Computing real solutions of fuzzy polynomial systems

Philippe Aubry, Jérémy Marrez, Annick Valibouze

PUBLISHED IN FUZZY SETS AND SYSTEMS 399: 55–76 (2020)
DOI : 10.1016/J.FSS.2020.01.004

AUTHOR'S VERSION

Computing real solutions of fuzzy polynomial systems

Philippe Aubry[*1], **Jérémy Marrez**[†1], **Annick Valibouze**[‡1,2]

[1] *Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005 Paris, France*
[2] *Sorbonne Université, CNRS, Laboratoire de Probabilités, Statistique et Modélisation, LPSM, F-75005 Paris, France*

January 21, 2020
(*Accepted FSS: 16 January 2020*)

ABSTRACT. This paper presents an efficient algorithm called `SolveFuzzySystem`, or `SFS`, for finding real solutions of polynomial systems whose coefficients are symmetrical L-R fuzzy numbers with bounded support for which the spread functions L and R are bijective. The real solutions of such a system are deduced from solutions of some polynomial systems with real coefficients. This algorithm is based on new results that are universal because they are independent from the spread functions. These theoretical results include the management of the fuzzy system's solutions signs. An implementation in the Fuzzy package of the free computer algebra software SageMath and a parallel version of the algorithm are described.

Keywords: Fuzzy numbers; Fuzzy systems; Fuzzy mathematical programming; Polynomial systems

1. INTRODUCTION

Modeling problems with uncertain data has important applications in engineering, economics and social sciences [1, 12]. As fuzzy functions involved in equations can be approximated by fuzzy polynomials [2, 3], the problem of solving fuzzy polynomial systems is of main importance and has motivated many studies, among them [4, 5, 6, 7].

However, the question of the meaning of a solution was soon raised. In 1990, Buckley and Qu [4] conclude that too often the extension principle approach is too restrictive and that a new solution concept is needed so that a quadratic univariate equation has always a solution. Considering inclusion or non-empty intersection rather than equality of membership functions has led to a non-unique formulation. The question exists even for linear systems: Friedman et al. [8] thus distinguish between strong and weak fuzzy solutions, and various algorithms in later papers may give solutions with unclear meaning. As interval linear systems can be considered as a special case of fuzzy linear systems, interval analysis techniques have been exploited: Muzzioli and Reynaerts [9] thus study the existence of counterpart in

[*]philippe.aubry@sorbonne-universite.fr
[†]jeremy.marrez@sorbonne-universite.fr
[‡]annick.valibouze@sorbonne-universite.fr

the fuzzy linear case for different notions of solutions in interval linear systems solving. However, the methods based on these techniques face problems due to the improper intervals that may appear. In 2015, Lodwick and Dubois [10] investigate the semantics and the links between fuzzy linear systems and interval linear systems in a unified approach that clarifies anomalous or inconsistent results in previous papers.

Some problems are modeled by a system of continuous fuzzy valued functions defined over $\mathbb{R}^n$ (see [11]); Liu [2] proposes a polynomial interpolation that provides an interface between solving these problems and solving fuzzy polynomial systems. Therefore several authors have been interested in searching for real solutions of polynomial equations with fuzzy coefficients. Note that their work, and ours too, is based only on Zadeh's extension principle. The methods of resolution were initially based on local techniques, firstly for one univariate polynomial [13, 14, 15] and later for multivariate systems [16, 17]. Abbasbandy et al.[16] study in particular systems of the form

$$\begin{cases} \widetilde{a}_{1,1}xy + \widetilde{a}_{1,2}x^2y^2 + \cdots + \widetilde{a}_{1,d}x^dy^d &= \widetilde{a}_{1,0} \\ \widetilde{a}_{2,1}xy + \widetilde{a}_{2,2}x^2y^2 + \cdots + \widetilde{a}_{2,d}x^dy^d &= \widetilde{a}_{2,0} \end{cases}$$

and they present several systems coming from applications like cross location of quadratic surfaces or in economics. In Section 6.2 we detail our method on two of them as illustrations.

Recently, a global approach using classical algebraic techniques has been developed [18, 19, 20, 21]. Indeed, despite a name that may be confusing, fuzzy numbers benefit from a perfectly formal definition. We revisit this approach and we significantly strengthen it. In the past, both local and global approaches focused on so-called *triangular* fuzzy numbers, that is, with linear spread functions. The results presented here consider more generally symmetric L-R fuzzy coefficients such as, for example, quadratic fuzzy numbers, that have quadratic spread functions. Like previous works on this topic, our notion of solution lies on the equality of membership functions.

In a fuzzy algebraic system, the fuzzy coefficients coming from the experiments are generally L-R fuzzy numbers, given under a representation called "tuple". Although the tuple representation is formal, it cannot be handle by usual algebraic methods (Gröbner bases [22], triangular decomposition [23], rational univariate representation [24], ...) to solve the system. Nevertheless, the tuple representation of any L-R fuzzy number is transformable into another representation called "parametric", where the coefficients are no longer fuzzy but real. We give the expression of the parametric representation as a function of the inverse of its spread functions (see Proposition 2.2).

Throughout this paper we consider a fuzzy system (S) formed by s equations of k variables with L-R symmetrical fuzzy coefficients. We search its real solutions. Obtaining directly the whole set of real solutions of (S) raises a problem of signs since the coefficients are fuzzy. We overcome this problem by proceeding in two steps. Firstly, without using the assumption of symmetry on the coefficients, we apply our Theorem 3.4 to find positive real solutions of (S) which are those of its *real transform* $\mathscr{T}$(S), a new algebraic system formed by 3^s equations of k variables with real coefficients. We compare the real transform with the previous results mentioned above for systems restricted to triangular fuzzy coefficients. Secondly, we introduce a technical tool called *induced systems*; it allows to compute the whole set of real solutions of (S) from the positive solutions of 2^k fuzzy induced systems with our main Theorem 4.1. By applying these results and reducing the number of induced systems to solve, we propose an optimized parallelizable algorithm called `SolveFuzzySystem` or `SFS`. Note that in Section 3.6, by the same reasoning, we obtain the real transform of a polynomial system with more general L-R fuzzy coefficients called trapezoidal.

This paper is organized as follows: Section 2 introduces fuzzy numbers, the specific L-R fuzzy numbers and the required transformations on representations. Section 3 is devoted to real transforms and induced equations forming the induced systems. Section 4 applies the results of Section 3 to a fuzzy polynomial system. It establishes the main Theorem 4.1 which leads to a basic algorithm called `BA-SFS`. In Section 5, this latter is optimized in Algorithm `SFS` then parallelized. The implementation of `SFS` in the `SageMath Fuzzy` package [25] is briefly presented in Section 6, together with examples illustrating `SFS` and its parallelization.

2. FUZZY NUMBERS

After some generalities, this section recalls the parametric representation of any fuzzy number, and then the special L-R fuzzy numbers, that may be represented by a tuple. To go further on fuzzy quantities the reader may be interested in [26]. We give some formulas which express the parametric representation of a L-R fuzzy number in function of its

tuple representation when the number has a bounded support and bijective spread functions (Proposition 2.2). These formulas are the key of our algebraic method to solve in $\mathbb{R}$ the algebraic fuzzy systems.

2.1. **Generalities.** The concept of fuzzy sets generalizes the classical concept of characteristic function of a subset E of the universal set X: as depicted in Zadeh's seminal paper [27], a fuzzy set $\widetilde{\mathrm{E}}$ is a class of objects with a grade of membership. The grade of membership of an element x is $\mu_{\widetilde{\mathrm{E}}}(x)$, where $\mu_{\widetilde{\mathrm{E}}}$ is a function from X to $[0,1]$. $\widetilde{\mathrm{E}}$ is said *convex* if each of its *r-cuts*, defined as the set $\widetilde{\mathrm{E}}_r = \{x \in \mathrm{X} \mid \mu_{\widetilde{\mathrm{E}}}(x) \geq r\}$ for a real number r in $(0,1]$, is convex.

Fuzzy numbers are particular fuzzy sets. A fuzzy number is a convex fuzzy set whose membership function from $\mathbb{R}$ to the real interval $[0,1]$ is continuous and such that its *core* (the interval of values with membership 1) is reduced to a singleton. This latter requirement is added for the sake of clarity in our study and since most applications handle this kind of fuzzy numbers. Anyway, we will show in Section 3.6 that our analysis simply extends to fuzzy numbers with an interval as core that we call *trapezoidal fuzzy numbers*.

Let $n \in \mathbb{R}$. A fuzzy number with core $\{n\}$ will be denoted by $\widetilde{n}$. We define the *support* of $\widetilde{n}$ as the support of its membership function, i.e. the set of $a \in \mathbb{R}$ such that $\mu_{\widetilde{n}}(a) \neq 0$. We denote it by $\mathrm{Supp}(\widetilde{n})$. For a real number r in $(0,1]$, the *r-cut* of $\widetilde{n}$ is the convex set $\widetilde{n}_r = \{x \in \mathbb{R} \mid \mu_{\widetilde{n}}(x) \geq r\}$. The 0-cut $\widetilde{n}_0$ is the closure of $\mathrm{Supp}(\widetilde{n})$.

The *left restriction* (resp. *right restriction*) $\mu_{\widetilde{n}_-}$ (resp. $\mu_{\widetilde{n}_+}$) is the restriction of $\mu_{\widetilde{n}}$ to the left (resp. right) of n. Some families of fuzzy numbers can be characterized from the functions $\mu_{\widetilde{n}_-}$ and $\mu_{\widetilde{n}_+}$. The most popular family contains the *triangular* fuzzy numbers for which the functions $\mu_{\widetilde{n}_-}$ and $\mu_{\widetilde{n}_+}$ are linear on $\mathrm{Supp}(\widetilde{n})$. Similarly one speaks of fuzzy numbers which are *gaussian, quadratic,...*

Throughout the paper, we will consider fuzzy numbers with bounded support.

2.2. **Parametric representation.** The definition below is adapted from [29] where R. Goetschel and W. Voxman introduces the parametric representation of trapezoidal fuzzy numbers to embed them into a topological vector space.

Definition 2.1. The *parametric form* of a fuzzy number $\widetilde{n}$ is an ordered pair $[\underline{n}, \overline{n}]$ of functions from the real interval $[0,1]$ to $\mathbb{R}$ which satisfy the following conditions:

(i) $\overline{n}$ is a bounded left continuous non-increasing function on $[0,1]$,
(ii) $\underline{n}$ is a bounded left continuous non-decreasing function on $[0,1]$,
(iii) $\underline{n}(1) = \overline{n}(1) = n$.

The fuzzy number $\widetilde{n}$ defined by functions $\underline{n}$ and $\overline{n}$ has membership function $\mu_{\widetilde{n}} : \mathbb{R} \to [0,1]$ such that $\mu_{\widetilde{n}}(x) = \sup\{r \mid \underline{n}(r) \leq x \leq \overline{n}(r)\}$; for $r \in (0,1]$, the r-cut of $\widetilde{n}$ is $[\underline{n}(r), \overline{n}(r)]$.

An addition and a scalar multiplication defined by Zadeh extension principle structure the set of fuzzy numbers; Stefanini and Sorini [30] recall that they are equivalently expressed with parametric representation as described in the following lemma:

Lemma 2.1. *Let and $\widetilde{m} = [\underline{m}, \overline{m}]$ and $\widetilde{n} = [\underline{n}, \overline{n}]$ two fuzzy numbers and $q \in \mathbb{R} \setminus \{0\}$. Then*

(1) $\widetilde{m} = \widetilde{n}$ if and only if $\underline{m}(r) = \underline{n}(r)$ and $\overline{m}(r) = \overline{n}(r)$ for each real $r \in [0,1]$;
(2) $\widetilde{m} + \widetilde{n} = [\underline{m} + \underline{n},\ \overline{m} + \overline{n}]$;
(3) $q \cdot \widetilde{n} = \begin{cases} [q \cdot \underline{n},\ q \cdot \overline{n}] & \text{if } q > 0, \\ [q \cdot \overline{n},\ q \cdot \underline{n}] & \text{if } q < 0 \end{cases}$

where, for any function f from $\mathbb{R}$ to $\mathbb{R}$, the product $g = q \cdot f$ represents the function defined as $g(r) = q f(r)$ for each $r \in \mathbb{R}$.

2.3. L-R **fuzzy numbers and tuple representation.** The well known LR model was introduced by D. Dubois and H. Prade in [28]. It considers families of special fuzzy numbers in which the arithmetic operations are internal and express very simply. Inside a given family the fuzzy numbers are represented by a tuple representation recalled below.

A function H defined on the real interval $[0, +\infty)$ with values in the real interval $(-\infty, 1]$ is called a *spread function* if $\mathrm{H}(0) = 1$, $\mathrm{H}(1) = 0$, H is continuous and decreasing on its domain.

Definition 2.2. Let L and R be two spread functions. A fuzzy number $\widetilde{n}$ with a bounded support is said of *type* L-R, or is a L-R *fuzzy number*, if there exist two positive real numbers α and β such that the membership function $\mu_{\widetilde{n}}$ of $\widetilde{n}$ is

given as follows:

$$\mu_{\widetilde{n}}(x) = \begin{cases} \mathrm{L}\left(\frac{n-x}{\alpha}\right) & \text{for} \quad n-\alpha \le x < n \quad \text{when} \quad \alpha \neq 0 \\ 1 & \text{for} \quad x = n \\ \mathrm{R}\left(\frac{x-n}{\beta}\right) & \text{for} \quad n < x \le n+\beta \quad \text{when} \quad \beta \neq 0 \\ 0 & \text{for} \quad x \in (-\infty, n-\alpha) \cup (n+\beta, +\infty) \end{cases}$$

The triplet (n, α, β) is called the *tuple representation* of the L-R fuzzy number $\widetilde{n}$. Real numbers α and β are respectively called the *left spread* and the *right spread* of $\widetilde{n}$.

