Computing real solutions of polynomial fuzzy systems

Philippe Aubry, Jérémy Marrez, Annick Valibouze

COMPUTING REAL SOLUTIONS OF POLYNOMIAL FUZZY SYSTEMS

PHILIPPE AUBRY, JÉRÉMY MARREZ, AND ANNICK VALIBOUZE

ABSTRACT. This paper presents an efficient algorithm called SolvingFuzzySystem, or SFS, for finding real solutions of polynomial systems whose coefficients are fuzzy numbers with finite support and bijective spread functions. The real solutions of a given fuzzy 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.

1. INTRODUCTION

Modeling problems with uncertain data has important applications in engineering, economics and social sciences [4]. In this context, many researchers have been interested in searching for real solutions of a polynomial equation with fuzzy coefficients. This issue is crucial for the interpolation of fuzzy functions [1]. The methods of resolution were initially based on local techniques [2, 3, 14, 15]. Recently, a global approach using classical algebraic techniques has been developed [7, 13]. Indeed, despite a name that may be confusing, fuzzy numbers benefit from a perfectly formal definition.

We revisit this approach and 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 apply more generally to fuzzy numbers with finite support and bijective spread functions (such as, for example, quadratic fuzzy numbers, that have quadratic spread functions).

In a fuzzy algebraic system, the fuzzy coefficients (coming from the experiments) are generally given under a representation called "tuple". Although the tuple representation is formal, it cannot be handle by usual algebraic methods (Gröbner bases [6], triangular decomposition [5], rational univariate representation [16], ...) to solve the system. Nevertheless, for any fuzzy number with finite support and of bijective spread functions, this tuple representation 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 2.9).

We show how, in the context of fuzzy numbers with finite support and bijective spread functions, the resolution of a system (S) of s equations with k variables and with fuzzy coefficients reduces to computing positive solutions of 2^k systems, each of them formed by $3s$ equations of k variables with real coefficients (Theorem 3.9 and main Theorem 4.1). This new algebraic real system denoted by $\mathscr{T}(S)$ is called the *real transform* of (S). We extend this result

Date: December 18, 2017.

[0] *AMS Subject Classification 2000:* . *Keywords:* Fuzzy numbers, polynomial systems.

to so-called *trapezoidal* fuzzy numbers to get $4\,s$ equations instead of $3\,s$ (Section 3.6). After describing a basic algorithm requiring the resolution of 2^k systems, we propose an optimized algorithm called `SolveFuzzySystem` which reduces the number of systems to solve.

In Section 2, we introduce fuzzy numbers, their different representations, and the transition from one to the other in the case of a fuzzy number with finite support. In Section 3, we define the real transform and the induced fuzzy equations of a given fuzzy polynomial equation. We show that, in the triangular fuzzy case, the real transform is a system equivalent to the collected crisp form calculated by previous methods. Finally, the study is extended to the trapezoidal case. Section 4 applies the results of previous section to a fuzzy polynomial system. It establishes main Theorem 4.1 which leads to a basic algorithm called `BA-SFS`. In Section 5, this latter is optimized in the algorithm `SolveFuzzySystem`, or `SFS`. A discussion about the parallelization of `SFS` follows. The implementation of Algorithm `SFS` in the `Fuzzy` package in `SageMath` [17] is briefly presented in Section 6 together with examples illustrating the sequential algorithm and the parallel algorithm.

2. Fuzzy numbers

This section presents basic generalities on the theory of fuzzy sets and fuzzy numbers, and the two classical representations of fuzzy numbers respectively called “tuple“ and “parametric“. To go further the reader may be interested in [9]. We give some formulas which express the parametric representation of a fuzzy number in function of its tuple representation in case where the number has a finite support and bijective spread functions (Proposition 2.9). These formulas are the key of our algebraic method to solve in $\mathbb{R}$ the algebraic fuzzy systems, where $\mathbb{R}$ is the set of reals.

2.1. **Generalities.**

Fuzzy numbers are particular *fuzzy sets*, that we first define. In classical set theory, a subset E of the universal set X is defined by its Dirichlet's function, called also its characteristic function, defined from X to $\{0,1\}$ such that the image of $x \in \mathrm{X}$ is 1 if $x \in \mathrm{E}$ and 0 otherwise. The concept of fuzzy sets generalizes this classical concept. As depicted in Zadeh's seminal paper [20], a fuzzy set $\widetilde{\mathrm{E}}$ is a class of objects with a continuum of grades of membership. An element x belongs to $\widetilde{\mathrm{E}}$ with a validity degree which is represented by a function with values between 0 and 1.

Definition 2.1. A *fuzzy set* $\widetilde{\mathrm{E}}$ is a pair formed by a set E of X and its *membership function* $\mu_{\widetilde{\mathrm{E}}} : \mathrm{X} \to [0,1]$.
For each element x of X, $\mu_{\widetilde{\mathrm{E}}}(x)$ is called the *grade of membership* of x to $\widetilde{\mathrm{E}}$.

The support $\mathrm{Supp}(f)$ of a function f defined on a set A is the set of $a \in \mathrm{A}$ such that $f(a) \neq 0$. The support of $\widetilde{\mathrm{E}}$ is the support of its membership function: $\mathrm{Supp}(\widetilde{\mathrm{E}}) := \mathrm{Supp}(\mu_{\widetilde{\mathrm{E}}})$. Fuzzy numbers are defined from fuzzy sets by means of the notion of r*-cut*, which is also involved in the parametric representation of Section 2.3.

Definition 2.2. Let $\widetilde{\mathrm{E}}$ be a fuzzy set and r be a real number in $]0,1]$. The r*-cut* of $\widetilde{\mathrm{E}}$ is the following set

$$\widetilde{\mathrm{E}}_r = \{x \in \mathrm{X} \mid \mu_{\widetilde{\mathrm{E}}}(x) \geq r\}\,.$$

The 0-cut $\widetilde{\mathrm{E}}_0$ is the closure of *Supp*($\widetilde{\mathrm{E}}$). A fuzzy set is said *convex* if each of its r-cuts is convex for all r in $[0,1]$.

