^{-1} \cdot D \cdot \mathbf{v} : \mathbf{v} \in [\mathbf{v}]\}$.

First, let us prove that $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}]\} \supseteq \{D \cdot \mathbf{v} + \tilde{\mathbf{x}} : \mathbf{v} \in [\mathbf{v}]\}$. Let $\mathbf{x}$ be an element of the first parallelepiped, thus $\mathbf{x} = D \cdot \mathbf{v} + \tilde{\mathbf{x}}$ with $\mathbf{v} \in [\mathbf{v}]$. This can be written $\mathbf{x} = C\mathbf{u} + \tilde{\mathbf{x}}$ with $\mathbf{u} = C^{-1} \cdot D \cdot \mathbf{v}$. Finally, $\mathbf{u} \in [\mathbf{u}]$ since $[\mathbf{u}]$ is the interval hull of $\{C^{-1} \cdot D \cdot \mathbf{v} : \mathbf{v} \in [\mathbf{v}]\}$ and therefore $\mathbf{x}$ is inside the second parallelepiped.

Second, let us prove that the first parallelepiped is the smallest that contains the second. To this end, let us consider a smaller parallelepiped $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}']\}$ where $[\mathbf{u}']$ must be strictly included inside $[\mathbf{u}]$. As $[\mathbf{u}]$ is the interval hull of $\{C^{-1} \cdot D \cdot \mathbf{v} : \mathbf{v} \in [\mathbf{v}]\}$, there exists $\mathbf{v} \in [\mathbf{v}]$ such that $C^{-1} \cdot D \cdot \mathbf{v} \notin [\mathbf{u}']$. As C is nonsingular, this shows that

$$C \cdot (C^{-1} \cdot D \cdot \mathbf{v}) + \tilde{\mathbf{x}} \notin \{C\mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}']\}, \tag{16}$$

which contradicts $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}]\} \supseteq \{D \cdot \mathbf{v} + \tilde{\mathbf{x}} : \mathbf{v} \in [\mathbf{v}]\}$. □

Remark 3 Be aware that interval matrix and vector multiplication is not associative, and that $C^{-1} \cdot (D \cdot [\mathbf{v}])$ is generally much larger than $(C^{-1} \cdot D) \cdot [\mathbf{v}]$, the latter being the best to use.

The outwardly rounded interval arithmetic is used to compute $[\mathbf{u}] = (C^{-1} \cdot D) \cdot [\mathbf{v}]$ so as to ensure a rigorous enclosure.

4.1.2 Verification of the efficiency of the candidate new parallelepiped domain

We intend to apply the interval Newton operator to the square subsystem formed of the first m variables (cf. Section 2 and Section 4.2). Formally, the interval Newton will be efficient if the square matrix formed of the first m columns of $J_{\mathbf{f}}([\mathbf{x}]) \cdot C$, where $[\mathbf{x}]$ is the interval hull of the parallelepiped domain, is centered on the identity matrix and strictly diagonally dominant (cf. Theorem 5.2.5 in [24]). Therefore, the interval matrix $J_{\mathbf{f}}([\mathbf{x}]) \cdot C$ is computed explicitly, and the strict diagonal dominance of the square matrix formed of its first m columns is checked. If it is actually strictly diagonally dominant, the parallelepiped domain is updated, and some inequality constraints are added to the NCSP (cf. the next section). Otherwise, the former parallelepiped domain is kept, i.e. $C = D$ and $[\mathbf{u}] = [\mathbf{v}]$.

Remark 4 If $\mathbf{x}$ is a solution (i.e. $\mathbf{f}(\mathbf{x}) = \mathbf{0}$) and is singular (i.e. $D\mathbf{f}(\mathbf{x})$ is not full rank, e.g. has two proportional lines) then $J_{\mathbf{f}}([\mathbf{x}]) \cdot C$ will never be diagonally dominant. In this case, the algorithm keeps working with box domains around this singular solution. This situation is untypical as some arbitrary small perturbation of the problem can change the singular solutions to regular solutions, excepted if for example $\mathbf{f}$ has two equal components as mentioned in introduction.