We denote by $\mathfrak{F}(\mathrm{L},\mathrm{R})$ the family of fuzzy numbers of type L-R. Functions L and R are respectively called the *left spread function* and the *right spread function* of the family $\mathfrak{F}(\mathrm{L},\mathrm{R})$, and by extension the spread functions of $\widetilde{n}$ itself.

Inside a given family $\mathfrak{F}(\mathrm{L},\mathrm{R})$, a tuple (n,α,β) represents a unique element $\widetilde{n}$. The addition of fuzzy numbers is an internal law in $\mathfrak{F}(\mathrm{L},\mathrm{R})$. The tuple representation of the sum of two L-R fuzzy numbers $\widetilde{n} = (n,\alpha,\beta)$ and $\widetilde{n'} = (n',\alpha',\beta')$ is given by:

$$\widetilde{n} + \widetilde{n'} = (n+n', \alpha+\alpha', \beta+\beta') \ .$$

For $\widetilde{n} \in \mathfrak{F}(\mathrm{L},\mathrm{R})$ the scalar product $q \cdot \widetilde{n}$ is in $\mathfrak{F}(\mathrm{L},\mathrm{R})$ if the real q is positive while it is $\mathfrak{F}(\mathrm{R},\mathrm{L})$ if q is negative. Its tuple representation is given by:

$$q \cdot \widetilde{n} = \begin{cases} (q\,n, q\,\alpha, q\,\beta) \in \mathfrak{F}(\mathrm{L},\mathrm{R}) & \text{if } q > 0 \\ (q\,n, -q\,\beta, -q\,\alpha) \in \mathfrak{F}(\mathrm{R},\mathrm{L}) & \text{if } q < 0 \end{cases} \quad . \tag{1}$$

The scalar product is an internal law of $\mathfrak{F}(\mathrm{L},\mathrm{R})$ only when $\mathrm{L} = \mathrm{R}$ [*]. The elements of $\mathfrak{F}(\mathrm{L},\mathrm{L})$ will be called *symmetrical* fuzzy numbers.

Note that the inversion of the spreads keeps them positive when $q < 0$: $-q\beta$ and $-q\alpha$ are respectively the left spread and the right spread of $q \cdot \widetilde{n}$. In particular, in $\mathfrak{F}(\mathrm{L},\mathrm{L})$ we have

$$-\widetilde{n} = (-n, \beta, \alpha) \quad . \tag{2}$$

Remark 2.3. When all the computations involve products $q \cdot (n,\alpha,\beta)$ where $q \ge 0$, there is no need that $\mathrm{L} = \mathrm{R}$.

2.4. **From tuple to parametric representation of** L-R **fuzzy numbers.** In this paper, we consider polynomials equations with coefficients that are fuzzy numbers of a same family $\mathfrak{F}(\mathrm{L},\mathrm{R})$ satisfying the sufficient requirement that the spread functions L and R are bijective. Our solving method implies to rewrite algebraically each fuzzy coefficient $\widetilde{n} \in \mathfrak{F}(\mathrm{L},\mathrm{R})$ from tuple representation (n,α,β) into parametric representation. The change of representation is given by formulas of Proposition 2.2 below.

The parametric representation of $\widetilde{n}$ is strongly related to its r-cuts $\widetilde{n}_r$ since functions $\underline{n}$ and $\overline{n}$ defined by

$$\underline{n}(r) = \inf_{r\in[0,1]} \widetilde{n}_r \quad \text{and} \quad \overline{n}(r) = \sup_{r\in[0,1]} \widetilde{n}_r$$

satisfy the requirements of Definition 2.1. This relation appears graphically in Figure 1 where $x_1 = \underline{n}(1/2)$ and $x_2 = \overline{n}(1/2)$ for a triangular fuzzy number $\widetilde{3} = (3,2,3)$. The graph of functions $\underline{n}$ and $\overline{n}$ is obtained by a plane rotation of the graph of the membership function followed by a vertical symmetry.

[*] To have $\mathfrak{F}(\mathrm{L},\mathrm{R}) = \mathfrak{F}(\mathrm{R},\mathrm{L})$, it is necessary that $\mathrm{L}(x) = \mathrm{R}(\frac{x}{c})$ with $c > 0$; then $\mathrm{L} = \mathrm{R}$ because $0 = \mathrm{L}(1) = \mathrm{R}(1)$ and R strictly monotonous.

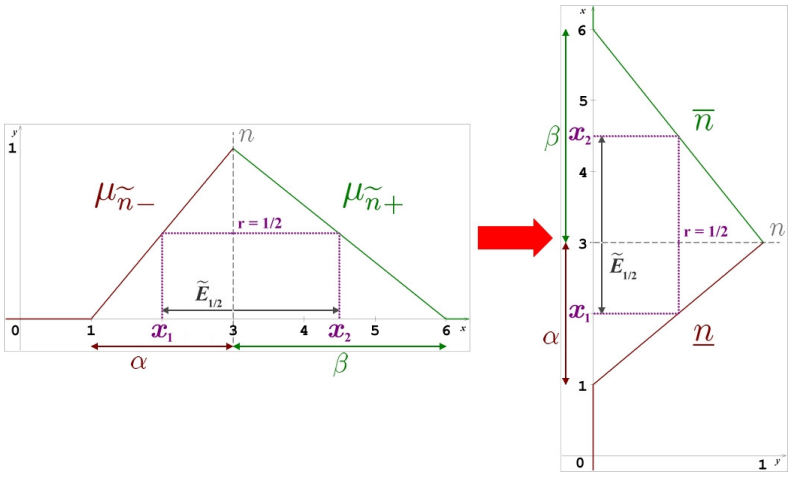

FIGURE 1. Graph of functions $\underline{3}$ and $\overline{3}$ from the graph of a linear membership function

Formally, the transformation is described by the formulas below.

Proposition 2.2. *Let $\widetilde{n} = (n, \alpha, \beta) \in \mathfrak{F}(L, R)$ where L and R are bijective spread functions. Then the parametric representation $[\underline{n}, \overline{n}]$ of $\widetilde{n}$ satisfies the following formulas:*

$$\begin{cases} \underline{n}(r) = n - \alpha L^{-1}(r) \\ \overline{n}(r) = n + \beta R^{-1}(r) \end{cases} . \tag{3}$$

In particular, when the fuzzy number $\widetilde{n}$ is triangular, we have:

$$\underline{n}(r) = \alpha\, r + n - \alpha \quad et \quad \overline{n}(r) = -\beta\, r + n + \beta \quad . \tag{4}$$

Proof. For the real number $r \in [0,1]$, Definition 2.2 implies $r = L\left(\frac{n-\underline{n}(r)}{\alpha}\right) = R\left(\frac{\overline{n}(r)-n}{\beta}\right)$. As L and R are bijective, $\underline{n}(r)$ and $\overline{n}(r)$ satisfy Identity (3) of the proposition.
In the triangular case, we obtain formula (4) because $L = R = F$ where $F(x) = 1 - x$ is bijective with $F^{-1} = F$. □

3. THE REAL TRANSFORM OF A FUZZY EQUATION

The goal of this paper is to solve in $\mathbb{R}$ a system of polynomial equations whose coefficients are symmetrical fuzzy numbers belonging to a same family $\mathfrak{F}(L, L)$ where L is bijective. It is performed by transforming independently each equation in order to obtain polynomial systems with real coefficients so that they can be solved by algebraic methods. This section is devoted to the transform of only one equation. Note that, in practice, we do not encounter one isolated multivariate equation. For a system reduced to a unique equation, the number of variable is generally reduced to only one too. This particular case can be treated with our method or by others such as [13], and recently [21], but it is not the purpose of this paper.

We will use the following terminology: a variable is said *real* if its represents any real number; a real variable is said *positive* if it represents any positive real number, i.e. belonging to $\mathbb{R}^+$; a k-uplet $(b_1, \ldots, b_k)$ of real variables or real numbers is said *positive* if each component b_i is positive. In this section we consider an algebraic equation (E) with symmetric L-R fuzzy coefficients and k real variables $x_1, \ldots, x_k$ also called the indeterminates.

The problem when computing with real variables and fuzzy numbers comes from the fuzzy numbers expressed as a product $q \cdot \widetilde{n}$ because the spreads depend on the sign of $q \in \mathbb{R}$ (see Lemma 2.1). In the fuzzy equation (E) each monomial $m = x_1^{d_1} \cdots x_k^{d_k}$ ($d_i \in \mathbb{N}$) whose sign is generally a priori unknown, plays the role of q; note that if each exponent d_i is even then m is positive. When the k-uplet $\boldsymbol{x} = (x_1, \ldots, x_k)$ is positive, the monomial m is necessary positive.

To avoid this sign problem, Section 3.2 seeks to obtain the real solutions of (E) in $\mathbb{R}^k$ from *positive* solutions in $\mathbb{R}^k$ of 2^k auxiliary fuzzy equations E(I), where $I \in \{-1, 1\}^k$. We called the latter equations the *induced* equations of E.

In Section 3.3 and 3.4, as we deal only with positive real variables, the fuzzy coefficients do not need to be symmetrical but may lie more generally in a unique family $\mathfrak{F}(L,R)$ where L and R are bijective. In Section 3.3 a *crisp form* of (E) is constructed in order to deduce a *collected crisp form* of (E) ; in other words, an algebraic system of equations with real coefficients whose positive solutions are those of (E). The *collected crisp form* is formed by four equations by an algorithm that applies only when the fuzzy coefficients are triangular. Moreover Section 3.4 establishes a formula that provides a particular collected crisp form of (E) formed by only three equations. We call it the *real transform* of (E) and denote it by $\mathscr{T}(E)$.

To obtain the real solutions of (E), it is necessary and sufficient to collect the positive real solutions of the 2^k real transforms $\mathscr{T}(E(I))$, where E(I) is an induced fuzzy equation of (E). In practice, the equations E(I) are not pairwise distinct and it is not necessary to solve each of the 2^k systems $\mathscr{T}(E(I))$. This subject will be the center of the discussion of Section 4.

Section 3.5 compares the real transform $\mathscr{T}(E)$ to the usual collected crisp form given in literature for the triangular case. Section 3.6 finally generalizes the results to trapezoidal fuzzy numbers.

3.1. **Preliminaries.** Let $\boldsymbol{d} = (d_1,\dots,d_k) \in \mathbb{N}^k$, $\boldsymbol{x} = (x_1,\dots,x_k)$ and $\boldsymbol{x}^{\boldsymbol{d}} = x_1^{d_1}\cdots x_k^{d_k}$ the *monomial* of multidegree $\boldsymbol{d}$ in the variables $x_1,\dots,x_k$. In the same way, for $\boldsymbol{a} = (a_1,\dots,a_k) \in \mathbb{R}^k$, we denote by $\boldsymbol{a}^{\boldsymbol{d}}$ the product $a_1^{d_1}\cdots a_k^{d_k}$. For $\boldsymbol{y} = (y_1,\dots,y_k)$, we denote by $\boldsymbol{x}\times\boldsymbol{y}$ the classical product $(x_1y_1,\dots,x_ky_k)$. Note that $(\boldsymbol{x}\times\boldsymbol{y})^{\boldsymbol{d}} = \boldsymbol{x}^{\boldsymbol{d}}\boldsymbol{y}^{\boldsymbol{d}}$.

In this section, we consider the polynomial equation

$$(E): \sum_{\boldsymbol{d}\in \mathrm{Expon(E)}} \widetilde{n_{\boldsymbol{d}}}\,\boldsymbol{x}^{\boldsymbol{d}} = \widetilde{m} \quad , \tag{5}$$

where the coefficients are given under their respective tuple representations: $\widetilde{m} = (m,\alpha,\beta)$ and $\widetilde{n_{\boldsymbol{d}}} = (n_{\boldsymbol{d}},\alpha_{\boldsymbol{d}},\beta_{\boldsymbol{d}}) \neq (0,0,0)$ for each d in Expon(E), a finite subset of $\mathbb{N}^k$. For example, when $k=3$ and (E): $\widetilde{3}\,x_1^2x_2 + \widetilde{1}\,x_3^4 = \widetilde{6}$, Expon(E) is the subset $\{(2,1,0),(0,0,4)\}$ of $\mathbb{N}^3$.

When we search for positive solutions of (E) only, following Case $q \geq 0$ in Identities (1), there is no need to impose L = R: $\widetilde{m} \in \mathfrak{F}(L,R)$ and $\widetilde{n_{\boldsymbol{d}}} \in \mathfrak{F}(L,R)\backslash\{(0,0,0)\}$, $d \in$ Expon(E). We denote by $\mathrm{Sol}^+(E)$ the set of positive real solutions of (E):

$$\mathrm{Sol}^+(E) = \{\,\boldsymbol{a} \in {\mathbb{R}^+}^k \mid \sum_{\boldsymbol{d}\in \mathrm{Expon(E)}} \widetilde{n_{\boldsymbol{d}}}\,\boldsymbol{a}^{\boldsymbol{d}} = \widetilde{m}\,\} \quad .$$

Following Case $q \leq 0$ in Identities (1), for solutions in the whole set of real numbers, the coefficients are necessary symmetrical fuzzy numbers: $\widetilde{m} \in \mathfrak{F}(L,L)$ and $\widetilde{n_{\boldsymbol{d}}} \in \mathfrak{F}(L,L)\backslash\{(0,0,0)\}$, $d \in$ Expon(E). We denote by Sol(E) the set of real solutions of (E):

$$\mathrm{Sol}(E) = \{\,\boldsymbol{a} \in \mathbb{R}^k \mid \sum_{\boldsymbol{d}\in \mathrm{Expon(E)}} \widetilde{n_{\boldsymbol{d}}}\,\boldsymbol{a}^{\boldsymbol{d}} = \widetilde{m}\,\} \quad .$$

We search for Sol(E) by using the r-cuts in order to obtain an algebraic system with real coefficients that can be solved with classical computer algebra methods.