Some examples of r-cuts for finite and infinite supports appear in Figures 1 to 3. They represent specific fuzzy sets of $\mathbb{R}$, defined below:

Definition 2.3. Let n be a real number. A *fuzzy number* $\tilde{n}$ is a convex fuzzy set whose membership function $\mu_{\tilde{n}}$ from $\mathbb{R}$ to the real interval $[0,1]$ is continuous and satisfies $\mu_{\tilde{n}}^{-1}(\{1\}) = \{n\}$; i.e. n is the only real with grade of membership 1. The value n is called the *mean value* of the fuzzy number $\tilde{n}$.

According to the definition of a fuzzy number $\tilde{n}$, the grade of membership of a real x increases when x approaches the mean value n. Two fuzzy numbers are equal if they have the same membership functions [20].

Remark 2.4. In literature there are more general definitions than the one given above. They include the so-called *trapezoidal* fuzzy numbers, for which the grade of membership equals 1 over an interval of $\mathbb{R}$ containing the mean value. Definition 2.3 excludes them for the sake of clarity in our study. Note that most applications make use of non-trapezoidal numbers. However, we will show in Section 3.6 that our analysis, once established, simply extends to the trapezoidal case.

Let $\tilde{n}$ be a fuzzy number and $\mu_{\tilde{n}}$ its membership function. The *left restriction* (resp. *right restriction*) $\mu_{\tilde{n}-}$ (resp. $\mu_{\tilde{n}+}$) is the restriction of $\mu_{\tilde{n}}$ to the left (resp. right) of the mean value n. Some families of fuzzy numbers can be characterized from the functions $\mu_{\tilde{n}-}$ and $\mu_{\tilde{n}+}$. Their denomination derives directly from the nature of these restrictions. The most popular family contains the *triangular* fuzzy numbers for which the functions $\mu_{\tilde{n}-}$ and $\mu_{\tilde{n}+}$ are linear on Supp($\tilde{n}$). Similarly one speaks of fuzzy numbers which are *gaussian*, *quadratic*,... Remark that both restrictions functions are not necessarily of the same type. Then, for example, a triangular-quadratic fuzzy number denotes a fuzzy number $\tilde{n}$ with a left restriction $\mu_{\tilde{n}-}$ linear on Supp($\tilde{n}$)$\cap]-\infty, n]$ and a right restriction $\mu_{\tilde{n}+}$ quadratic on Supp($\tilde{n}$) $\cap [n, +\infty[$.

Figure 1 represents a gaussian fuzzy number $\widetilde{3}$, i.e. of mean value $n = 3$. Its support is infinite and the 0-cut $\widetilde{\mathrm{E}}_0$ of $\widetilde{\mathrm{E}} = \tilde{n}$ is the whole set $\mathbb{R}$.

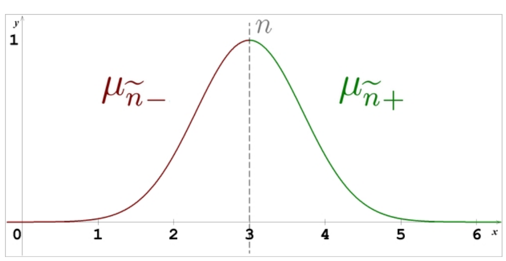

FIGURE 1. Infinite support: a gaussian fuzzy number $\widetilde{3}$; $\mathbb{R}$ is the 0-cut.

2.2. **Tuple representation.**

The tuple representation of a fuzzy number with finite support was proposed in 1978 by D. Dubois and H. Prade in [10]. In this representation the arithmetic operations have very simple expressions as soon as they are performed within a family described in Definition 2.5, based on *spread functions*.

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 continue and decreasing on its domain.

Definition 2.5. Let L and R be two spread functions. A fuzzy number $\tilde{n}$ with a finite support is said of *type* L-R if there exist two positive real numbers α and β such that the membership function $\mu_{\tilde{n}}$ of $\tilde{n}$ is given as follows:

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

In particular, real number$n$ can be identified to fuzzy number $\tilde{n}$ with $\alpha=\beta=0$. The triplet (n,α,β) is called the *tuple representation* of fuzzy number $\tilde{n}$. Real numbers α and β are respectively called the *left spread* and the *right spread* of $\tilde{n}$. Functions L et R are respectively called the *left spread function* and the *right spread function* of the family of fuzzy numbers of type L-R, denoted by $\mathfrak{F}(\mathrm{L},\mathrm{R})$, and by extension the spread functions of $\tilde{n}$ itself.

The support of $\tilde{n}$ (or equivalently of $\mu_{\tilde{n}}$) is thus $]n-\alpha,n+\beta[$ when $\alpha\neq 0$ and $\beta\neq 0$, $[n,n+\beta[$ when $\alpha=0$ and $\beta\neq 0$, $]n-\alpha,n]$ when $\alpha\neq 0$ and $\beta=0$ and the singleton $\{n\}$ when $\alpha=\beta=0$.

The *triangular family*, formed by the triangular fuzzy numbers, is the family $\mathfrak{F}(\mathrm{L},\mathrm{R})$ where the spread functions L and R are both linear. In this case, as $\mathrm{L}(0)=\mathrm{R}(0)=1$ and $\mathrm{L}(1)=\mathrm{R}(1)=0$, we have $\mathrm{L}=\mathrm{R}$ and this spread function equals the function F defined by $\mathrm{F}(x)=-x+1$.

Example 2.6. *Let* $\tilde{3}$ *be a triangular fuzzy number with tuple representation* $(3,2,2)$. *Figures 2 and 3 describe respectively the* $1/2$*-cut* $\widetilde{\mathrm{E}}_{1/2}=[2,4]$ *and the* 0*-cut* $\widetilde{\mathrm{E}}_0=[3-\alpha,3+\beta]=[1,5]$ *where* $\widetilde{\mathrm{E}}=\tilde{3}$.