4.1.3 Adding linear inequalities to the new NCSP

As illustrated in Section 2, the enclosure of the former parallelepiped domain by a new parallelepiped domain is not perfect. Hence, some solutions that are inside the new domain may not be in the former, which would lead to redundant solutions if not properly treated. To this end, the former parallelepiped domain is reintroduced in the new NCSP as $2n$ linear constraints:

$$\underline{v}_i \leq \sum_{1 \leq j \leq n} D_{ij} x_j \leq \overline{v}_i, \tag{17}$$

for $1 \leq i \leq n$. As a consequence, the former and the new NCSPs have the same solution set. These linear inequalities will be denoted using the scalar product notation $\mathbf{a} \cdot \mathbf{x} \leq b$ where $\mathbf{a} = (D_{i1}, \ldots, D_{in})$ and $b = \overline{v}_i$, or $\mathbf{a} = -(D_{i1}, \ldots, D_{in})$ and $b = -\underline{v}_i$.

4.2 Contracting a parallelepiped domain

When the domain of the NCSP is a box, the usual techniques are used to contract it. When it is a parallelepiped, two kind of constraints are in the store: the linear inequalities $\mathbf{a} \cdot \mathbf{x} \leq b$ and the nonlinear equalities $\mathbf{f}(\mathbf{x}) = \mathbf{0}$.

4.2.1 Linear inequalities

In order to contract the characteristic domain $[\mathbf{u}]$ of the parallelepiped $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}]\}$ under the linear inequality $\mathbf{a} \cdot \mathbf{x} \leq b$, this constraint is simply expressed in the basis of the parallelepiped:

$$\mathbf{a} \cdot \mathbf{x} \leq b \iff (\mathbf{a} \cdot C) \cdot \mathbf{u} \leq b - \mathbf{a} \cdot \tilde{\mathbf{x}}. \tag{18}$$

Then, $[\mathbf{u}]$ is contracted under the new linear inequality using the 2B-consistency.

4.2.2 Nonlinear equalities

The nonlinear equalities are also expressed in the basis of the parallelepiped using the affine change of basis $\mathbf{x} = C \cdot \mathbf{u} + \tilde{\mathbf{x}}$. Thus $\mathbf{f}(\mathbf{x}) = \mathbf{0} \iff \mathbf{g}(\mathbf{u}) = \mathbf{0}$, with $\mathbf{g}(\mathbf{u}) = \mathbf{f}(C \cdot \mathbf{u} + \tilde{\mathbf{x}})$. As discussed in Section 2, contracting the parallelepiped domain $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}]\}$ for the constraints $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ consists of contracting its characteristic box $[\mathbf{u}]$ for the constraints $\mathbf{g}(\mathbf{u}) = \mathbf{0}$. Two cases arise when contracting domains using equality constraints depending on the success of the verification performed at Subsection 4.1.2.

If it never succeeded upstream the search tree then $C = I$ (i.e. the domain is still a box) and the m first columns of $J_{\mathbf{f}}([\mathbf{x}]) \cdot C) = J_{\mathbf{f}}([\mathbf{x}])$ do not verify the conditions presented in Subsection 4.1.2. In this case, the usual interval contractors are used to contract $[\mathbf{u}]$ (the 2B-consistency in the experiments presented in Section 5) under the constraints $\mathbf{g}(\mathbf{u}) = \mathbf{0}$, which is equivalent to contracting a box domain using the constraints $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ as $C = I$.

Otherwise, the candidate parallelepiped has been selected at least once following the criteria of Subsection 4.1.2,[4] which implies that the first m columns of $J_{\mathbf{f}}([\mathbf{x}]) \cdot C$ form a strictly diagonally dominant interval matrix centered on identity, while the other entries of $J_{\mathbf{f}}([\mathbf{x}]) \cdot C$ are centered on zero. The key point is to consider the under-constrained system $\mathbf{g}(\mathbf{u}) = \mathbf{0}$ as a parametric square system $\mathbf{g}(\mathbf{u}', \mathbf{u}'') = \mathbf{0}$ where $\mathbf{u}' = (u_1, \ldots, u_m)$ are the variables and $\mathbf{u}'' = (u_{m+1}, \ldots, u_n)$ are the parameters. The conditions verified at Subsection 4.1.2 then read as follows: $J_{\mathbf{g}}^{\mathbf{u}'}([\mathbf{u}'], [\mathbf{u}'']) \in \mathbb{IR}^{m \times m}$ is centered on the identity matrix and strictly diagonally dominant while $J_{\mathbf{g}}^{\mathbf{u}''}([\mathbf{u}'], [\mathbf{u}'']) \in \mathbb{IR}^{m \times n}$ is centered on zero. Therefore, this parametric square system is potentially very well suited to the application of the parametric Newton operator as presented in Subsection 3.2.3. Hence, the domains of the variables of the parametric system, which correspond to the m first variables of the under-constrained system, will be efficiently contracted. In addition, the parametric interval Newton will be able to prove that $\forall \mathbf{u}'' \in [\mathbf{u}''], \exists \mathbf{u}' \in [\mathbf{u}'], \mathbf{g}(\mathbf{u}', \mathbf{u}'') = \mathbf{0}$ holds, hence proving that the solution set crosses the whole parallelepiped domain transversally. When the existence is proved, this information is recorded together with the NCSP.