However, according to Identity (3) of Lemma 2.1, the r-cut of $\widetilde{n_{\boldsymbol{d}}}\boldsymbol{x}^{\boldsymbol{d}}$ depends on the sign of $\boldsymbol{x}^{\boldsymbol{d}}$ while this sign is as unknown as those of the indeterminates $x_1,\dots,x_k$.

This is why we need to consider the case where the indeterminates $x_1,\dots,x_k$ are real and positive. But we want to obtain the whole set of real zeros of (E), not only positive zeros. This is the discussion of Section 3.2. This section aims to construct Sol(E) from the 2^k sets of positive real zeros $\mathrm{Sol}^+(E(I))$ of the induced fuzzy equations E(I), where I runs throughout the k-uplets of $\{-1,1\}^k$.

After Section 3.2, it will remain to know how to calculate the 2^k sets $\mathrm{Sol}^+(E(I))$. For this perspective, Section 3.3 and Section 3.4 suppose that every variable is real and positive. In this context, we seek the formula of the real transform $\mathscr{T}(E)$ of (E). The positive solutions of the real algebraic system $\mathscr{T}(E)$ are exactly the positive solutions of the fuzzy algebraic equation (E). In these two sections the fuzzy equation (E) plays the role of each of its induced equation E(I).

3.2. **Solutions of** (E) **in function of positive solutions of its induced equations.** Let $\boldsymbol{a} = (a_1,\dots,a_k) \in \mathbb{R}^k$. We put $|\boldsymbol{a}| = (|a_1|,\dots,|a_k|)$ where $|b|$ denotes the absolute value of $b \in \mathbb{R}$. Note that $|\boldsymbol{a}^{\boldsymbol{d}}| = |\boldsymbol{a}|^{\boldsymbol{d}}$. The k-uplet of signs of the

components of $\boldsymbol{a}$ is $\varepsilon(\boldsymbol{a}) = (\text{sign}(a_1), \ldots,$
$\text{sign}(a_k)) \in \{-1,1\}^k$. The value $\varepsilon(\boldsymbol{a})^{\boldsymbol{d}} = \prod_{i=1}^{k} \text{sign}(a_i)^{d_i}$ is the sign 1 or -1 of the real $\boldsymbol{a}^{\boldsymbol{d}}$ and we have

$$\boldsymbol{a}^{\boldsymbol{d}} = \varepsilon(\boldsymbol{a})^{\boldsymbol{d}} \, |\boldsymbol{a}^{\boldsymbol{d}}| = \varepsilon(\boldsymbol{a})^{\boldsymbol{d}} \, |\boldsymbol{a}|^{\boldsymbol{d}} \quad .$$

The evaluation in the value $\boldsymbol{a}$ of a term $\widetilde{n_{\boldsymbol{d}}}\boldsymbol{x}^{\boldsymbol{d}}$ consists in replacing the monomial $\boldsymbol{x}^{\boldsymbol{d}}$ by its value $\boldsymbol{a}^{\boldsymbol{d}} = \varepsilon(\boldsymbol{a})^{\boldsymbol{d}} \, |\boldsymbol{a}|^{\boldsymbol{d}}$. According to Identity (3) of Lemma 2.1, when the sign $\varepsilon(\boldsymbol{a})^{\boldsymbol{d}}$ of $\boldsymbol{a}^{\boldsymbol{d}}$ is 1, the r-cut of the fuzzy number $\widetilde{n_{\boldsymbol{d}}}\,\boldsymbol{a}^{\boldsymbol{d}}$ is

$$[\, \underline{n_{\boldsymbol{d}}}(r) \, |\boldsymbol{a}|^{\boldsymbol{d}} \, , \, \overline{n_{\boldsymbol{d}}}(r) \, |\boldsymbol{a}|^{\boldsymbol{d}} \,] \, ,$$

and if this sign is -1 then the r-cut is

$$[\, -\overline{n_{\boldsymbol{d}}}(r) \, |\boldsymbol{a}|^{\boldsymbol{d}} \, , \, -\underline{n_{\boldsymbol{d}}}(r) \, |\boldsymbol{a}|^{\boldsymbol{d}} \,] \quad .$$

It is therefore impossible to transform $\widetilde{n_{\boldsymbol{d}}}x^{\boldsymbol{d}}$ into its parametric representation without knowing the sign of $x^{\boldsymbol{d}}$. Our idea to work around this problem of unknown sign of $\boldsymbol{x}^{\boldsymbol{d}}$ is to introduce an artificial k-uplet $\mathrm{I} \in \{-1,1\}^k$ for the signs of the indeterminates. Then the real $\mathrm{I}^{\boldsymbol{d}}$ can be mixed in with the coefficient $\widetilde{n_{\boldsymbol{d}}}$ and $\boldsymbol{x}$ can be supposed to be positive. It leads to construct from (E) 2^k fuzzy equations defined below.

Definition 3.1. Let (E) be algebraic equation with k variables and symmetric fuzzy coefficients given by (5). The *induced equations* of (E) are the following equations

$$\text{(6)} \qquad \mathrm{E(I)} \quad : \quad \sum_{\boldsymbol{d} \in \mathrm{Expon(E)}} \mathrm{I}^{\boldsymbol{d}} \, \widetilde{n_{\boldsymbol{d}}} \, \boldsymbol{x}^{\boldsymbol{d}} = \widetilde{m} \qquad \text{where} \quad \mathrm{I} \in \{-1,1\}^k.$$

Note that (E) = E((1,1,…,1)). In E(I), the k-uplet I plays the role of the k-uplet $\varepsilon(\boldsymbol{a})$ of the signs of a possible solution $\boldsymbol{a}$ of (E). For each term $\mathrm{I}^{\boldsymbol{d}} \, \widetilde{n_{\boldsymbol{d}}} \, \boldsymbol{x}^{\boldsymbol{d}}$ in the left side of E(I), $\mathrm{I}^{\boldsymbol{d}} \in \{-1,1\}$ plays the role of the sign $\varepsilon(\boldsymbol{a})^{\boldsymbol{d}}$ of $\boldsymbol{a}^{\boldsymbol{d}}$ while the sign of $\boldsymbol{x}^{\boldsymbol{d}}$ itself is considered as positive. In E(I), there is no problem to express the r-cuts of $\mathrm{I}^{\boldsymbol{d}} \, \widetilde{n_{\boldsymbol{d}}} \, \boldsymbol{x}^{\boldsymbol{d}}$ with $\boldsymbol{x}$ positive whereas it is impossible in (E) for $\widetilde{n_{\boldsymbol{d}}} \, \boldsymbol{x}^{\boldsymbol{d}}$ with $\boldsymbol{x}$ not necessarily positive.

To describe relationships between the solutions of equations (E) and E(I), for $\boldsymbol{q} = (q_1, \ldots, q_k) \in \mathbb{R}^k$ and $\mathrm{S} \subset \mathbb{R}^k$, we introduce the following notation:

$$\boldsymbol{q} \otimes \mathscr{S} \; = \; \{\, \boldsymbol{q} \times \boldsymbol{a} = (q_1 a_1, \ldots, q_k a_k) \mid \boldsymbol{a} = (a_1, \ldots, a_k) \in \mathscr{S} \,\} \quad .$$

Lemma 3.1. *Let* $\mathrm{I} \in \{-1,1\}^k$ *and* E(I) *be an induced equation of* (E). *Then*

$$\mathrm{Sol(E)} = \mathrm{I} \otimes \mathrm{Sol(E(I))} \quad \text{and} \quad \mathrm{Sol(E(I))} = \mathrm{I} \otimes \mathrm{Sol(E)} \, .$$

Proof. As $\mathrm{I} \times \mathrm{I} = (1, \ldots, 1)$, it is sufficient to prove first equality. For $\boldsymbol{b} \in \mathbb{R}^k$ a solution of E(I), we have

$$\widetilde{m} = \sum_{\boldsymbol{d}} \mathrm{I}^{\boldsymbol{d}} \, \widetilde{n_{\boldsymbol{d}}} \, \boldsymbol{b}^{\boldsymbol{d}} = \sum_{\boldsymbol{d}} \widetilde{n_{\boldsymbol{d}}} \, (\mathrm{I} \times \boldsymbol{b})^{\boldsymbol{d}} \, .$$

Hence $\mathrm{I} \times \boldsymbol{b}$ is a solution of (E).
Conversely, let $\boldsymbol{a} \in \mathrm{Sol(E)}$. Then $\boldsymbol{b} = \mathrm{I} \times a$ is a solution of E(I) since

$$\widetilde{m} = \sum_{\boldsymbol{d}} \widetilde{n_{\boldsymbol{d}}} \, (\mathrm{I} \times \mathrm{I} \times \boldsymbol{a})^{\boldsymbol{d}} = \sum_{\boldsymbol{d}} \mathrm{I}^{\boldsymbol{d}} \, \widetilde{n_{\boldsymbol{d}}} \, (\mathrm{I} \times \boldsymbol{a})^{d} \, .$$

□

For each $\boldsymbol{a} \in \mathbb{R}^k$, there exists $\mathrm{I} \in \{-1,1\}^k$ such that $\varepsilon(\boldsymbol{a}) = \mathrm{I}$ and for which the evaluation of E(I) in $|\boldsymbol{a}|$ is identical to the evaluation of (E) in $\boldsymbol{a}$. The question of finding all solutions of (E) from the positive solutions of its induced fuzzy equations E(I) is solved by the following fundamental theorem:

Theorem 3.2. *Consider an algebraic equation* (E) *with symmetrical fuzzy coefficients. Then the set of real solutions of* (E) *is formed by the* $\mathrm{I} \times \boldsymbol{b}$ *where* I *runs throughout the set of* k*-uplets* $\{-1,1\}^k$ *and* $\boldsymbol{b}$ *runs throughout the set of positive solutions of* E(I). *In other words,*

$$\mathrm{Sol(E)} = \bigcup_{\mathrm{I} \in \{-1,1\}^k} \mathrm{I} \otimes \mathrm{Sol}^{+}(\mathrm{E(I)}) \, .$$

Proof. It follows from Lemma 3.1 that each $\mathrm{I} \otimes \mathrm{Sol}^+(\mathrm{E}(\mathrm{I})) \subset \mathrm{Sol}(\mathrm{E})$. Conversely, if $\boldsymbol{a} \in \mathrm{Sol}(\mathrm{E})$ then $\boldsymbol{a}^{\boldsymbol{d}} = \varepsilon(\boldsymbol{a})^{\boldsymbol{d}}\,|\boldsymbol{a}^{\boldsymbol{d}}| = \varepsilon(\boldsymbol{a})^{\boldsymbol{d}}\,|\boldsymbol{a}|^{\boldsymbol{d}}$. It follows that

$$\widetilde{m} = \sum_{\boldsymbol{d}} \boldsymbol{a}^{\boldsymbol{d}}\,\widetilde{n_{\boldsymbol{d}}} = \sum_{\boldsymbol{d}} \varepsilon(\boldsymbol{a})^{\boldsymbol{d}}\,|\boldsymbol{a}|^{\boldsymbol{d}}\,\widetilde{n_{\boldsymbol{d}}}\,.$$

Putting $\mathrm{I} = \varepsilon(\boldsymbol{a})$ and $\boldsymbol{b} = |\boldsymbol{a}| \in \mathbb{R}^{+k}$, we obtain $\mathrm{I} \times \boldsymbol{b} = \varepsilon(\boldsymbol{a}) \times (\varepsilon(\boldsymbol{a}) \times \boldsymbol{a}) = \boldsymbol{a}$ where $\boldsymbol{b}$ is a solution of $\mathrm{E}(\mathrm{I})$. □

Remark 3.2. *The fuzzy symmetrical coefficients in* (E) *being given by their tuple representation, the induced equations* E(I) *actually derive directly from* (E). *For every* $\boldsymbol{d}$ *in Expon*(E) *the coefficient* $\mathrm{I}^{\boldsymbol{d}}\,\widetilde{n_{\boldsymbol{d}}}$ *is equal to* $\widetilde{n_{\boldsymbol{d}}}$ *itself when* $\mathrm{I}^{\boldsymbol{d}} = 1$, *otherwise following* (2), *it is equal to* $(-n_d, \beta_d, \alpha_d)$ *when* $\mathrm{I}^{\boldsymbol{d}} = -1$. *This ensures that for finding the whole set of real solutions of* (E) *it is sufficient to be able to compute positives real solutions of any fuzzy equation.*

In practice, the 2^k fuzzy equations E(I) are not always pairwise distinct. In particular, in case of each component d_i of $\boldsymbol{d} = (d_1, \ldots, d_k) \in \mathrm{Expon}(\mathrm{E})$ being even, all induced equations E(I) are identical to equation (E).

Example 3.1. Let $\widetilde{3}, \widetilde{1}, \widetilde{6} \in \mathfrak{F}(\mathrm{L}, \mathrm{L})$. Take $k = 3$ variables x_1, x_2, x_3 and consider the equation

$$(\mathrm{E}): \quad \widetilde{3}\,x_1^2 x_2 + \widetilde{1}\,x_3^4 = \widetilde{6} \quad .$$

The $8 = 2^3$ triplets of $\{-1,1\}^3$ are: $\mathrm{I}_1 = (1,1,1), \mathrm{I}_2 = (1,1,-1), \mathrm{I}_3 = (-1,1,1), \mathrm{I}_4 = (-1,1,-1), \mathrm{I}_5 = (1,-1,1), \mathrm{I}_6 = (1,-1,-1), \mathrm{I}_7 = (-1,-1,1)$ and $\mathrm{I}_8 = (-1,-1,-1)$. There are only two distinct induced equations $\mathrm{E}(\mathrm{I}_j)$. Indeed, since $\mathrm{Expon}(\mathrm{E}) = \{\boldsymbol{d}_1 = (2,1,0), \boldsymbol{d}_2 = (0,0,4)\}$, we have $\mathrm{I}_j^{\boldsymbol{d}_2} = 1$ for all $j = 1, \ldots, 8$. Thus the eight I_j split up into the two following groups: that for $j = 1, \ldots, 4$ with $\mathrm{I}_j^{\boldsymbol{d}_1} = 1$ and that for $j = 5, \ldots, 8$ with $\mathrm{I}_j^{\boldsymbol{d}_1} = -1$. Hence $(\mathrm{E}) = \mathrm{E}(\mathrm{I}_j)$ for $j = 1, \ldots, 4$ and $\mathrm{E}(\mathrm{I}_5) = \mathrm{E}(\mathrm{I}_j)$ for $j = 5, \ldots, 8$.