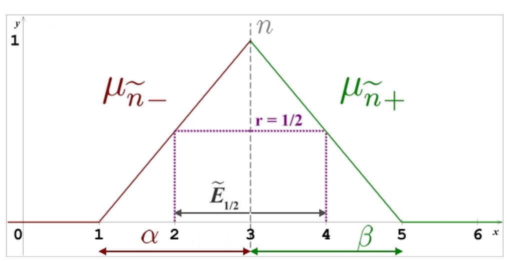

FIGURE 2. Finite support $]1,5[$: $1/2$-cut $\widetilde{\mathrm{E}}_{1/2}=[2,4]$ of the triangular fuzzy number $\widetilde{\mathrm{E}}=\tilde{3}=(3,2,2)$.

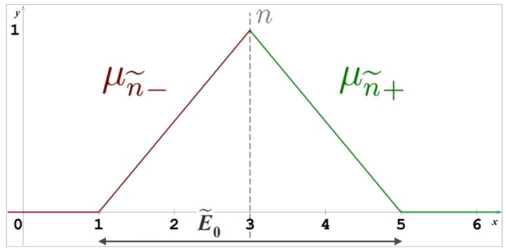

FIGURE 3. Finite support $]1,5[$: 0-cut $\widetilde{\mathrm{E}}_0 = [1,5]$ of the triangular fuzzy number $\widetilde{\mathrm{E}} = \widetilde{3} = (3,2,2)$.

Inside a given family $\mathfrak{F}(\mathrm{L},\mathrm{R})$, a tuple (n,α,β) represents an unique element $\widetilde{n}$. The addition is an internal law of $\mathfrak{F}(\mathrm{L},\mathrm{R})$ defined by:

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

The family $(\mathfrak{F}(\mathrm{L},\mathrm{R}),+)$ is an abelian monoïd with the tuple $0 = (0,0,0)$ as identity element. There exists an approximate product "$\cdot$" such that $(\mathfrak{F}(\mathrm{L},\mathrm{R}),\cdot)$ is an abelian monoïd with the tuple $1 = (1,0,0)$ as identity element. The product is distributive with respect to addition. As we study equations with fuzzy equations with real indeterminates, we consider only products of the form $q\cdot\widetilde{n}$, where $q \in \mathbb{R}$. In this case, the product described by Dubois and Prade becomes exact and is defined by:

$$q\cdot(n,\alpha,\beta) = \begin{cases} (q\,n, q\,\alpha, q\,\beta) & \text{if } q \geq 0 \\ (q\,n, -q\,\beta, -q\,\alpha) & \text{if } q \leq 0 \end{cases} \ . \tag{1}$$

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

$$-\widetilde{n} = -1\cdot(n,\alpha,\beta) = (-n,\beta,\alpha) \quad . \tag{2}$$

2.3. **Parametric representation.**

The parametric representation introduced in 1986 by R. Goetschel et W. Voxman [11] allows them to embed all the trapezoidal fuzzy numbers into a topological vector space. The following definition is an adaptation for non-trapezoidal fuzzy numbers:

Definition 2.7. 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) $\underline{n}(1) = \overline{n}(1) = n$.

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 $\mu(x) = \sup\{r \mid \underline{n}(r) \leq x \leq \overline{n}(r)\}$.

A fuzzy arithmetic, described in the following lemma, operates on parametric representation. It is coherent with those of the tuple representation given in Section 2.2.

Lemma 2.8. [18] *Let $q \in \mathbb{R}$ and $\widetilde{m} = [\underline{m},\overline{m}]$ and $\widetilde{n} = [\underline{n},\overline{n}]$ two fuzzy numbers. Then*

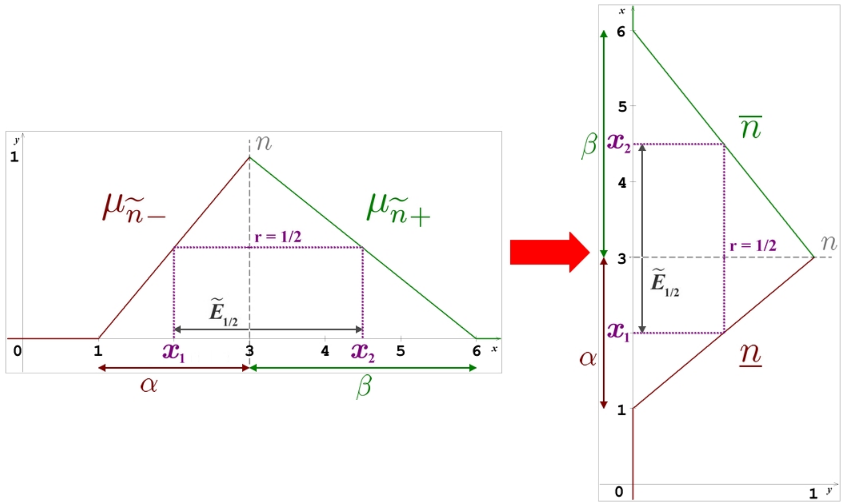

FIGURE 4. Graph of functions $\underline{3}$ and $\overline{3}$ from the graph of a linear membership function

(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) $q \cdot \widetilde{n} = \begin{cases} [q \cdot \underline{n},\ q \cdot \overline{n}] & \textit{if } q \geq 0, \\ [q \cdot \overline{n},\ q \cdot \underline{n}] & \textit{if } q \leq 0 \end{cases}$

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

2.4. **From tuple to parametric representation.**

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 is based on the change of representation given by formulas of Proposition 2.9 below. It is fundamental because we have 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 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 2.7. This relation appears graphically in Figure 4 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. This transformation is illustrated in Figure 4 and in Figure 5 for a quadratic fuzzy number $\widetilde{3} = (3,2,3)$. Formally, it is given by the formulas below.