Note that the domains of the parameters of the parametric system, which correspond to the $n - m$ last variables of the under-constrained system, are not contracted using the nonlinear equalities. As illustrated in Section 2, such a contraction would be impossible or inefficient in general.

4.3 Bisecting a parallelepiped domain

A parallelepiped domain $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}} : \mathbf{u} \in [\mathbf{u}]\}$ is bisected splitting $[\mathbf{u}]$. Again, two case arise depending on the success of the conditions of Subsection 4.1.2. If they were never satisfied upstream the search tree, then the largest component of $[\mathbf{u}]$ is bisected, thus forcing the convergence of the branch and prune algorithm. If it has succeeded then the same bisection heuristic is restricted to the last $n - m$ variables (i.e. the parameters domains of the parametric system). This is sufficient to ensure the convergence of the algorithm by Theorem 5.2.5 in [24]: since the bisection of the last $n - m$ variables domains will eventually force their width to converge to zero, Theorem 5.2.5 of [24] proves that the interval Newton applied to the m first variables will be able to reject any non solution vector. Therefore, $[\mathbf{u}]$ is bisected to $[\tilde{\mathbf{u}}_1]$ and $[\tilde{\mathbf{u}}_2]$ where the largest component of $([u_m], [u_{m+1}], \ldots, [u_n])$ has been bisected. Finally, the two bisected parallelepipeds $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}}_1 : \mathbf{u} \in [\mathbf{u}_1]\}$ and $\{C \cdot \mathbf{u} + \tilde{\mathbf{x}}_2 : \mathbf{u} \in [\mathbf{u}_2]\}$ are defined as follows: for both $k = 1$ and $k = 2$, $\tilde{\mathbf{x}}_k = C \cdot \text{mid}([\tilde{\mathbf{u}}_k]) + \tilde{\mathbf{x}}$ so that this vector is at the center of the parallelepiped, and hence is representative of the domain. The domains $[\mathbf{u}_1]$ and $[\mathbf{u}_2]$ are then updated accordingly translating $[\tilde{\mathbf{u}}_1]$ and $[\tilde{\mathbf{u}}_2]$, i.e. $[\mathbf{u}_k] = [\tilde{\mathbf{u}}_k] + C^{-1}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_k)$.

5 Experiments

Experiments presented in this section cover a wide range of situations, from 2D implicit functions to higher dimensional systems. For comparing the usage of

[4] Actually, once the verification performed at Subsection 4.1.2 has succeeded, it will always succeed downstream the search tree.

parallelepiped domains and box domains, the volume of each enclosure will be compared. In order to have a dimension free measure, the *reduced-volume*, defined as the n^{th} root of the volume where n is the dimension of the problem, will be used. Note that the volume of a parallelepiped is simply the volume of its characteristic domain multiplied by the absolute value of the determinant of its characteristic matrix. The algorithms have been run on a Intel(R) Core(TM)2 Duo CPU at 2.20 GHz, with 4Gb of memory, under Windows XP.

5.1 Intersection of surfaces

The validated intersection between a sphere and a cylinder is modeled by a NCSP with 3 variables and 2 constraints. Both parallelepiped domains and box domains have been used to compute the enclosure of this geometrical object for comparison purpose. Figure 9 shows that, for the same number of bisections, the parallelepiped domains provide a much more accurate enclosure of the solution set (the volume of the parallelepipeds enclosure is 0.00078 while the volume of the boxes enclosure is 0.033). Furthermore, the solution set has been proved rigorously to cross each parallelepiped domain. That information is unavailable when using box domains.