For $j = 5, \ldots, 8$, the tuple representation of the coefficients of $\mathrm{E}(\mathrm{I}_j)$ is determined with the help of Remark 3.2: with $\widetilde{3} = (3, \alpha_1, \beta_1)$, $\widetilde{1} = (1, \alpha_2, \beta_2)$ and $\widetilde{6} = (6, \alpha, \beta)$, we have $-\widetilde{3} = (-3, \beta_1, \alpha_1)$ and

$$\mathrm{E}(\mathrm{I}_j) : (-3, \beta_1, \alpha_1) x_1^2 x_2 + (1, \alpha_2, \beta_2) x_3^4 = (6, \alpha, \beta).$$

According to Theorem 3.2, we have

$$\mathrm{Sol}(\mathrm{E}) = \bigcup_{j=1}^{4} \mathrm{I}_j \otimes \mathrm{Sol}^+(\mathrm{E}(\mathrm{I}_1)) \qquad \bigcup \qquad \bigcup_{j=5}^{8} \mathrm{I}_j \otimes \mathrm{Sol}^+(\mathrm{E}(\mathrm{I}_5)) \quad .$$

The following sections are devoted to finding the positive solutions of (E) that we will apply to each equation E(I), taking into account Remark 3.2 illustrated in previous example.

3.3. **Crisp form of** (E) **to find** $\mathrm{Sol}^+(\mathrm{E})$**.** In this section and the following one, we express the positive solutions of (E). Since there is no need to assume L = R, we consider more generally that the fuzzy coefficients all lie in a unique family $\mathfrak{F}(\mathrm{L}, \mathrm{R})$ where L and R are bijective.

As explained above, algebraic solving of fuzzy equation (E) is usually based on the passage of the L-R representation of fuzzy numbers to their parametric representation. In the presentation below we significally strengthen this classical method for triangular fuzzy coefficients by applying it to a generic system and by extending it to more general fuzzy coefficients.

Following Lemma 2.1, equation (E) rewrites into two equalities on the r-cuts of its left and right members if all the $\boldsymbol{x}^{\boldsymbol{d}}$, $\boldsymbol{d} \in \mathrm{Expon}(\mathrm{E})$, are supposed to represent reals of the same sign. Indeed, according to Rule (3) of this lemma, the multiplication of a fuzzy number by a scalar q splits into two cases: $q \le 0$ and $q \ge 0$. Thus we search only for the solutions $\boldsymbol{a} \in \mathbb{R}^{+k}$ since the real $q := \boldsymbol{a}^{\boldsymbol{d}}$ is then positive for each $\boldsymbol{d} \in \mathbb{N}^k$. We will apply the results of this section to every induced equation E(I).

According to Lemma 2.1, the following equivalence applies to any $\boldsymbol{a} \in \mathbb{R}^{+k}$:

$$(7) \qquad \boldsymbol{a} \in \mathrm{Sol}^+(\mathrm{E}) \iff \left[\sum_{\boldsymbol{d} \in \mathrm{Expon}(\mathrm{E})} \underline{n_{\boldsymbol{d}}}(r)\,\boldsymbol{a}^{\boldsymbol{d}}, \sum_{\boldsymbol{d} \in \mathrm{Expon}(\mathrm{E})} \overline{n_{\boldsymbol{d}}}(r)\,\boldsymbol{a}^{\boldsymbol{d}} \right] = \left[\underline{m}(r), \overline{m}(r) \right] \quad .$$

This equivalence leads us to consider the following system $\mathscr{C}(\mathrm{E})$ of two equations with real coefficients and $k+1$ variables $x_1,\dots,x_k,r$, called the *crisp form* of (E) :

$$\mathscr{C}(\mathrm{E}):\begin{cases}\sum\limits_{\boldsymbol{d}\in\mathrm{Expon(E)}}\underline{n_{\boldsymbol{d}}}(r)\,\boldsymbol{x}^{\boldsymbol{d}} &= \underline{m}(r)\\ \sum\limits_{\boldsymbol{d}\in\mathrm{Expon(E)}}\overline{n_{\boldsymbol{d}}}(r)\,\boldsymbol{x}^{\boldsymbol{d}} &= \overline{m}(r)\end{cases}.$$

Let F be a finite set of equations in $\mathbb{R}[x_1,\dots,x_k,r]$. We put

$$\mathrm{Sol}^+_k(\mathrm{F})=\{\boldsymbol{a}\in\mathbb{R}^{+k}\mid\forall r\in[0,1]\ (a_1,\dots,a_k,r)\in\mathrm{Sol(F)}\}$$

where Sol(F) is the set of the solutions of F in $\mathbb{R}^{k+1}$. Take $\boldsymbol{a}=(a_1,\dots,a_k)\in\mathbb{R}^{+k}$. According to Equivalence (7), the k-uplet $\boldsymbol{a}$ is a solution of (E) if and only if for all real $r\in[0,1]$ the $(k+1)$-uplet $(a_1,\dots,a_k,r)$ is a solution of the crisp form $\mathscr{C}(\mathrm{E})$. In other words, $\mathrm{Sol}^+_k(\mathscr{C}(\mathrm{E}))$ is the set of the positive solutions of the fuzzy equation (E):

$$\mathrm{Sol}^+(\mathrm{E})=\mathrm{Sol}^+_k(\mathscr{C}(\mathrm{E}))\quad. \tag{8}$$

Example 3.2. Let us take again the fuzzy equation (E) of Example 3.1 with triangular fuzzy coefficients. We recall that their r-cuts are given by the formulas (4) and that $\mathrm{Expon(E(I_1))}=\mathrm{Expon(E(I_5))}=\{(2,1,0),(0,0,4)\}$. The parametric representations of the coefficients of the two induced equations are then written as follows:

- $\widetilde{n_{\boldsymbol{d}_1}}=\widetilde{3}=[\alpha_1 r+3-\alpha_1,-\beta_1 r+3-\beta_1]$ for $\mathrm{E(I_1)}$;
- $\widetilde{n_{\boldsymbol{d}_1}}=-\widetilde{3}=[\beta_1 r-3-\beta_1,-\alpha_1 r-3-\alpha_1]$ for $\mathrm{E(I_5)}$;
- $\widetilde{n_{\boldsymbol{d}_2}}=\widetilde{1}=[\alpha_2 r+1-\alpha_2,-\beta_2 r+1-\beta_2]$ and $\widetilde{6}=[\alpha r+6-\alpha,-\beta r+6+\beta]$ for $\mathrm{E(I_1)}$ and $\mathrm{E(I_5)}$.

Therefore the respective crisp forms of the two distincts induced equation of (E) are:

$$\mathscr{C}(\mathrm{E(I_1)}):\begin{cases}(\alpha_1 r+3-\alpha_1)\,x_1^2x_2 &+ &(\alpha_2 r+1-\alpha_2)\,x_3^4 &=\alpha r+6-\alpha\\ (-\beta_1 r+3-\beta_1)\,x_1^2x_2 &+ &(-\beta_2 r+1-\beta_2)\,x_3^4 &=-\beta r+6+\beta\end{cases}\quad\text{and}$$

$$\mathscr{C}(\mathrm{E(I_5)}):\begin{cases}(\beta_1 r-3-\beta_1)\,x_1^2x_2 &+ &(\alpha_2 r+1-\alpha_2)\,x_3^4 &=\alpha r+6-\alpha\\ (-\alpha_1 r-3-\alpha_1)\,x_1^2x_2 &+ &(-\beta_2 r+1-\beta_2)\,x_3^4 &=-\beta r+6+\beta\end{cases}.$$

In the particular triangular case, illustrated in Example 3.2 above, the crisp form has two equations with linear expressions w.r.t. the variable r in each side. It is a consequence of formulas (4) restricted to the particular triangular case. The triangular case is easy because the spread functions are equal to $\mathrm{F}:x\mapsto 1-x$, with $\mathrm{F}^{-1}=\mathrm{F}$. The following theorem deals with 2 indeterminate in the general case, when the spread functions are simply bijective.

Theorem 3.3. *Let* L *and* R *be two spread functions and*

$$(\mathrm{E}):\sum_{\boldsymbol{d}\in Expon(\mathrm{E})}\widetilde{n_{\boldsymbol{d}}}\,\boldsymbol{x}^{\boldsymbol{d}}=\widetilde{m}\,,$$

be a fuzzy equation with coefficients in the family $\mathfrak{F}(\mathrm{L},\mathrm{R})$ *given by their tuple representations as follows:* $\widetilde{m}=(m,\alpha,\beta)$ *and* $\widetilde{n_{\boldsymbol{d}}}=(n_{\boldsymbol{d}},\alpha_{\boldsymbol{d}},\beta_{\boldsymbol{d}})$ *for* $\boldsymbol{d}\in Expon(\mathrm{E})$. *If the spread functions* L *and* R *are bijective then the crisp form of* (E) *is given by:*

$$\mathscr{C}(\mathrm{E}):\begin{cases}\sum\limits_{\boldsymbol{d}} n_{\boldsymbol{d}}\,\boldsymbol{x}^{\boldsymbol{d}}-m+(\alpha-\sum\limits_{\boldsymbol{d}}\alpha_{\boldsymbol{d}}\,\boldsymbol{x}^{\boldsymbol{d}})\,u &= 0\\ \sum\limits_{\boldsymbol{d}} n_{\boldsymbol{d}}\,\boldsymbol{x}^{\boldsymbol{d}}-m+(-\beta+\sum\limits_{\boldsymbol{d}}\beta_{\boldsymbol{d}}\,\boldsymbol{x}^{\boldsymbol{d}})\,v &= 0\,,\end{cases} \tag{9}$$

where $u=\mathrm{L}^{-1}(r)$ *and* $v=\mathrm{R}^{-1}(r)$ *for all* $r\in[0,1]$. *For* $\boldsymbol{a}\in\mathbb{R}^{+k}$, *we have* $\boldsymbol{a}\in\mathrm{Sol}^+(\mathrm{E})$ *if and only if, for all* $r\in[0,1]$, *system* (9) *is satisfied by the* $(k+2)$*-uplet* $(a_1,\dots,a_k,\mathrm{L}^{-1}(r),\mathrm{R}^{-1}(r))$.

Proof. By definition, a spread function H sends $[0,1]$ to itself and if moreover H is bijective then its inverse H^{-1} is continuous and decreasing with $\mathrm{H}^{-1}(1)=0$ and $\mathrm{H}^{-1}(0)=1$. Suppose that the spread functions L and R are bijective. As each r belongs to $[0,1]$, we can put $u=\mathrm{L}^{-1}(r)$ and $v=\mathrm{R}^{-1}(r)$. When r runs throughout $[0,1]$ in the increasing sens, the parameters u and v run throughout the same interval $[0,1]$ in the decreasing sens by definition of the spread functions. Thus, according to formulas (3), the parametric form of the coefficients of the equation (E) are given by

$$\begin{array}{rclcrcll}\underline{n_{\boldsymbol{d}}}(r) &=& n_{\boldsymbol{d}}-\alpha_{\boldsymbol{d}}\,u &,& \overline{n_{\boldsymbol{d}}}(r) &=& n_{\boldsymbol{d}}+\beta_{\boldsymbol{d}}\,v & \text{for } \boldsymbol{d}\in\mathrm{Expon(E)}\\ \underline{m}(r) &=& m-\alpha\,u &,& \overline{m}(r) &=& m+\beta\,v &.\end{array} \tag{10}$$

Then the crisp form $\mathscr{C}(\mathrm{E})$ of (E) given in (8) is written as a system of two equations with $k+2$ variables $x_1,\dots,x_k,u,v$, where u and v are dependent on each other:

$$\mathscr{C}(\mathrm{E}):\begin{cases}\sum\limits_{\boldsymbol{d}} n_{\boldsymbol{d}}\,\boldsymbol{x}^{\boldsymbol{d}}-\alpha_{\boldsymbol{d}}\,u\,\boldsymbol{x}^{\boldsymbol{d}}=m-\alpha\,u\\ \sum\limits_{\boldsymbol{d}} n_{\boldsymbol{d}}\,\boldsymbol{x}^{\boldsymbol{d}}+\beta_{\boldsymbol{d}}\,v\,\boldsymbol{x}^{\boldsymbol{d}}=m+\beta\,v\end{cases}$$

By gathering all the terms to the left of the two equalities, we find the crisp form expressed in the form (9) of the theorem. Last assertion in the theorem about $\mathrm{Sol}^+(\mathrm{E})$ follows directly from Identity (8). □

What is noteworthy is that Theorem 3.3 only requires that the restrictions on $[0,1]$ of the two spread functions L and R are bijective.

Our approach allows at the same time to improve and to generalize the methods known so far. For instance, results in [18] and [20] are restricted to triangular fuzzy numbers. Indeed, the crisp form of (E) with two parameters $u=\mathrm{L}^{-1}(r)$ and $v=\mathrm{R}^{-1}(r)$ given in Identity (9) is a generalization of the crisp form known in triangular case with only one parameter r where $r\in[0,1]$.