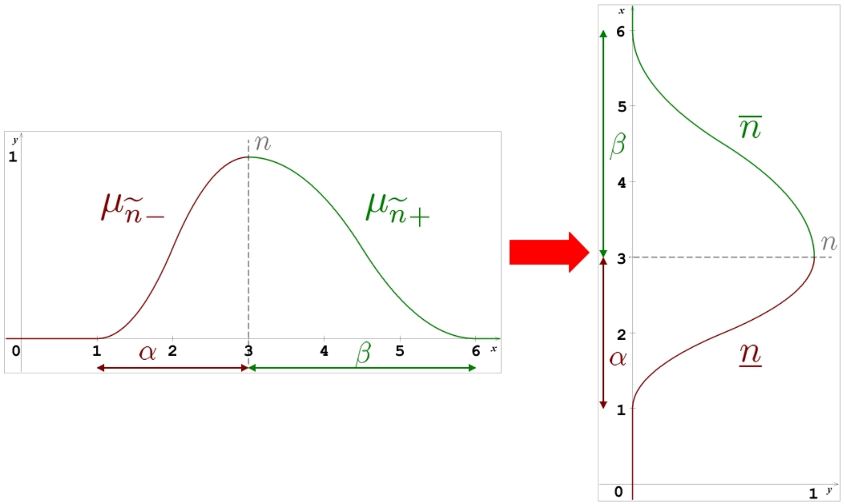

FIGURE 5. Functions $\underline{3}$ and $\overline{3}$ from the graph of a quadratic membership function

Proposition 2.9. *Let* $\tilde{n} = (n, \alpha, \beta) \in \mathfrak{F}(\mathrm{L}, \mathrm{R})$ *where* L *and* R *are bijective spread functions. Then the parametric representation* $[\underline{n}, \overline{n}]$ *of* $\tilde{n}$ *satisfies the following formulas:*

$$\left\{ \begin{array}{l} \underline{n}(r) = n - \alpha\,\mathrm{L}^{-1}(r) \\ \overline{n}(r) = n + \beta\,\mathrm{R}^{-1}(r) \end{array} \right. . \tag{3}$$

In particular, when the fuzzy number $\tilde{n}$ *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 2.5 implies $r = \mathrm{L}\left(\frac{n - \underline{n}(r)}{\alpha}\right) = \mathrm{R}\left(\frac{\overline{n}(r) - n}{\beta}\right)$. 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 $\mathrm{L} = \mathrm{R} = \mathrm{F}$ where $\mathrm{F}(x) = 1 - x$ is bijective with $\mathrm{F}^{-1} = \mathrm{F}$ □

In the rest of the paper the different fuzzy numbers appearing in an equation or a system are assumed to belong to a same family $\mathfrak{F}(\mathrm{L}, \mathrm{R})$.

3. THE REAL TRANSFORM OF A FUZZY EQUATION

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 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 $q \cdot \tilde{n}$ because the spreads depend on the sign of $q \in \mathbb{R}$ (see Lemma 2.8). 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 $\mathrm{I} \in \{-1,1\}^k$. We called the latter equations the *induced* equations of E.

Considering only positive real variables, 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). In the literature this *collected crisp form* is formed by four equations obtained from (E) by an algorithm and only valid for equations whose coefficients are triangular fuzzy numbers. In this paper, the coefficients are only required to have a finite support and bijective spread functions.

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 $\mathscr{T}(\mathrm{E})$.

To obtain the real solutions of (E), it is necessary and sufficient to collect the positive real solutions of the 2^k real transforms $\mathscr{T}(\mathrm{E(I)})$, 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 $\mathscr{T}(\mathrm{E(I)})$. This subjet will be the center of the discussion of Section 4.

Section 3.5 compares the real transform $\mathscr{T}(\mathrm{E})$ 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 $\boldsymbol{d} = (d_1,\dots,d_k) \in \mathbb{N}^k$ and $\boldsymbol{x} = (x_1,\dots,x_k)$. The *monomial* of multidegree $\boldsymbol{d}$ in the variables $x_1,\dots,x_k$ is the product $\boldsymbol{x}^{\boldsymbol{d}} = x_1^{d_1}\cdots x_k^{d_k}$. 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 $a_1^{d_1}\cdots a_k^{d_k}$. We denote by $\boldsymbol{x}\times\boldsymbol{y}$ the classical product $(x_1y_1,\dots,x_ky_k)$ where $\boldsymbol{y} = (y_1,\dots,y_k)$. 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) below whith coefficients $\widetilde{n_{\boldsymbol{d}}}$ and $\widetilde{m}$ belonging to a same family $\mathfrak{F}(\mathrm{L},\mathrm{R})$ whose spread functions L and R are bijective. These numbers are assumed to be known under their respective tuple representation $\widetilde{m} = (m,\alpha,\beta)$ and $\widetilde{n_{\boldsymbol{d}}} = (n_{\boldsymbol{d}},\alpha_{\boldsymbol{d}},\beta_{\boldsymbol{d}})$. Let

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

where Supp(E) is the support of (E), i.e. the finite set of $\boldsymbol{d} \in \mathbb{N}^k$ such that $\widetilde{n_{\boldsymbol{d}}} \neq (0,0,0)$ (we do not take into account the right hand side of (E) which may be zero or not).

We denote by Sol(E) the set of solutions of (E) in $\mathbb{R}^k$:

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

The set $\mathrm{Sol}^+(\mathrm{E})$ of positive solutions of (E) is defined by:

$$\mathrm{Sol}^+(\mathrm{E}) = \{(a_1,\dots,a_k) \in \mathrm{Sol(E)} \mid \forall i \in [\![1,k]\!] \quad a_i \geq 0\} = \mathrm{Sol(E)} \cap {\mathbb{R}^+}^k \;.$$

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 2.8, the r-cut of $\widetilde{n_d}\boldsymbol{x}^{\boldsymbol{d}}$ depends on the sign of $\boldsymbol{x}^{\boldsymbol{d}}$ while this sign is as unknown as those of the indeterminates $x_1,\dots,x_n$.