5.2 Two dimensional implicit plot

The problem consists in determining the implicit graph of a complicated function proposed in [27]:

$$f(\mathbf{x}) = x_1 \cos(x_2) \cos(x_1 x_2) + x_2 \cos(x_1) \cos(x_1 x_2) + x_1 x_2 \cos(x_1) \cos(x_2). \qquad (19)$$

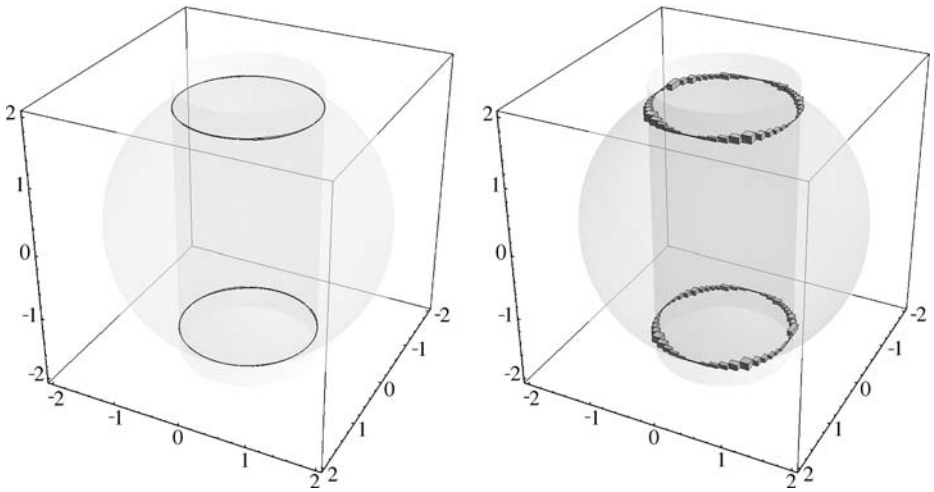

Fig. 9 Intersection of a sphere and a cylinder (both plotted in transparent for information). Enclosures obtained using parallelepiped domains (*left*) or with box domains (*right*) after 100 bisections

Note that in [27] each plus sign is actually consider as a plus/minus sign, hence leading to four different functions, while here we treat (19). After 72 seconds, the enclosure shown on Fig. 10 has been computed for $f(\mathbf{x}) = 0$. All the parallelepiped domains are rigorously proved to be crossed by the solution set, this latter being thus completely determined. Timings cannot be compared wrt. the algorithm proposed in [27] because the present approach provides much more information than [27], where a simple outer approximation is computed, exactly as accurate as the pixel size of the resolution.

5.3 The Layne-Watson exponential cosine curve

This system of 3 variables and 2 equations is taken from [17]:

$$f_k(\mathbf{x}) = x_k - x_{n+1} \exp\big(\cos\big(k \sum_{1 \le i \le n} x_i\big)\big) \tag{20}$$

with $\mathbf{x} \in \mathbb{R}^{n+1}$, $k \in \{1, \ldots, n\}$ and $n = 2$. This NCSP is an intersection of surfaces whose equations involve compositions of cosine and exponential functions. Once more, the parallelepiped domains provide a more accurate enclosure than box domains, and allow proving the existence of the whole manifold of solutions. For a more accurate comparison of performances, the left hand side graphic of Fig. 11 displays the time needed to obtain a given reduced-volume for both methods. At a first glance, the parallelepiped domains are much more efficient than box domains. Interpreting more precisely these log/log plots, both curves are almost lines. Therefore, both algorithms display a time increasing polynomially with the inverse of the reduced-volume. What is noteworthy is that the slopes of the two lines are different (corresponding to polynomials of different degree), which means that the parallelepiped domains improve not only the timings but also the complexity of the branch and prune algorithm.