In the aforementioned articles, for each problem to be solved, the algorithm computes the system $\mathscr{C}(\mathrm{E})$ in variables $x_1,\dots,x_k,r$, which is linear w.r.t. r. Then it is rewritten into an equivalent system of four algebraic equations in $x_1,\dots,x_k$ with real coefficients called *collected crisp form* of (E). In next section 3.4, we will show how to get a particular collected crisp form reduced to three equations instead of four, for any family $\mathfrak{F}(\mathrm{L},\mathrm{R})$ with bijective spread functions L and R. This is the real transform of (E). In addition, we explicitly give its formulation from (E).

3.4. **The real transform and the positive real solutions of** (E)**.** We define here the real transform of a fuzzy equation (E) and show that its positive real solutions are also those of (E).

Definition 3.3. Let L and R be two spread functions and

$$(\mathrm{E}):\sum_{\boldsymbol{d}\in\mathrm{Expon(E)}}\widetilde{n_{\boldsymbol{d}}}\,\boldsymbol{x}^{\boldsymbol{d}}=\widetilde{m}$$

a fuzzy equation with coefficients in the family $\mathfrak{F}(\mathrm{L},\mathrm{R})$ given by their representations in tuple as follows: $\widetilde{n_{\boldsymbol{d}}}=(n_{\boldsymbol{d}},\alpha_{\boldsymbol{d}},\beta_{\boldsymbol{d}})$ ($\boldsymbol{d}\in\mathrm{Expon(E)}$) and $\widetilde{m}=(m,\alpha,\beta)$. The *real transform* $\mathscr{T}(\mathrm{E})$ of (E) is the following polynomial system over $\mathbb{R}$:

$$\mathscr{T}(\mathrm{E}):\begin{cases}\sum\limits_{\boldsymbol{d}\in\mathrm{Expon(E)}} n_{\boldsymbol{d}}\,\boldsymbol{x}^{\boldsymbol{d}} &=& m\\ \sum\limits_{\boldsymbol{d}\in\mathrm{Expon(E)}} \alpha_{\boldsymbol{d}}\,\boldsymbol{x}^{\boldsymbol{d}} &=& \alpha\\ \sum\limits_{\boldsymbol{d}\in\mathrm{Expon(E)}} \beta_{\boldsymbol{d}}\,\boldsymbol{x}^{\boldsymbol{d}} &=& \beta\,.\end{cases}\tag{11}$$

This definition naturally extends to a system (S) of fuzzy equations such as (E). We denote by $\mathscr{T}(\mathrm{S})$ its real transform, i.e. the system formed by the real transforms of the equations in (S).

Theorem 3.4. *If the two spread functions* L *and* R *are bijective then the set of positive real solutions of* (E) *equals the one of its real transform* $\mathscr{T}(\mathrm{E})$*; in other words:*

$$\mathrm{Sol}^+(\mathrm{E})=\mathrm{Sol}^+(\mathscr{T}(\mathrm{E}))\quad.$$

Proof. Let $\boldsymbol{a}\in\mathbb{R}^{+k}$. As the spread functions L and R are bijective, we can apply Theorem 3.3. According to this theorem, we know that $\boldsymbol{a}\in\mathrm{Sol}^+(\mathrm{E})$ if and only if, for all $r\in[0,1]$, the crisp form of (E) expressed in (9) is satisfied by the $(k+2)$-uplet $(a_1,\dots,a_k,\mathrm{L}^{-1}(r),\mathrm{R}^{-1}(r))$.

With $r=1$, we have $u=\mathrm{L}^{-1}(1)=0$. Then $\boldsymbol{a}\in\mathrm{Sol}^+(\mathrm{E})$ satisfies the equation $\sum_d\ n_{\boldsymbol{d}}\boldsymbol{x}^{\boldsymbol{d}}=m$. Note that when $r=1$ we have $v=0$ too because $\mathrm{R}(0)=1$, and we find the same equation and not a second one. This is why we obtain three equations instead of four. Then, by taking $r=0$ we have $u=\mathrm{L}^{-1}(0)=1$ and $v=\mathrm{R}^{-1}(0)=1$. By replacing in (9) the expression $\sum_d\ n_{\boldsymbol{d}}\boldsymbol{x}^{\boldsymbol{d}}-m$ by 0 and each variable u and v by 1, we deduce that a positive solution of (E) is also a positive solution of the real transform $\mathscr{T}(\mathrm{E})$ of (E).

For the inverse inclusion, consider the crisp form $\mathscr{C}(\mathrm{E})$ as a polynomial system in the variables $\boldsymbol{x}$ and with coefficients in the ring $\mathbb{R}[u,v]$. Any solution $(a_1,\dots,a_k)\in\mathbb{R}^k$ of $\mathscr{T}(\mathrm{E})$ is also a solution of $\mathscr{C}(\mathrm{E})$ in $\mathbb{R}^k$ whatever the parameters u and v may be in the interval $[0,1]$. Obviously it remains true when they are furthermore connected by the constraint

$L^{-1}(u) = R^{-1}(v) \in [0,1]$. Hence any positive real solution of the real transform $\mathscr{T}(E)$ is also a positive real solution of the fuzzy equation (E). □

Theorem 3.4 ensures that finding the positive real roots of (E) amounts to finding the positive real roots of its real transform $\mathscr{T}(E)$. Therefore it is no use to develop intermediate computations on parametric representation like the previous methods did in the specific triangular case.

From Remark 3.2, we immediately deduce the expression of $\mathscr{T}(E(I))$:

Corollary 3.5. *Let* $I \in \{-1,1\}^k$. *The real transform of the induced equation* $E(I)$ *is given by:*

$$\mathscr{T}(E(I)): \begin{cases} \sum\limits_{d|I^d>0} n_d\, x^d - \sum\limits_{d|I^d<0} n_d\, x^d &= m \\ \sum\limits_{d|I^d>0} \alpha_d\, x^d + \sum\limits_{d|I^d<0} \beta_d\, x^d &= \alpha \\ \sum\limits_{d|I^d>0} \beta_d\, x^d + \sum\limits_{d|I^d<0} \alpha_d\, x^d &= \beta \quad . \end{cases} \tag{12}$$

If, in addition, the spread functions L *and* R *are both bijective then* $\mathrm{Sol}^+(E(I)) = \mathrm{Sol}^+(\mathscr{T}(E(I)))$.

3.5. **Comparison with previous methods in the triangular case.** Consider a system (S) formed by s polynomial equations with fuzzy coefficients.

In the specific case of triangular fuzzy numbers as coefficients, the authors of [18] and [20] compute a collected crisp form of (S) formed by $4s$ real algebraic equations. In this part we are interested in the relationship between their collected crisp form with $4s$ equations and our collected crisp system with $3s$ equations, that is the real transform of (S). The respective sets of positive real solutions of these two systems equal Sol(S); it is the principle of any collected crisp form of (S).

Consider below the system F_1 of Section 6 in [20]:

$$F_1: \begin{cases} (2,1,1)xy + (3,1,1)x^2y^2 + (2,1,1)x^3y^3 = (7,3,3) \\ (5,1,1)xy + (2,3,1)x^2y^2 + (2,2,1)x^3y^3 = (9,6,3)\ . \end{cases}$$

Applied to first equation, the algorithm proposed in [20] produces the following collected crisp form:

$$\begin{cases} xy + x^3y^3 - 3 + x^2y^2 = 0, \\ xy + 2x^2y^2 - 4 + x^3y^3 = 0 \\ -xy - x^3y^3 + 3 - x^2y^2 = 0 \\ 3xy + 4x^2y^2 - 10 + 3x^3y^3 = 0 \quad ; \end{cases}$$

and it produces the following collected crisp form of the second equation of F_1:

$$\begin{cases} xy + 3x^2y^2 + 2x^3y^3 - 6 = 0, \\ 4xy - x^2y^2 - 3 = 0, \\ -xy - x^3y^3 + 3 - x^2y^2 = 0, \\ 6xy + 3x^2y^2 - 12 + 3x^3y^3 = 0 \quad . \end{cases}$$

Call T_1 the system formed of the eight preceding equations.

Furthermore, by applying to F_1 our formula (11) defining the real transform, we get $\mathscr{T}(F_1)$, the following system of six equations:

$$\mathscr{T}(F_1): \begin{cases} 2xy + 3x^2y^2 + 2x^3y^3 = 7 \\ xy + x^2y^2 + x^3y^3 = 3 \\ xy + x^2y^2 + x^3y^3 = 3 \\ 5xy + 2x^2y^2 + 2x^3y^3 = 9 \\ xy + 3x^2y^2 + 2x^3y^3 = 6 \\ xy + x^2y^2 + x^3y^3 = 3 \quad . \end{cases}$$

An easy computation shows the equivalence of the systems T_1 and $\mathscr{T}(F_1)$, whose set of solutions is $\{(x,y) \in \mathbb{R} \mid xy = 1\}$. This phenomenon of equivalence between both approaches may be explained in a very general way as we show

below by considering the classical computation of the collected crisp form obtained by an application of the algorithm of [20] on the generic equation (5) of (E).

Let $\widetilde{m} = (n, \alpha, \beta)$ and $\widetilde{n_{\boldsymbol{d}}} = (n_{\boldsymbol{d}}, \alpha_{\boldsymbol{d}}, \beta_{\boldsymbol{d}})$ ($\boldsymbol{d} \in \mathrm{Expon(E)}$) be the respective tuple representations of the fuzzy coefficients of (E) that are assumed to be triangular. According to formulas (4), in the triangular case the r-cuts are given by

$$\begin{array}{rcl} \widetilde{n_{\boldsymbol{d}}}(r) & = & [\alpha_{\boldsymbol{d}} r + n_{\boldsymbol{d}} - \alpha_{\boldsymbol{d}}\,, -\beta_{\boldsymbol{d}} r + n_{\boldsymbol{d}} + \beta_{\boldsymbol{d}}] \quad \text{for } \boldsymbol{d} \in \mathrm{Expon(E)}, \quad \text{and} \\ \widetilde{m}(r) & = & [\alpha r + m - \alpha\,, -\beta r + m + \beta]\,. \end{array}$$

For a triangular fuzzy number, L = R = F, where $\mathrm{F}(x) = \mathrm{F}^{-1}(x) = 1 - x$, being known, the previous methods replace directly $\mathrm{L}^{-1}(r)$ and $\mathrm{R}^{-1}(r)$ by their expression in the variable r in the equations. That's how they end up in the crisp form of (E) below expressed as two polynomials in the variable r:

$$\mathscr{C}(\mathrm{E}) : \left\{ \begin{array}{l} (\sum\limits_{\boldsymbol{d}} \alpha_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} - \alpha) r + \sum\limits_{\boldsymbol{d}} (n_{\boldsymbol{d}} - \alpha_{\boldsymbol{d}})\, \boldsymbol{x}^{\boldsymbol{d}} - m + \alpha = 0 \\ (\beta - \sum\limits_{\boldsymbol{d}} \beta_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}}) r + \sum\limits_{\boldsymbol{d}} (n_{\boldsymbol{d}} + \beta_{\boldsymbol{d}})\, \boldsymbol{x}^{\boldsymbol{d}} - m - \beta = 0\,. \end{array} \right.$$

A k-uplet $(x_1, \ldots, x_k) \in \mathbb{R}^k$ is a solution of $\mathscr{C}(\mathrm{E})$ for all $r \in [0,1]$ if and only if the constant coefficients and the coefficients w.r.t. the variable r of these independent equations are all zero (because a non-zero polynomial of degree 1 has only one root). The collected crisp form of (E) is therefore written

$$\left\{ \begin{array}{lcl} \sum\limits_{\boldsymbol{d}} \alpha_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} & = & \alpha \\ \sum\limits_{\boldsymbol{d}} (n_{\boldsymbol{d}} - \alpha_{\boldsymbol{d}})\, \boldsymbol{x}^{\boldsymbol{d}} & = & m - \alpha \\ \sum\limits_{\boldsymbol{d}} \beta_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} & = & \beta \\ \sum\limits_{\boldsymbol{d}} (n_{\boldsymbol{d}} + \beta_{\boldsymbol{d}})\, \boldsymbol{x}^{\boldsymbol{d}} & = & m + \beta\,. \end{array} \right.$$

By applying this transformation to each equation in the system F_1, we find the collected crisp form T_1 of our example. For the generic equation (E), by injecting the first equation into the second one and by noting that the last equation is the sum of the three other ones, we obtain the real transform $\mathscr{T}(\mathrm{E})$ with three equations defined in (11).

3.6. **Case of trapezoidal fuzzy numbers.** In this section, we extend the definition of fuzzy numbers to *trapezoidal* fuzzy numbers by allowing $\mu_{\widetilde{n}}^{-1}(\{1\})$ to an interval $[a, b]$ and no longer the unique value n. As mentioned in Section 2.1 our results adapt to trapezoidal fuzzy numbers with bounded support.

In this context, a fuzzy number $\widetilde{n}$ with bounded support is of type L-R if its membership function $\mu_{\widetilde{n}}$ has the following form:

$$\mu_{\widetilde{n}}(x) = \left\{ \begin{array}{lll} \mathrm{L}\left(\frac{a-x}{\alpha}\right) & \text{for} \quad a - \alpha \le x < a \quad \text{when} \quad \alpha \neq 0 \\ 1 & \text{for} \quad x \in [a, b] & \\ \mathrm{R}\left(\frac{x-b}{\beta}\right) & \text{for} \quad b < x \le b + \beta \quad \text{when} \quad \beta \neq 0 \\ 0 & \text{for} \quad x \in (-\infty, a - \alpha) \cup (b + \beta, +\infty) & \end{array} \right. .$$

Then the tuple representation of the fuzzy number $\widetilde{n}$ is the quadruplet (a, b, α, β).