This is why we need to consider the case where the indeterminates $x_1,\dots,x_n$ are both real and positive. But we want all 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 $\mathrm{Sol}^+(\mathrm{E}(\mathrm{I}))$ 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 $\mathrm{Sol}^+(\mathrm{E}(\mathrm{I}))$. For this perspective, Section 3.3 and 3.4 suppose that the variables are both real and positive. In this context, we seek the formula of the real transform $\mathscr{T}(\mathrm{E})$ of (E). The positive solutions of the real algebraic system $\mathscr{T}(\mathrm{E})$ 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 absolue 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 $\varepsilon(\boldsymbol{a})=(\mathrm{sign}(a_1),\dots,\mathrm{sign}(a_k))\in\{-1,1\}^k$. The value $\varepsilon(\boldsymbol{a})^{\boldsymbol{d}}=\prod_{i=1}^k\mathrm{sign}(a_i)^{d_i}$ 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 $\widetilde{n_d}\boldsymbol{x}^{\boldsymbol{d}}$ 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 2.8, when the sign $\varepsilon(\boldsymbol{a})^{\boldsymbol{d}}$ of $\boldsymbol{a}^{\boldsymbol{d}}$ is 1, the r-cut of the fuzzy number $\widetilde{n_d}\,\boldsymbol{a}^{\boldsymbol{d}}$ is

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

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

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

It is therefore impossible to transform $\widetilde{n_d}x^{\boldsymbol{d}}$ in 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 $\widetilde{n_d}$ and $\boldsymbol{x}$ can be supposed to be positive. It leads to 2^k induced fuzzy equations E(I) defined below.

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

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

Note that $(\mathrm{E})=\mathrm{E}((1,1,\dots,1))$. 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 $\mathrm{I}^{\boldsymbol{d}}\,\widetilde{n_d}\,\boldsymbol{x}^{\boldsymbol{d}}$ in the left side of E(I), $\mathrm{I}^{d}\in\{-1,1\}$ 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 $\mathrm{I}^{\boldsymbol{d}}\,\widetilde{n_d}\,\boldsymbol{x}^{\boldsymbol{d}}$ with $\boldsymbol{x}$ positive whereas it is impossible in (E) for $\widetilde{n_d}\,\boldsymbol{x}^{\boldsymbol{d}}$ with $\boldsymbol{x}$ not necessarily positive.

To describe relationships between the solutions of equations (E) and E(I), for a k-uplet of real numbers $\boldsymbol{q} = (q_1, \dots, q_k)$ and any subset $\mathscr{S}$ of $\mathbb{R}^k$ we introduce the following notation:

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

Lemma 3.2. *Let* $\mathrm{I} \in \{-1, 1\}^k$ *and* E(I) *be an induced equation of equation* (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 $\mathrm{I} \times \mathrm{I} = (1, \dots, 1)$, 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

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

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 3.3. *Consider a fuzzy algebraic equation* (E). *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 3.2 that each $\mathrm{I} \otimes \mathrm{Sol}^+(\mathrm{E(I)}) \subset \mathrm{Sol(E)}$. Conversely, if $\boldsymbol{a} \in \mathrm{Sol(E)}$ 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 E(I) $\square$

Remark 3.4. The coefficients in (E) being given by their tuple representation, the equations E(I) actually derive directly from E. For every d in Supp(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 it is equal to $(-n_d, \beta_d, \alpha_d)$ when $\mathrm{I}^{\boldsymbol{d}} = -1$, following Equality (2). This ensures that for finding the whole set of real solutions of an equation (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 any $\boldsymbol{d} = (d_1, \dots, d_k) \in \mathrm{Supp(E)}$ is even, all induced equations E(I) are identical to equation (E).

Example 3.5. *Take* $k = 3$ *variables* x_1, x_2, x_3 *and consider the equation*

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

There are $8 = 2^3$ 3*-uplets in* $\{-1,1\}^3$*:* $\mathrm{I}_1 = (1,1,1), \mathrm{I}_2 = (1,1,-1), \mathrm{I}_3 = (-1,1,1), \mathrm{I}_4 = (-1,1,-1), \mathrm{I}_5 = (1,-1,1), \mathrm{I}_6 = (1,-1,-1), \mathrm{I}_7 = (-1,-1,1)$ *and* $\mathrm{I}_8 = (-1,-1,-1)$*. There are only two distinct induced equations* $\mathrm{E}(\mathrm{I}_j)$*. Indeed, since Supp*$(\mathrm{E}) = \{\boldsymbol{d}_1 = (2,1,0), \boldsymbol{d}_2 = (0,0,4)\}$*, we have* $\mathrm{I}_j^{\boldsymbol{d}_2} = 1$ *for all* $j \in [[1,8]]$ *; thus the eight* I_j *split up into the two following groups: that of* $j \in [[1,4]]$ *with* $\mathrm{I}_j^{\boldsymbol{d}_1} = 1$ *and that of* $j \in [[5,8]]$ *with* $\mathrm{I}_j^{\boldsymbol{d}_1} = -1$*. Hence* $\mathrm{E}(\mathrm{I}_1) = \mathrm{E}(\mathrm{I}_j)$ *for all* $j \in [[1,4]]$ *and* $\mathrm{E}(\mathrm{I}_5) = \mathrm{E}(\mathrm{I}_j)$ *for all* $j \in [[5,8]]$*. According to Theorem 3.3, we have*

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

We have $\mathrm{E}(\mathrm{I}_j) = (\mathrm{E})$ *for* $j \in [[1,4]]$ *and the tuple representation of the coefficients of* $\mathrm{E}(\mathrm{I}_j)$ *for* $j \in [[5,8]]$ *is determined with the help of Remark 3.4. 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*

$$\mathrm{E}(\mathrm{I}_j) : (-3,\beta_1,\alpha_1)x_1^2x_2 + (1,\alpha_2,\beta_2)x_3^4 = (6,\alpha,\beta) \quad \textit{for } j \in [[5,8]].$$

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 3.4 illustrated in previous example.

3.3. **Crisp form of** (E) **to find** $\mathrm{Sol}^+(\mathrm{E})$**.**

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 2.8, equation (E) rewrites into two equalities on the r-cuts of the left and right members of (E) if all the $\boldsymbol{x}^{\boldsymbol{d}}$, $\boldsymbol{d} \in \mathrm{Supp}(\mathrm{E})$, 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 to every induced equation E(I).