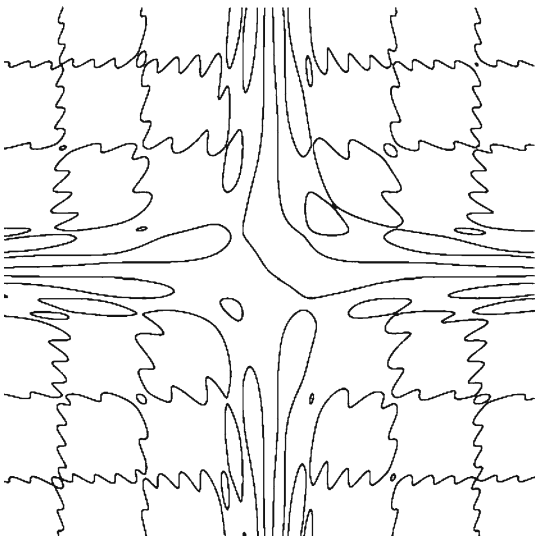

Fig. 10 Verified implicit plot of Tupper's function

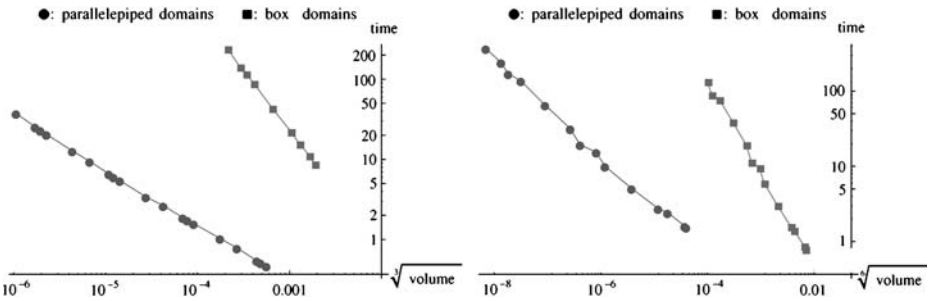

Fig. 11 Log/log plots of the reduced-volume of the enclosure against the time needed to compute this enclosure. *Left*: The Layne-Watson Exponential Cosine Curve. *Right*: The Parametrized Broyden Tridiagonal

5.4 The parametrized broyden tridiagonal

The Broyden tridiagonal problem is a well-known system of n equations and n unknowns [21]. The problem is changed to a parametric problem adding a variable α:

$$(\alpha - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1 = 0 \tag{21}$$

for $1 \leq i \leq n$, with $x_0 = x_{n+1} = 0$ for compact notations. The original value of α is 3, so we choose the domain $[2, 4]$ for this additional variable. As in the original problem, the domains of the other variables are $[-100, 100]$. Introducing such a parameter allows studying how the solutions of the original Broyden tridiagonal problem change with the variations of this parameter. Once more, the usage of parallelepiped domains drastically reduces the time needed to obtain an enclosure of a given reduced-volume, cf. Figure 11. And again, Fig. 11 shows that the time needed is approximately a polynomial of the inverse of the reduced-volume, while the usage of parallelepiped domains has reduced the degree of this polynomial.

6 Discussion

These first experiments show that indeed the global constraint contraction performed on parallelepiped domains drastically improves the resolution process (the degree of the polynomial time complexity to reach a given accuracy seems to have been reduced, leading to timings divided by more than 100). Furthermore, the proof of existence of solutions works well for non singular solutions, and allows giving a complete description of the solution set under the form of a sharp enclosure which is proved to contain solutions.

Under-constrained systems of equations appear in many contexts. An important part of the forthcoming work will be to include this framework in an efficient solver to be able to tackle real life problems, like robot generalized aspects computation [4]. Furthermore, parallelepiped domains may improve the efficiency of global optimization algorithms, which often have to solve under-constrained systems of equations. On a theoretical point of view, the introduction of universally quantified parameters

in this framework using techniques similar to [8] will allow tackling interesting problems, like the robust intersection of surfaces with uncertain parameters.

Acknowledgements This work was partially funded by the ANR grant number PSIROB06_174445. The authors are grateful to Arnold Neumaier for the useful discussions they had about the relationship between the preconditioned interval Newton operator and global constraints.