The expression of the parametric representation given in Proposition 2.2 takes the following form for a trapezoidal number of type L − R whose spread functions L and R are bijective:

$$\left\{ \begin{array}{l} \underline{n}(r) = a - \alpha\, u \\ \overline{n}(r) = b + \beta\, v \end{array} \right. \qquad \text{where } u = \mathrm{L}^{-1}(r) \text{ and } v = \mathrm{R}^{-1}(r) \text{ for } r \in [0,1] \quad . \tag{13}$$

Firstly, we seek for the positive real solutions of an equation (E) : $\sum_{\boldsymbol{d} \in \mathrm{Expon(E)}} \widetilde{n_{\boldsymbol{d}}}\, \boldsymbol{x}^{\boldsymbol{d}} = \widetilde{m}$ with trapezoidal fuzzy coefficients of type L-R, where $\widetilde{n_{\boldsymbol{d}}} = (a_{\boldsymbol{d}}, b_{\boldsymbol{d}}, \alpha_{\boldsymbol{d}}, \beta_{\boldsymbol{d}})$ and $\widetilde{m} = (a, b, \alpha, \beta)$ are their tuple representations. For this purpose,

we compute its real transform $\mathscr{T}(\mathrm{E})$ without resorting to the assumption of symmetry on the coefficients. Applying the argument of Section 3.4 (here $\mathrm{L}(1)=\mathrm{R}(1)=0$ and $\mathrm{L}(0)=\mathrm{R}(0)=1$), we obtain in the same way:

$$\mathscr{T}(\mathrm{E}):\begin{cases}\sum\limits_{\boldsymbol{d}} a_{\boldsymbol{d}}\,\boldsymbol{x}^{\boldsymbol{d}} &= a\\ \sum\limits_{\boldsymbol{d}} b_{\boldsymbol{d}}\,\boldsymbol{x}^{\boldsymbol{d}} &= b\\ \sum\limits_{\boldsymbol{d}} \alpha_{\boldsymbol{d}}\,\boldsymbol{x}^{\boldsymbol{d}} &= \alpha\\ \sum\limits_{\boldsymbol{d}} \beta_{\boldsymbol{d}}\,\boldsymbol{x}^{\boldsymbol{d}} &= \beta\,.\end{cases} \tag{14}$$

Secondly, the use of the real transform for solving polynomial fuzzy systems will directly transposes to systems with symmetrical trapezoidal fuzzy coefficients.

The algorithms proposed in this article will remain valid for symmetrical trapezoidal fuzzy numbers. Only the algorithmic function **RealTransform**(S), which returns the real transform of a fuzzy (S) system of s equations, will have to be adapted in order to return the real transform $\mathscr{T}(\mathrm{S})$ with $4\,s$ instead of $3\,s$ equations by slightly applying formula (14) to each equation of (S).

4. Real solving of fuzzy polynomial systems

The resolution of a fuzzy system (S) of s equations with coefficients in $\mathfrak{F}(\mathrm{L},\mathrm{L})$ follows directly from the results for a single equation. In Section 3, from equation (E) we deduce 2^k fuzzy induced equations E(I) where I runs throughout the 2^k k-uplets of $\{-1,1\}^k$. By applying Theorem 3.2 the solutions of (E) are deduced from the positive solutions of every fuzzy induced equation E(I). Corollary 3.5 ensures that the positive solutions of each E(I) are the positive solutions of its real transform $\mathscr{T}(\mathrm{E(I)})$. In this section, for the system S we will consider in the same way 2^k real transforms $\mathscr{T}(\mathrm{S(I)})$ of induced systems S(I) with $3\,s$ equations, where I runs throughout the 2^k k-uplets of $\{-1,1\}^k$.

This section establishes the main Theorem 4.1 that expresses the real solutions of a system from the positive solutions of the 2^k real transforms of these induced systems. A first algorithm results from a direct application of this theorem. Then it is discussed how to reduce the number of computational branches by deriving some of the 2^k sets of real solutions of (S) from the positive real solutions of some system $\mathscr{T}(\mathrm{S(I)})$ previously processed by the algorithm. These elements will lead to an optimized algorithm `SolveFuzzySystem` presented in next section.

4.1. Foundations.

Let $(\mathrm{E}_1),\dots,(\mathrm{E}_s)$ be the s equations of (S) with symmetrical fuzzy coefficients. For each $\mathrm{I}\in\{-1,1\}^k$, we denote by S(I) the induced fuzzy system formed by the s induced fuzzy equations $\mathrm{E}_1(\mathrm{I}),\dots,\mathrm{E}_s(\mathrm{I})$ according to the notation of Section 3.2. For each $\mathrm{I}\in\{-1,1\}^k$ the real transform $\mathscr{T}(\mathrm{S(I)})$ of the induced system S(I) is the system formed by the $3\,s$ equations coming from the real transforms of the induced equations $\mathrm{E}_1(\mathrm{I}),\dots,\mathrm{E}_s(\mathrm{I})$.

The following main theorem is a direct consequence of Theorem 3.2 and Corollary 3.5:

Theorem 4.1. *Let* (S) *be a fuzzy system with coefficients in the same family* $\mathfrak{F}(\mathrm{L},\mathrm{L})$. *If the spread function* L *is bijective then the set of solutions of* (S) *is the union of the* $\mathrm{I}\times\boldsymbol{b}$ *for all* k*-uplets* $\boldsymbol{b}$ *which are positive real solutions of the real transform* $\mathscr{T}(\mathrm{S(I)})$ *where* I *runs throughout* $\{-1,1\}^k$. *In other words,*

$$\mathrm{Sol(S)}=\bigcup_{\mathrm{I}\in\{-1,1\}^k}\{\,\mathrm{I}\times\boldsymbol{b}\;\mid\;\boldsymbol{b}\in\mathrm{Sol}^+(\mathscr{T}(\mathrm{S(I)}))\,\}=\bigcup_{\mathrm{I}\in\{-1,1\}^k}\mathrm{I}\otimes\mathrm{Sol}^+(\mathscr{T}(\mathrm{S(I)})).$$

A first algorithm for the real solving of fuzzy systems, called `BA-SFS`, derives naturally from Theorem 4.1 and the preceding results. It is given below. Its input is a polynomial system with fuzzy coefficients given in tuple representation and supposed to belong to a same family $\mathfrak{F}(\mathrm{L},\mathrm{L})$ whose spread function L is bijective. It is based on the following functions:

- **Multisign**(j) is the natural bijection between $\{0,\dots,2^k-1\}$ and $\{-1,1\}^k$: the image of the integer $j=\sum_{i=0}^{2^k-1}b_i2^i$ is the k-uplet $\mathrm{I}=(c_o,\dots,c_{2^k-1})\in\{-1,1\}^k$ with $c_i=2b_i-1$;
- **SolPos**(SR) returns the positive real solutions of a system SR of polynomial equations with coefficients in $\mathbb{R}$. It can be based on any known computer algebra method to solve such a system. For example, in the implementation described in section 6 it is Wu's method of decomposition into triangular polynomial systems that is used [31];

- **RealTransform**(S, I) applies formula (12) to return the $3s$ equations of the real transform of the fuzzy system S(I) of s equations (it is $4s$ equations when the fuzzy numbers are trapezoidal).

Algorithm 1 `BA-SFS`, a basic algorithm for solving fuzzy systems

Require: S, a polynomial system with fuzzy coefficients in $\mathfrak{F}(L,L)$ under tuple representation
k, the dimension, i.e. the number of variables
Ensure: sol, the set of real solutions of (S)

$sol := \{\}$
for $j := 0$ **to** $2^k - 1$ **do**
  I := **MultiSign**(j)
  TRSI := **RealTransform**(S, I)
  PR := **SolPos**(TRSI)
  $sol := sol \bigcup$ I $\otimes$ PR
end for

return sol

4.2. **Reduction of the number of algebraic systems to solve.** By extension of the notation used for one equation, we denote Expon(S) the set of $\boldsymbol{d}$ in $\mathbb{N}^k$ such that $\boldsymbol{x}^{\boldsymbol{d}}$ appears in the left hand side of some equation of (S) with a nonzero coefficient (the constant in right side can be possibly $0 = (0,0,0)$).

In the for loop of Algorithm `BA-SFS`, the system S(I) may be identical to one considered in some previous step. In this case, a new calculation of PR, the positive real solutions of S(I), is redundant and it is desirable to avoid it.

The optimized algorithm called `SolveFuzzySystem`, or `SFS`, proposed in Section 5 provides every real solution of (S) avoiding unnecessary calculations.

Thus we first seek to identify, from (S), induced systems S(I) which are identical. Example 3.1 dealing with a single equation shows how this situation can occur from the eight fuzzy equations that reduces to only two distinct ones. When, for example, all the components of each k-uplet $\boldsymbol{d}$ of Expon(S) are even, the 2^k induced systems S(I) are identical. Our goal is to automize the recognition of identical induced systems.

For this, let us construct the matrix M(S) of signs where the columns are indexed by the 2^k k -uplets I of $\{-1,1\}^k$ and the lines by the k -uplets $\boldsymbol{d}$ of Expon(S). The element at line $\boldsymbol{d}$ and column I is $\mathrm{I}^{\boldsymbol{d}} \in \{-1,1\}$ which replaces the sign of $\boldsymbol{x}^{\boldsymbol{d}}$ in S(I):

$$\mathrm{M(S)} = \begin{array}{c} \\ \boldsymbol{d} \end{array} \begin{array}{c} \begin{array}{c} \mathrm{I} \end{array} \\ \begin{pmatrix} \vdots \\ \cdots \quad \mathrm{I}^{\boldsymbol{d}} \\ \end{pmatrix} \end{array}$$

Let $\mathrm{I}_1, \mathrm{I}_2 \in \{-1,1\}^k$ two distinct k -uplets and consider $\mathrm{C(I_1)}$ and $\mathrm{C(I_2)}$ the two columns of M(S) respectively indexed by $\mathrm{I_1}$ and $\mathrm{I_2}$. If $\mathrm{C(I_1)} = \mathrm{C(I_2)}$ then $\mathrm{S(I_1)} = \mathrm{S(I_2)}$. Indeed, according to the definition of the induced equations given in (6), for every equation (E) of the system (S), the coefficient $\mathrm{I_1^{\boldsymbol{d}}}\,\widetilde{n_{\boldsymbol{d}}}$ of each monomial $\boldsymbol{x}^{\boldsymbol{d}}$ appearing in the left hand side of each equation $\mathrm{E(I_1)}$ is identical to the coefficient $\mathrm{I_2^{\boldsymbol{d}}}\,\widetilde{n_{\boldsymbol{d}}}$ in $\mathrm{E(I_2)}$.

The number of distinct systems S(I) among the 2^k induced by (S) is thus at most equal to the number of distinct columns in M(S). The matrix of signs avoids solving many induced systems. Other rapid detections of identical induced systems can be implemented. Indeed, two systems $\mathrm{S(I_1)}$ and $\mathrm{S(I_2)}$ can be identical up to a permutation of the equations whereas $\mathrm{C(I_1)} \neq \mathrm{C(I_2)}$. We can also find equations of the form "$a = 0$" and "$-a = 0$" since they have the same solutions.

Remark 4.1. If two systems are such that $\mathrm{S(I_1)} = \mathrm{S(I_2)}$, only one system will be solved, for example $\mathrm{Sol} = \mathrm{Sol}^+(\mathrm{S(I_1)})$, but they will produce two sets of solutions: $\mathrm{I_1} \otimes \mathrm{Sol}$ and $\mathrm{I_2} \otimes \mathrm{Sol}$.

5. ALGORITHM SolveFuzzySystem

This section proposes an optimized algorithm SolveFuzzySystem, or SFS in contracted form, for computing the real solutions of a polynomial fuzzy system. The symmetrical coefficients of the system are fuzzy numbers given under tuple representation and are supposed to belong to a same family $\mathfrak{F}(L,L)$ where the spread function L is bijective. After the description of sequential algorithm SFS in Section 5.1, we discuss its parallelization in Section 5.2.

5.1. **The sequential algorithm** SolveFuzzySystem**.** Like in the first algorithm BA-SFS, algorithm SFS consists in iteratively going through the columns of the matrix of signs M(S) described in Section 4.2. For each column C(I) indexed by a k-uplet I of signs, it looks for real solutions of the system (S) from S(I) avoiding unnecessary computations if one of the previous columns allows it.

For this purpose, it uses the following three vectors, indexed from 0 to 2^k-1, which are empty at the beginning of the algorithm:

- $DistinctColumns$ will contain the distinct columns of the matrix of signs;
- $DistinctSystems$ will contain the distinct fuzzy systems whose positive solutions have been calculated; Its indexes are related to the indexes of $DistinctColumns$;
- lb will contain positive real solutions of S(I) such that C(I) is in $DistinctColumns$; so it will be naturally indexed by following $DistinctColumns$.

For a current column I, as explained in Section 4.2, redundant calculations may be avoided in the following cases:

Case 1 If the column C(I) is identical to a previous column C(J) in M(S) then S(I) = S(J). In this case, the algorithm does not add the column C(I) in the vector $DistinctColumns$. It uses the positive real solutions of S(J) already been calculated and stored in lb during a previous step (see Remark 4.1). This is the first step of the algorithm (see the **4:** statement).

Case 2 If S(I) is identical to a system S(J) where C(J) is a column preceding C(I) but being distinct from it, then the algorithm goes to the following index. It adds the column C(I) to the vector $DistinctColumns$ and the positive solutions of S(J) already calculated in lb. As it is not relevant to add S(I) to vector $DistinctSystems$, the algorithm does not add to it at the corresponding index cpt and its place remains empty. We can then apply tests (1) and (2) to the following columns of M(S) (see the **7:** statement).