For $\boldsymbol{a} \in {\mathbb{R}^+}^k$, according to Lemma 2.8, the following equivalence applies:

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

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

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

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

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

where Sol(F) is the set of the solutions of F in $\mathbb{R}^{k+1}$. Take $\boldsymbol{a} = (a_1,\ldots,a_k) \in {\mathbb{R}^+}^k$. 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 $(a_1,\ldots,a_k,r)$ 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 3.6. *Let us take again the fuzzy equation of Example 3.5 with triangular fuzzy numbers as coefficients. We recall that their r-cuts are given by the formulas* (4) *and that the triplets of the common support to* $\mathrm{E}(\mathrm{I}_1) = (\mathrm{E})$ *and* $\mathrm{E}(\mathrm{I}_5)$ *are* $\boldsymbol{d}_1 = (2,1,0)$ *and* $\boldsymbol{d}_2 = (0,0,4)$. *The parametric representations of the coefficients of the equation are then written as follows:*

- *for both* $\mathrm{E}(\mathrm{I}_1)$ *and* $\mathrm{E}(\mathrm{I}_5)$, *we have* $\widetilde{n_{\boldsymbol{d}_2}} = \widetilde{1} = [\alpha_2 r + 1 - \alpha_2, -\beta_2 r + 1 - \beta_2]$ *and* $\widetilde{6} = [\alpha r + 6 - \alpha, -\beta r + 6 + \beta]$;
- *for* $\mathrm{E}(\mathrm{I}_1)$, *we have* $\widetilde{n_{\boldsymbol{d}_1}} = \widetilde{3} = [\alpha_1 r + 3 - \alpha_1, -\beta_1 r + 3 - \beta_1]$;
- *for* $\mathrm{E}(\mathrm{I}_5)$, *we have* $\widetilde{n_{\boldsymbol{d}_1}} = -\widetilde{3} = [\beta_1 r - 3 - \beta_1, -\alpha_1 r - 3 - \alpha_1]$.

Therefore the respective crisp forms of the two distincts induced equation of (E) *which are necessary and sufficient to obtain the real solutions of* (E) *are:*

$$\mathscr{C}(\mathrm{E}(\mathrm{I}_1)) : \left\{ \begin{array}{lcll} (\alpha_1 r + 3 - \alpha_1)\, x_1^2 x_2 & + & (\alpha_2 r + 1 - \alpha_2)\, x_3^4 & = \alpha r + 6 - \alpha \\ (-\beta_1 r + 3 - \beta_1)\, x_1^2 x_2 & + & (-\beta_2 r + 1 - \beta_2)\, x_3^4 & = -\beta r + 6 + \beta \end{array} \right. \quad \textit{and}$$

$$\mathscr{C}(\mathrm{E}(\mathrm{I}_5)) : \left\{ \begin{array}{lcll} (\beta_1 r - 3 - \beta_1)\, x_1^2 x_2 & + & (\alpha_2 r + 1 - \alpha_2)\, x_3^4 & = \alpha r + 6 - \alpha \\ (-\alpha_1 r - 3 - \alpha_1)\, x_1^2 x_2 & + & (-\beta_2 r + 1 - \beta_2)\, x_3^4 & = -\beta r + 6 + \beta \end{array} \right. \quad .$$

In the particular triangular case, illustrated in Example 3.6, 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). 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 general case, when the spread functions are simply bijective, requires using inversion formulas (3) with two indeterminates instead of only one. This leads to the crisp form with two parameters in the following theorem:

Theorem 3.7. *Let* L *and* R *be two spread functions and*

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

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* $\boldsymbol{d} \in Supp(\mathrm{E})$. *If the spread functions* L *and* R *are bijective then the crisp form of* (E) *is given by:*

$$\mathscr{C}(\mathrm{E}) : \left\{ \begin{array}{lcl} \sum_{\boldsymbol{d}} n_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} - m + (\alpha - \sum_{\boldsymbol{d}} \alpha_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}})\, u & = & 0 \\ \sum_{\boldsymbol{d}} n_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} - m + (-\beta + \sum_{\boldsymbol{d}} \beta_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}})\, v & = & 0\, , \end{array} \right. \tag{9}$$

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* $(a_1, \ldots, a_k, \mathrm{L}^{-1}(r), \mathrm{R}^{-1}(r))$.

Proof. By definition, a spread function H sends $[0,1]$ to itself and if moreover H is bijective then its inverse H^{-1} is continue 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 $r \in [0,1]$, 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. According to formulas (3), the parametric form of the coefficients of the equation are given by

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

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:

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

By collecting the respective factors of u and v as well as the constant factor into both u and v, the crisp form is thus expressed as we want in the form (9) of the theorem.

Last assertion in the theorem about $\mathrm{Sol}^+(\mathrm{E})$ follows directly from equality (8) □

Note that Theorem 3.7 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 [13] and [7] 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, for any family $\mathfrak{F}(\mathrm{L},\mathrm{R})$ such that the spread functions L and R are bijective. 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 3.8. Let L and R be two spread functions and

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

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

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

This definition naturally extends to a system (S) of fuzzy equations such (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 3.9. *Under the assumptions of Definition 3.8, 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; in other words:*

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

Proof. As the spread functions L and R are bijective, we can apply Theorem 3.7. Let be $\boldsymbol{a} \in {\mathbb{R}^+}^k$. 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 $\sum_{\boldsymbol{d}} n_{\boldsymbol{d}} \boldsymbol{x}^{\boldsymbol{d}} = m$. 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 $\sum_{\boldsymbol{d}} n_{\boldsymbol{d}} \boldsymbol{x}^{\boldsymbol{d}} - m$ 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 $\mathrm{G}^{-1}(u) = \mathrm{D}^{-1}(v) \in [0,1]$. Hence any positive real solution of the real transform $\mathscr{T}(\mathrm{E})$ is also a positive real solution of the fuzzy equation (E) □

Theorem 3.9 ensures that finding the positive real roots of (E) amounts to finding the positive real roots of its real transform $\mathscr{T}(\mathrm{E})$. Therefore it is no use to develop intermediate computations on parametric representation like the previous methods did in the specific triangular case.