References

1. Benhamou, F., Goualard, F., Granvilliers, L., & Puget, J. F. (1999). Revising hull and box consistency. In *International conference on logic programming* (pp. 230–244).
2. Benhamou, F., McAllister, D., & Van Hentenryck, P. (1994). CLP(Intervals) revisited. In *International symposium on logic programming* (pp. 124–138).
3. Benhamou, F., & Older, W. J. (1997). Applying interval arithmetic to real, integer and boolean constraints. *Journal of Logic Programming, 32*(1), 1–24.
4. Chablat, D., & Wenger, Ph. (1998). Working modes and aspects in fully-parallel manipulator. In *IEEE international conference on robotics and automation* (pp. 1970–1976). Piscataway: IEEE.
5. Collavizza, H., Delobel, F., & Rueher, M. (1999). Comparing partial consistencies. *Reliable Computing, 1*, 1–16.
6. Goldberg, D. (1991). What every computer scientist should know about floating-point arithmetic. *Computing Surveys, 23*(1), 5–48.
7. Goldsztejn, A. (2005). A right-preconditioning process for the formal-algebraic approach to inner and outer estimation of AE-solution sets. *Reliable Computing, 11*(6), 443–478.
8. Goldsztejn, A. (2006). A branch and prune algorithm for the approximation of non-linear AE-solution sets. In *Proc. of ACM SAC 2006* (pp. 1650–1654).
9. Goldsztejn, A. (2008). *Sensitivity analysis using a fixed point interval iteration*. Technical Report hal-00339377, CNRS.
10. Goldsztejn, A., & Granvilliers, L. (2008). A new framework for sharp and efficient resolution of NCSP with manifolds of solution. In *Proceedings of CP 2008. LNCS* (Vol. 5202/2008, pp. 190–204).
11. Goldsztejn, A., & Hayes, W. (2006). Reliable inner approximation of the solution set to initial value problems with uncertain initial value. In *Proc. of SCAN 2006*.
12. Granvilliers, L., & Benhamou, F. (2006). RealPaver: An interval solver using constraint satisfaction techniques. *ACM Transactions on Mathematical Software, 32*(1), 138–156.
13. Hansen, E. (1992). *Global optimization using interval analysis*, Second Edn. New York: Marcel Dekker.
14. Hayes, B. (2003). A lucid interval. *American Scientist, 91*(6), 484–488.
15. Jaulin, L., Kieffer, M., Didrit, O., & Walter, E. (2001). *Applied interval analysis with examples in parameter and state estimation, robust control and robotics*. New York: Springer.
16. Kearfott, R. B. (1996). Interval computations: Introduction, uses, and resources. *Euromath Bulletin, 2*(1), 95–112.
17. Kearfott, R. B., & Xing, Z. (1994). an interval step control for continuation methods. *SIAM Journal on Numerical Analysis, 31*(3), 892–914.
18. Lhomme, O. (1993). Consistency techniques for numeric CSPs. In *Proceedings of IJCAI 1993* (pp. 232–238).
19. Merlet, J. P. (2000). *Parallel robots*. Dordrecht: Kluwer.
20. Moore, R. (1966). *Interval analysis*. Englewood Cliffs: Prentice-Hall.
21. Moré, J. J., Garbow, B. S., & Hillstrom, K. E. (1981). Testing unconstrained optimization software. *ACM Transactions on Mathematical Software, 7*(1), 136–140.
22. Nedialkov, N. S., Jackson, K. R., & Corliss, G. F. (1999). Validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation, 105*(1), 21–68.
23. Neumaier, A. (1987). Overestimation in linear interval equations. *SIAM Journal on Numerical Analysis, 24*(1), 207–214.
24. Neumaier, A. (1990). *Interval methods for systems of equations*. Cambridge: Cambridge University Press.

25. Ortega, J. M., & Rheinboldt, W. C. (2000). *Iterative solution of nonlinear equations in several variables*. Philadelphia: SIAM.
26. Rump, S. M. (1988). Rigorous sensitivity analysis for systems of linear and nonlinear equations. *Mathematics of Computation, 54*(10), 721–736.
27. Tupper, J. (2001). Reliable two-dimensional graphing methods for mathematical formulae with two free variables. In *SIGGRAPH '01* (pp. 77–86). New York: ACM.
28. Van Hentenryck, P., McAllester, D., & Kapur, D. (1997). Solving polynomial systems using a branch and prune approach. *SIAM Journal on Numerical Analysis, 34*(2), 797–827.