Outside of situations 1. and 2., the positive solutions of the fuzzy system S(I) are computed, as in the basic algorithm BA-SFS, with the functions **RealTransform** and **SolPos**. At index cpt in the vectors $DistinctColumns$, $DistinctSystems$ and lb, the algorithm stores respectively C(I), S(I) (in case 2., the place remains empty) and the positive solutions of S(I). It will be possible to apply tests (1) and (2) to the following columns of M(S) (see the **9:** statements).

In each step of the for loop, in other words, for each k-uplet I of signs Theorem 4.1 is applied to obtain the real solutions of (S) associated with the positive real solutions of the induced system S(I).

Functions used by the algorithm SolveFuzzySystem are those of the basic algorithm BA-SFS completed by the following ones:

- the function **SignColumn**(j,S) returns the $(j+1)$-th column of the matrix M(S) of signs (j starts to 0 not 1);
- the function **IsIn**(e , $Distinct$) returns -1 if e is not in the vector $Distinct$, otherwise it returns the index of the first occurrence of e in $Distinct$. This function is called indifferently on the signs columns of M(S) and on the polynomial systems. For an efficient search on a polynomial system, we order polynomials according to a total order on the monomials and by assigning the sign + to the dominant monomial (the monomials in a same polynomial are ordered and the equations in the system are also ordered).

5.2. **A parallel version of the algorithm** SolveFuzzySystem**,** The positive solutions of each induced system S(I) are computed independently of those of the other fuzzy induced systems (with the functions **RealTransform** and **SolPos** successively). This part being the most expensive of the algorithm SFS, its possible parallelization is welcome. It requires to modify the algorithm in order to identify the distinct induced systems S(I) to be solved, but without performing the resolution, and simultaneously store the useful informations for further application of Theorem 4.1.

Algorithm 2 `SolveFuzzySystem` or `SFS`, an optimized algorithm for solving fuzzy systems

Require: S, a polynomial system with fuzzy coefficients in $\mathfrak{F}(L,L)$ under tuple representation
k, the dimension, i.e. the number of variables

Data: $cpt := -1$, the counter of the number of distinct columns of M(S)
$DistinctColumns := []$, the distinct columns of M(S) already scanned
$DistinctSystems := []$, the distinct systems S(I) already met
$lb := []$, $lb[i]$ will be the set $\mathrm{Sol}^+(S(I))$ when C(I) will be $DistinctColumns[i]$
$sol := \{\}$

Ensure: sol , the set of real solutions of (S)

 # the index j runs throughout the columns of M(S) which are indexed by MultiSign(j)
1: for $j := 0$ **to** $2^k - 1$ **do**
2: I := **MultiSign**(j)
C := **SignColumn**(j) **# C is the current column in M(S)**
 # test if C is equal to a previous column of M(S), for Case 1
i := **IsIn**(C , $DistinctColumns$)
3: if $i \neq -1$ **then**
4: **# $lb[i]$ contains the positive real solutions of S(I): apply Theorem4.1 to $lb[i]$ and I**
$sol := sol \bigcup I \otimes lb[i]$
 # move to the next column in M(S) and cpt is not incremented
go to 1:
5: end if
 # here C is a new column
$cpt := cpt + 1$
$DistinctColumns[cpt] :=$ C
 # test if S(I) is a new system, for Case 2
i := **IsIn**(S(I) , $DistinctSystems$)
6: if $i \neq -1$ **then**
7: **# the i-th system equals S(I). Its positive real solutions in $lb[i]$ are copied in $lb[cpt]$ because C is a "new" column**
$lb[cpt] := lb[i]$
8: else
9: **# store the new system at cpt index, calculate its positive real solutions and store them in $lb[cpt]$**
$DistinctSystems[cpt] :=$ S(I)
TRSI := **RealTransform**(S(I))
$lb[cpt]$:= **SolPos**(TRSI)
10: end if
 # Apply Theorem 4.1
$sol := sol \bigcup I \otimes lb[cpt]$
11: end for
12: return sol

To achieve this aim, we modify the role of vector *lb*. It will be indexed from 0 to $2^k - 1$, like the columns of M(S), and $lb[j]$ will contain the index in $DistinctSystems$ of the system corresponding to column **SignColumn**(j). When the resolution of a system would have been terminated, its positive solutions will be stored in a new vector $SPos$, with same indexes as $DistinctSystems$.

The general parallel algorithm takes place in the following three steps:

(1) Detect efficiently the distinct systems to solve in parallel. Moreover, for each column I of M(S) (and equivalently for each j from 0 to 2^k-1), the index of the first previous system equals to the real transform of S(I) is affected to $lb[j]$ (it can be j itself if S(I) is new). The distinct systems are stored in the vector D$istinct$S$ystems$. This is realized within the framework of the sequential algorithm SFS modified in the following way:
   - in **4:** replace the statement $sol := sol \bigcup \mathrm{I} \otimes lb[i]$ by $lb[j] := lb[i]$,
   - in **7:** replace the statement $lb[cpt] := lb[i]$ by $lb[j] := lb[i]$,
   - in **9:** replace both statements TRSI := **RealTransform**(S(I)) and $lb[cpt]$:= **SPos** (TRSI) by $lb[j] := cpt$.

(2) In parallel, solves every distinct system of the vector D$istinct$S$ystems$. For each system SI := D$istinct$S$ystems[cpt]$, if it exists (see Remark 5.1 below), we apply the following statements:
   TRSI := **RealTransform**(SI)
   SP$os[cpt]$:= **SolPos**(TRSI)

(3) Cross over the vector lb to build all the solutions of the system (S) from the results of previous step ; It consists mainly in applying Theorem 4.1. This last step can also be performed in parallel. Each $lb[j]$ owns the value of the cpt index of positive real solutions of S(I) in the vector SPos where I = **MultiSign**(j). This step is realized by the following loop:
   for $j := 0$ **to** 2^k-1 **do**
   $sol := sol \bigcup$ **MultiSign**$(j) \otimes$ SP$os[lb[j]]$
   end for

Remark 5.1. As in algorithm SFS, when C = C(I) is a new column but S(I) is not a new system, nothing is stored at the corresponding index in D$istinct$S$ystems$.

A detailed example of this parallel version is given in Section 6.2.

6. Implementation of the algorithm SFS and Examples

We used the package Fuzzy under SageMath written by J. Marrez [32]. It contains a function `resolution_reelle_systeme` that implements algorithm SFS of Section 5. Recall that fuzzy numbers and must lies in a family $\mathfrak{F}(\mathrm{L},\mathrm{L})$ where L is bijective.

Section 6.1 describes the function `resolution_reelle_systemes_flous`. Then two complete examples are given in Section 6.2. The first one details intermediate computations of the function `resolution_reelle_systemes_flous`. The second one illustrates on another fuzzy system the computations performed by the parallel version of the algorithm. These examples initially proposed in [16] have been treated by several other authors.

6.1. Representations and main functions implemented in `Fuzzy`.

The external representation of data in Fuzzy package is described below:

- a spread function H has a representation rep(H); for example `"Quad"` if H is quadratic;
- a fuzzy number $\widetilde{n} = (n, a, b)$ with spread functions L and R is represented by:
  rep$(\widetilde{n})$ = `NombreFlouRed((`n, a, b, rep(L), rep(R)`)`;
- a term $x^{\boldsymbol{d}}$ is classically represented by rep$(x^{\boldsymbol{d}})$ = $x_1 ** d_1 \cdots x_k ** d_k$;
- a monomial $m = \widetilde{n}\, x^{\boldsymbol{d}}$ is represented by the pair rep(m) = (rep$(\widetilde{n})$, rep$(x^{\boldsymbol{d}})$);
- a polynomial p with fuzzy coefficients is represented by rep(p), the list of the rep(m) where m is a monomial of p ; i.e. it is a sparse representation;
- a polynomial equation (E) : $p = q$ is represented by the pair rep((E))=(rep(p), rep(q)) ;
- a system (S) formed by s equations $(\mathrm{E}_1), \ldots, (\mathrm{E}_s)$ is represented by the list
  rep((S))=[rep((E_1)), ... , rep((E_2))].

For example, with quadratic fuzzy numbers, the system

$$\mathrm{F}: \begin{cases} x + (-1,1,1) = (-2,1,1)\, y^2, \\ x + (3,1,1) = (2,1,1)\, y^2 \end{cases} \tag{15}$$

is represented by the variable System defined as follows:

```
LeftSide1 = [(NombreFlouRed(1,0,0,"Quad","Quad"),x),
             (NombreFlouRed(-1,1,1,"Quad","Quad"),1)]
RightSide1 = [(NombreFlouRed(-2,1,1,"Quad","Quad"), y**2) ]
LeftSide2 = [(NombreFlouRed(1,0,0,"Quad","Quad"),x),
             (NombreFlouRed(3,1,1,"Quad","Quad"),1)]
RightSide2 = [(NombreFlouRed(2,1,1,"Quad","Quad"),y**2) ]
System = [ (LeftSide1, RightSide1 ), (LeftSide2, RightSide2)]
```

Remark 6.1. The fuzzy equations of system F does not have exactly the same form than the generic equation (E) studied in our paper. The right side of (E) is not restricted to a fuzzy number $\widetilde{m}$. However our results clearly extend to such a form of equations.

The function `resolution_reelle_systemes_flous(S,k)` implements the sequential algorithm SFS of this paper. It takes as first parameter the representation of a fuzzy system (S) with s equations. The second parameter k is the number of variables. It returns the set of k-uplets real solutions of (S). For this, it mainly uses the functions `transformee_reelle()` and `SolPos()` described below.

The function `transformee_reelle(S,rep(I))`, where rep(I) is a list representing a k-uplet of signs $I \in \{-1,1\}^k$, returns the real transform of the induced fuzzy system S(I), that is the system of $3s$ equations with real coefficients obtained by transforming each of its equations E(I) as in Corollary 3.5. It implements our function **RealTransform** of the algorithms in Sections 4 and 5.

The function `SolPos(Sr)` returns the set of positive real solutions of a polynomial system `Sr` with real coefficients. The representation of `Sr` is a list $[p_1,.,\ldots,p_r]$ of polynomials such that $p_i = 0$. It first calls the function `Wu(Sr)` which implements Wu's algorithm [31]. This one returns a set Z of polynomial sets called *characteristics* sets such that the variety V(`Sr`) of zeroes of `Sr` admits a decomposition into triangular polynomial systems of the form $V(\texttt{Sr}) = \bigcup_{B\in Z} V(B) \setminus V(I_B)$ où $I_B = \prod_{b\in B} init(b)$. One can find the definition of the initial $init(b)$ of a polynomial b in ([33]) or in [20], a paper where Wu's method is described in order to solve polynomial systems with triangular fuzzy numbers as coefficients. From the set Z, a function `get_zeros(Z)` returns the elements in $\mathbb{R}^{+k}$ of the variety V(`Sr`).

6.2. **Examples.**

Example 1 : We consider the system F given above in (15). As in Example 6.1 of [20], the call `resolution_reelle_systemes` returns the variety $V(F) = \{(x = -1, y \pm 1)\}$. Here we describe the intermediate computations.

With $k = 2$ variables x, y, there are the $2^2 = 4$ following multisigns: $I_0 = [-1,-1]$, $I_1 = [-1,1]$, $I_2 = [1,-1]$, $I_3 = [1,1]$. By taking them in this order corresponding respectively to $j = 0,1,2,3$ in the algorithm SFS and by ordering the monomials present in F as follows: $x, 1, y^2$, the matrix of signs M(F) is

$$M(F) = \begin{array}{c} \\ x \\ 1 \\ y^2 \end{array} \begin{array}{c} \begin{array}{cccc} I_0 & I_1 & I_2 & I_3 \end{array} \\ \begin{pmatrix} -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

Note that in the performance of the algorithm below, a column of M(F) is represented by a line vector.

The variable `Sr` is the real transform of the induced fuzzy system F(I) of $s = 2$ equations, with $I \in \{I_0,\ldots,I_3\}$; in theory `Sr` is formed by $3s = 6$ equations. But in practice, some are identical. For example, when $I = [-1,-1]$, the real transform of F(I) is formed by the six equations $2y^2 - x - 1 = 0, -y^2 + 1 = 0, -y^2 + 1 = 0, -2y^2 - x + 3 = 0, -y^2 + 1 = 0, -y^2 + 1 = 0$, that reduces to `Sr`$= [-y^2 + 1, 2y^2 - x - 1, -2y^2 - x + 3]$ by removing duplicates polynomials. It is the first real polynomial system to solve in the program. For obvious practical reasons, this system is the one registered in the variable `DistinctSystems`, and not the fuzzy system F(I) as mentioned in the algorithm SFS.

Now let us present the trace of function `resolution_reelle_systemes_flous(F,2)` by linking the effective parameters to formal parameters: `S=F` and `k=2`.
At the beginning, we have: `cpt = -1, DistinctColumns = []` and `DistinctSystems = []`.

```
j=0 :  I=I_{0}=[-1,-1],  C=[-1,1,1]
   As the column C is new
      cpt=cpt+1 ; i.e. cpt =0
      DistinctColumns[0] = C
      Sr = transformee_reelle(S,I) gives Sr=[-y^2+1,2*y^2-x-1,-2*y^2-x+3]
      DistinctSystems[0] = Sr
      SolPos(Sr) gives lb[0]=set([(1, 1)]), the positives real solutions of Sr
      lb[0]= set([(1, 1)])
         the product of I with (1, 1) in lb[0] is added to sol:
      sol =  set([(-1, -1)])

  DistinctColumn=[[1,-1,1]]
  DistinctSystems=[[y^2-1,2*y^2-x-1,2*y^2+x-3]]

j=1 : I=I_{1}=[-1,1], C=[-1,1,1]
    As C equals DistinctColumns[0]
         the product of I with (1, 1) in lb[0] is added to sol:
         sol = set([(-1, 1), (-1, -1)])

j=2 :  I=I_{2}=[1,-1],  C=[1,1,1]
    As the column C is new
      cpt=cpt+1 ; i.e. cpt =1
      DistinctColumns[1] = C
      Sr = transformee_reelle(S,I) gives Sr=[2*y^2+x-1,-2*y^2+x+3,-y^2+1]
      DistinctSystems[1] = Sr
      SolPos(Sr) gives lb[1]= set([]), the positive real solutions of Sr
            any product of elements of lb[1] with I is added in sol:
      sol =  set([(-1, 1), (-1, -1)])

We have
  DistinctColumn=[[-1,1,1],[1,1,1]]
  DistinctSystems=[[y^2-1,2*y^2-x-1,2*y^2+x-3],[2*y^2+x-1,2*y^2-x-3,y^2-1]]

j=3  :  I=I_{3}=[1,1], C=[1,1,1]
   C equals DistinctColumns[1].
       any product of elements of lb[1] with I is added in sol:
   sol =  set([(-1, 1), (-1, -1)])
```

Therefore we find the variety solution $V(F) = \{(x = -1, y \pm 1)\}$.