From the expression of the real transform $\mathscr{T}(\mathrm{E})$ of E, we can now derive those of its induced equations E(I). With reference to Remark 3.4 we immediately obtain the following corollary:

Corollary 3.10. *Under the assumptions of Definition 3.8, let* $\mathrm{I} \in \{-1, 1\}^k$ *and* E(I) *the induced equation of* E *defined in* (6)*. Then the real transform of* E(I) *is given by:*

$$\mathscr{T}(\mathrm{E}(\mathrm{I})) : \left\{ \begin{array}{lcl} \sum\limits_{\boldsymbol{d} | \mathrm{I}^{\boldsymbol{d}} > 0} n_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} - \sum\limits_{\boldsymbol{d} | \mathrm{I}^{\boldsymbol{d}} < 0} n_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} & = & m \\ \sum\limits_{\boldsymbol{d} | \mathrm{I}^{\boldsymbol{d}} > 0} \alpha_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} + \sum\limits_{\boldsymbol{d} | \mathrm{I}^{\boldsymbol{d}} < 0} \beta_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} & = & \alpha \\ \sum\limits_{\boldsymbol{d} | \mathrm{I}^{\boldsymbol{d}} > 0} \beta_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} + \sum\limits_{\boldsymbol{d} | \mathrm{I}^{\boldsymbol{d}} < 0} \alpha_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} & = & \beta \quad . \end{array} \right. \tag{12}$$

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

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 [13] and [7] compute a collected crisp form of (S) formed by $4\,s$ real algebraic equations. In this part we are interested in the relationship between their collected crisp form with $4\,s$ equations and our particular collected crisp system with $3\,s$ equations, that is the real transform of (S). For both systems their positive real solutions are the same than those of (S). It is the principle of any collected crisp form of (S).

Consider below the system F_1 of Section 6 in [7]:

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

Applied to first equation, the algorithm proposed in [7] produces the following collected crisp form:

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

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

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

Call T_1 the system formed of the eight preceding equations.

Besides, 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 [7] on the generic equation (5) of (E).

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

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

Unlike our treatment of the general case, the transformation does not require the use of $u = L^{-1}(r)$ and $v = R^{-1}(r)$ where the spread functions L and R are bijective but not determined. For a triangular fuzzy number, $L = R = F$, where $F(x) = F^{-1}(x) = 1 - x$, being known, the previous methods replace directly $L^{-1}(r)$ and $R^{-1}(r)$ 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:

$$\mathscr{C}(\mathrm{E}) : \begin{cases} (\sum_{\boldsymbol{d}} \alpha_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} - \alpha) r + \sum_{\boldsymbol{d}} (n_{\boldsymbol{d}} - \alpha_{\boldsymbol{d}})\, \boldsymbol{x}^{\boldsymbol{d}} - m + \alpha = 0 \\ (\beta - \sum_{\boldsymbol{d}} \beta_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}}) r + \sum_{\boldsymbol{d}} (n_{\boldsymbol{d}} + \beta_{\boldsymbol{d}})\, \boldsymbol{x}^{\boldsymbol{d}} - m - \beta = 0\,. \end{cases}$$

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 each coefficient w.r.t. the variable r of these independent equations is zero. The collected crisp form of (E) is therefore written

$$\begin{cases} \sum_{\boldsymbol{d}} \alpha_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} & = & \alpha \\ \sum_{\boldsymbol{d}} (n_{\boldsymbol{d}} - \alpha_{\boldsymbol{d}})\, \boldsymbol{x}^{\boldsymbol{d}} & = & m - \alpha \\ \sum_{\boldsymbol{d}} \beta_{\boldsymbol{d}}\, \boldsymbol{x}^{\boldsymbol{d}} & = & \beta \\ \sum_{\boldsymbol{d}} (n_{\boldsymbol{d}} + \beta_{\boldsymbol{d}})\, \boldsymbol{x}^{\boldsymbol{d}} & = & m + \beta\,. \end{cases}$$

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.**

As mentioned in Remark 2.4, our results adapt to trapezoidal fuzzy numbers with finite support. The latter numbers extend Definition 2.3 that we give about fuzzy numbers by allowing $\mu_{\widetilde{n}}^{-1}(\{1\})$ to be an interval $[a,b]$. In this context, a fuzzy number $\widetilde{n}$ with finite support is of type L-R if its membership function $\mu_{\widetilde{n}}$ has the following form:

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

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 2.9 takes the following form for a trapezoidal number of type L – R whose spread functions L and R are bijective:

$$\begin{cases} \underline{n}(r) = a - \alpha \mathrm{L}^{-1}(r) \\ \overline{n}(r) = b + \beta \mathrm{R}^{-1}(r) \quad . \end{cases} \tag{13}$$

When the equation $(\mathrm{E}) : \sum_{\boldsymbol{d} \in \mathrm{Supp(E)}} \widetilde{n_{\boldsymbol{d}}}\, \boldsymbol{x}^{\boldsymbol{d}} = \widetilde{m}$ has 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 the tuple representations of the coefficients, the parametric forms (10) given in the proof of Theorem 3.7 become

$$\begin{array}{rclcrcll} \underline{n_{\boldsymbol{d}}}(r) & = & a_{\boldsymbol{d}} - \alpha_{\boldsymbol{d}}\, u & , & \overline{n_{\boldsymbol{d}}}(r) & = & b_{\boldsymbol{d}} + \beta_{\boldsymbol{d}}\, v & \text{for } \boldsymbol{d} \in \mathrm{Supp(E)} \quad ; \\ \underline{m}(r) & = & a - \alpha\, u & , & \overline{m}(r) & = & b + \beta\, v & . \end{array}$$

Applying the argument of Section 3.4 (here L(1) = L(1) = 0 and R(0) = R(0) = 1), we obtain in the same way a real transform of (E), but this time with four equations:

$$\mathscr{T}(\mathrm{E}):\begin{cases}\sum_d a_d\, x^d &= a\\ \sum_d b_d\, x^d &= b\\ \sum_d \alpha_d\, x^d &= \alpha\\ \sum_d \beta_d\, x^d &= \beta\,.\end{cases} \tag{14}$$

Consequently the use of the real transform presented further for solving polynomial fuzzy systems will directly transpose to systems with trapezoidal fuzzy coefficients.