Example 2 : Detailed example of the parallel version of the algorithm SFS on the following fuzzy system F_1:

$$F_1 : \begin{cases} (2,1,1)\,xy + (3,1,1)\,x^2y^2 + (2,1,1)\,x^3y^3 = (7,3,3), \\ (5,1,1)\,xy + (2,3,1)\,x^2y^2 + (2,2,1)\,x^3y^3 = (9,6,3) \end{cases}$$

The solution returned by the function `resolution_reelle_systemes_flous(F1,2)` of Fuzzy package is the variety $V(F_1) = \{(x = \frac{1}{y}, y) \mid y \in \mathbb{R}\backslash\{0\})\}$, like in [20]. We notice that in the first equation (E), the left spread of each coefficient equals the right spread. Thus, in the real transform of (E), both equations

$$\sum_{\boldsymbol{d}\in \mathrm{Expon(E)}} \alpha_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} = \alpha \quad \text{and} \quad \sum_{\boldsymbol{d}\in \mathrm{Expon(E)}} \beta_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} = \beta$$

are equal to $x^3y^3+x^2y^2+xy=3$. We find again this equation with the right spreads of the second equation of F_1. This is why we will have 4 and not $6=3s$ equations in the real transform of any induced fuzzy equation $F_1(I)$. During an implementation, it is possible to take this into account in order to identify the identical equations once and for all on the initial fuzzy system and then use the result on each of its induced systems.

We will find $V(F_1)$ by applying the parallel version of our algorithm. The multi-signs are the same as for the F system of the first example and the monomials present in F_1 are $\{x^3y^3, x^2y^2, xy, 1\}$. The matrix of signs of F_1 is the following:

$$M(F_1) = \begin{array}{c} \\ x^3y^3 \\ x^2y^2 \\ xy \\ 1 \end{array} \begin{array}{c} \begin{array}{cccc} I_0 & I_1 & I_2 & I_3 \end{array} \\ \begin{pmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

We see that it is sufficient to solve only the systems corresponding to the first two columns; that is, we have the following identities on fuzzy induced systems: $F_1(I_0)=F_1(I_3)$ and $F_1(I_1)=F_1(I_2)$. In the performance of the algorithm, this information ends up in the vector `lb`. Thus, the number of systems to solve is reduced by half.

In the first step of the parallel algorithm, after the loops `j= 0` and `j=1`, we have:

```
DistinctColumns = [[1, 1, 1, 1], [-1, 1, -1, 1]]
DistinctSystems=
    [[2*x^3*y^3+2*x^2*y^2+5*x*y-9,2*x^3*y^3+3*x^2*y^2+2*x*y-7,
                        2*x^3*y^3+3*x^2*y^2+x*y-6, x^3*y^3+x^2*y^2+x*y-3],
     [2*x^3*y^3-3*x^2*y^2+2*x*y+7,2*x^3*y^3-3*x^2*y^2+x*y+6,
                    x^3*y^3-x^2*y^2+x*y+3, 2*x^3*y^3-2*x^2*y^2+5*x*y+9]]
lb = [0,1]
```

After the loops `j=2` and `j=3`, only the vector `lb` is modified as follows: `lb=[0,1,1,0]`.

As `lb[2]=lb[1]` the positive real solutions of the third induced system $F_1(I_2)$ corresponding to `lb[2]` are those of the second induced system $F_1(I_1)$ corresponding to `lb[1]`; after second step, this positive real solutions will be in `SPos[1]`. The same occurs with `lb[3]=lb[0]`.

In the second step, function `SolPos` is called in parallel on each of both systems stored in `DistinctSystems` in order to compute their respective positive real solutions. These solutions are recovered and stored in the vector `SPos`. We then have:

`SPos=[set([(1/y, 'R+')]),set([])]`.

Last step computes in parallel for each `j = 0,1,2,3` the products `I` ⊗ `SPos[lb[j]]` corresponding to the last column "`returned solutions`" below.

j	I	lb[j]	SPos[j]	returned solutions
0	[-1,-1]	0	set([(1/y, 'R+')])	set([(1/y, 'R-')])
1	[-1,1]	1	set([])	set([])
2	[1,-1]	1	SPos[1]	set([])
3	[1,1]	0	SPos[0]	set([(1/y, 'R+')])

The non-empty solutions are added in the variable `sol` containing the real solutions of the fuzzy system F_1:

```
sol =  set([(1/y, 'R+'), (1/y, 'R-')])
```

We find the variety $V(F_1)=\{(x=\frac{1}{y}, y) \mid y\in\mathbb{R}\backslash\{0\})\}$. As in the sequential version, only two distinct systems are solved.

7. Conclusion

Up to now, given a fuzzy system (S) of s equations and k indeterminates, the existing algebraic methods have performed computations with the parametric representation of the coefficients to obtain the collected crisp form of (S) formed by $4s$ real equations. We show that these computations are superfluous and exhibit a formula that defines an equivalent system with $3s$ real equations. We call it the real transform $\mathcal{T}(S)$ of the system (S). As a main property, it has the same positive solutions as (S) (Theorem 3.4).

Unlike the previous methods that were restricted to triangular fuzzy numbers, our results apply to any family $\mathfrak{F}(L,R)$ where the spread functions L and R are bijective. Moreover there is no use to compute the inverse of the spread functions since the real transform is a universal formula independent from L and R.

For solving equations with symmetrical fuzzy coefficients all in a same family $\mathfrak{F}(L,L)$, one must face the issue of the sign of solutions. It is intrinsic to fuzzy numbers, since the product by a real scalar is expressed differently depending on the sign of this scalar. Our strategy has been to only focus on positive solutions by putting back the issue on the fuzzy coefficients. Theorem 4.1 made it possible since it expresses the real solutions of (S) from the positive solutions of at most 2^k real systems. From this theorem we devise a first algorithm that automatizes the research of solutions by avoiding the studies of signs needed in the methods known until then. Our approach is independent of the choice of the method to calculate the positive solutions of a system of polynomial with real coefficients.

Among the 2^k induced systems of (S), some of them are identical. Our examples show that it is not rare to substantially reduce the number of induced systems to solve. We describe a strategy to avoid redundant branches of computations that leads to an optimized algorithm, `SolvingFuzzySystem`, that is implemented in the package `Fuzzy` of the computer algebra system `SageMath`.

The most costly part of this algorithm lies in finding the positive solutions of the real transform of the distinct induced systems of (S). We suggest in Section 5.2 a parallelization of the algorithm `SolvingFuzzySystem` that executes in parallel these independent computations, then we illustrate its performance with the second example of Section 6.

ACKNOWLEGMENT

We warmly thank Didier Dubois and Henri Prade for clarifying some aspects of the fuzzy numbers. We thank Binh-Minh Bui-Xuan for his remarks following his careful reading of the paper. We also thank the reviewers for their valuable comments that allowed us to substantiate our findings.

REFERENCES

[1] J. Aluja, A. Tacu and H. Teodorescu, editors. *Fuzzy Systems in Economy and Engineering*. Publishing House of The Romanian Academy, 1994.

[2] P. Liu. Analysis of approximation of continuous fuzzy functions by multivariate fuzzy polynomials. *Fuzzy Sets and Systems*, 127(3):299–313, 2002.

[3] S. Abbasbandy and M. Amirfakhrian. Numerical approximation of fuzzy functions by fuzzy polynomials. *Applied Mathematics and Computation*, 174(2):1001–1006, 2006.

[4] J.J. Buckley and Y. Qu. Solving linear and quadratic fuzzy equations. *Fuzzy Sets and Systems*, 38(1):43–59, 1990.

[5] J.J. Buckley and E. Eslami. Neural net solutions to fuzzy problems: The quadratic equation. *Fuzzy Sets and Systems*, 86(3):289–298, 1997.

[6] J.J. Buckley, T. Feuring and Y. Hayashi. Solving fuzzy equations using evolutionary algorithms and neural net. *Soft Computing*, 6(2):116–123, 2002.

[7] M. Tavassoli Kajani, B. Asady and A. Hadi Vencheh. An iterative method for solving dual fuzzy nonlinear equation. *Applied Mathematics and Computation*, 167(1):316–323,2005.

[8] M. Friedman, M. Ming and A. Kandel. Fuzzy linear systems. *Fuzzy Sets and Systems*, 96(2): 201–209, 1998.

[9] S Muzzioli and H. Reynaerts. Fuzzy linear systems of the form A1x+b1=A2x+b2. *Fuzzy Sets and Systems*, 157(7): 939–951, 2006.

[10] W.A. Lodwick and D. Dubois. Interval linear systems as a necessary step in fuzzy linear system. *Fuzzy Sets and Systems*, 281:227–251, 2015.

[11] W. Lodwick. Constructing consistent fuzzy surfaces from fuzzy data. *Fuzzy Sets and Systems*, 135(2): 259–277, 2003.

[12] W. Lodwick. *Fuzzy surfaces in GIS and geographical analysis: Theory, analytical methods, algorithms, and applications*. CRC Press, 2007.

[13] S. Abbasbandy and M. Otadi. Numerical solution of fuzzy polynomials by fuzzy neural network. *Applied Mathematics and Computation*, 181(2):1084–1089, 2006.

[14] M. Amirfakhrian. Numerical solution of algebraic fuzzy equations with crisp variable by Gauss–Newton method *Applied Mathematical Modelling*, 32(9):1859–1868, 2008.

[15] H. Rouhparvar. Solving fuzzy polynomial equation by ranking method. First Joint Congress on Fuzzy and Intelligent Systems, Ferdowsi University of Mashhad, Iran, 2007.

[16] S. Abbasbandy, M. Otadi and M. Mosleh. Numerical solution of a system of fuzzy polynomials by fuzzy neural network. *Information Sciences*, 178(8):1948–1960, 2008.

[17] A. Noor'ani, J. Kavikumar, M. Mustafa and S. Nor. Dual fuzzy polynomial equation by ranking method. *Far East J. Math. Sci*, 51(2):151–163, 2011.

[18] A. Abbasi Molai, A. Basiri and S. Rahmany. Resolution of a system of fuzzy polynomial equations using the Gröbner basis. *Information Sciences*, 220:541–558, 2013.

[19] H. Farahani, S. Rahmany, A. Basiri and A. Abbasi Molai. Resolution of a system of fuzzy polynomial equations using eigenvalue method. *Soft Computing*, 19(2):283–291, 2015.

[20] M. Boroujeni, A. Basiri, S. Rahmany and A. Valibouze. Finding solutions of fuzzy polynomial equations systems by an algebraic method. *Journal of Intelligent & Fuzzy Systems*, 30(2):791–800, 2016.

[21] H. Farahani, M. Paripour and S. Abbasbandy. Resolution of single-variable fuzzy polynomial equations and an upper bound on the number of solutions. *Soft Computing*, 23(3): 837–845, 2019.

[22] T. Becker and V. Weispfenning. *Gröbner bases: a computational approach to commutative algebra*. Springer-Verlag New York, 1993.

[23] P. Aubry and M. Moreno Maza. Triangular sets for solving polynomial systems: a comparative implementation of four methods. *Journal of Symbolic Computation*, 28:125–154, 1999.

[24] F. Rouillier. Solving zero dimensional systems through the Rational Univariate Representation. *AAECC*, 9(5):433–461, 1999.

[25] The Sage Developers. *SageMath, the Sage Mathematics Software System*, 2017. http://www.sagemath.org.

[26] D. Dubois, E. Kerre, R. Mesiar and H. Prade. *Fuzzy Interval Analysis. In: Fundamentals of Fuzzy Sets, Dubois,D. Prade,H., Eds: Kluwer, Boston, Mass, The Handbooks of Fuzzy Sets Series*, 483–581. Springer US, Boston, MA, 2000.

[27] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[28] D. Dubois and H. Prade. Operations on fuzzy numbers. *International Journal of systems science*, 9(6):613–626, 1978.

[29] R. Goetschel and W. Voxman. Elementary fuzzy calculus. *Fuzzy Sets Syst.*, 18(1):31–43, January 1986.

[30] L. Stefanini and L. Sorini. Fuzzy arithmetic with parametric LR fuzzy numbers. In *Proceedings of the Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference*, 600–605, Lisbon, Portugal, July 20-24, 2009.

[31] W.T. Wu. A zero structure theorem for polynomial equations solving. *Mathematics-Mechanization Research Preprints*, 2–12, Academia Sinica, China, 1987.

[32] J. Marrez. *Bibliothèque Fuzzy en SageMath, Documentation. Preprint HAL-CNRS*, HAL-ID hal-01663476, 2017. http://hal.upmc.fr/hal-01663476.

[33] S.C. Chou and X.S. Gao. Ritt-Wu's decomposition algorithm and geometry theorem proving. In *International Conference on Automated Deduction*, 207–220. Springer, 1990.