The algorithms proposed in this article will remain valid for 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 follows directly from the results for a single equation. In Section 3, we deduce from equation (E) 2^k fuzzy induced equations E(I) where I runs throughout the 2^k k-uplets of $\{-1,1\}^k$. Following Theorem 3.3 the solutions of (E) are deduced from the positive solutions of every fuzzy induced equation E(I). And following Corollary 3.10, the positive solutions of each E(I) are the positive solutions of the real transform $\mathscr{T}(\mathrm{E(I)})$. In the same way, we will consider 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 fuzzy equations of the system (S). 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 sytem 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 3.3 and Corollary 3.10:

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

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

A first algorithm for the real solving of fuzzy systems, called `BA-SFS`, derives naturally from Theorem 4.1 and the results above. 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 $\mathfrak{F}(\mathrm{L},\mathrm{R})$ whose spread functions L and R are bijective. It is based on the following functions:

- The function **Multisign**(j) is the natural bijection between the interval $[[0, 2^k - 1]]$ and $\{-1, 1\}^k$: the image of the integer $j = \sum_{i=0}^{2^k-1} b_i 2^i \in [[0, 2^k - 1]]$ is the k-uplet $\mathrm{I} = (c_o, \dots, c_{2^k-1}) \in \{-1, 1\}^k$ with $c_i = 2b_i - 1$;
- the function **SolPos**(SR) returns the positive real solutions of a system SR of polynomial equations whith coefficients in $\mathbb{R}$. It can be based on any known computer algebra method. In the implementation described in section 6 it is Wu's method of decomposition into triangular polynomial systems that is used [19];
- the function **RealTransform**(S, I) applies formula (12) to return the $3s$ equations of the real transform of the fuzzy system S(I) with 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 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)
  $sol := sol \bigcup \mathrm{I}\otimes \mathrm{PR}$
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 Supp(S) the support of (S), that is to say 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 of a 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 3.5 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 the support of (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 Supp(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):

$$\mathrm{M(S)} = \begin{array}{cc} & \mathrm{I} \\ d & \begin{pmatrix} & \vdots & \\ \dots & \mathrm{I}^{\boldsymbol{d}} & \\ & & \end{pmatrix} \end{array}$$

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 I_1 and I_2. If $\mathrm{C(I_1)} = \mathrm{C(I_2)}$ then $\mathrm{S(I_1)} = \mathrm{S(I_2)}$. Indeed, 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 $\mathrm{E(I_2)}$, according to the definition of the induced equations given in (6).

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.

5. Algorithm `SolveFuzzySystem`

This section proposes an optimized algorithm `SolveFuzzySystem`, or `SFS` in contracted form, for computing the real solutions of a polynomial fuzzy systems. The coefficients of the systems are fuzzy numbers given under tuple representation and are supposed to belong to a same family $\mathfrak{F}(\mathrm{L},\mathrm{R})$ where the spread functions L and R are 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 the first algorithm `BA-SFS`, algorithm `SolveFuzzySystem` (or `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 is 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 $DistinctColumns$.

As explained in the section 4.2, two cases can avoid redundant calculations when the current column is I:

(1) If 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. This is the first step of the algorithm (see the **4:** statement).

(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 adds to the following index cpt 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 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**. The system S(I) is added to the $DistinctSystems$ vector at the cpt index, which contains the fuzzy (distinct) systems whose positive solutions were calculated and stored in a lb vector at the same corresponding index. The column C(I) in $DistinctColumns$ is kept at index cpt also; 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 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 occurence 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 successively the functions **RealTransform** and **SolPos**). 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.

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

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

```
Require: S, a polynomial system with fuzzy coefficients under tuple representation
         k, the dimension, i.e. the number of variables
Data:    cpt := −1, the counter (up to a unit) 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 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)
        i := IsIn( C , DistinctColumns)
 3:     if i ≠ −1 then
 4:         # lb[i] contains the positive real solutions of S(I): apply Theorem4.1
            sol := sol ∪ I ⊗ 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
        i := IsIn( S(I) , DistinctSystems )
 6:     if i ≠ −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 ∪ I ⊗ lb[cpt]

11: end for

12: return sol
```

terminated, its positive solutions will be stored in a new vector $\mathrm{SP}os$, with same indexes as $\mathrm{D}istinct\mathrm{S}ystems$.

The general parallel algorithm takes place in three steps, as follows.

(1) Detect efficiently the distinct systems to solve in parallel. Moreover, for each column I of $\mathrm{M(S)}$ (and equivalently for each j from 0 to $2^k - 1$), the index of the first previous system equals to the real transform of $\mathrm{S(I)}$ is affected to $lb[j]$ (it can be j itself if $\mathrm{S(I)}$ is new). The distinct systems are stored in the vector $\mathrm{D}istinct\mathrm{S}ystems$.
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 $\mathrm{TRSI} :=$ **RealTransform**$(\mathrm{S(I)})$ and $lb[cpt] :=$ **SPos** (TRSI) by $lb[j] := cpt$.

(2) In parallel, solves every distinct system of the vector $\mathrm{D}istinct\mathrm{S}ystems$. For each system $\mathrm{SI} := \mathrm{D}istinct\mathrm{S}ystems[cpt]$, if it exists (see Remark 5.1), we apply the following statements:
   TRSI := **RealTransform**(SI)
   $\mathrm{SP}os[cpt] :=$ **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 $\mathrm{S(I)}$ in the vector $\mathrm{SP}os$ where $\mathrm{I} =$ **MultiSign**(j). This step is realized by the following loop:
   for $j := 0$ **to** $2^k - 1$ **do**
      $sol := sol \bigcup$ **MultiSign**$(j) \otimes \mathrm{SP}os[lb[j]]$
   end for

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

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

6. Implementation of the algorithm SFS and Examples

We already have a package `Fuzzy` under `SageMath` written by J. Marrez [12]. It contains a function `resolution_reelle_systemes_flous` that implements algorithm `SFS` of Section 5. Recall that fuzzy numbers must have bounded support and must lies in a family $\mathfrak{F}(\mathrm{L},\mathrm{R})$ where L and R are bijective.

In Section 6.1 we describe this 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 presents on another fuzzy system the computations performed by the parallel version of the algorithm.

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)`);`
- as usually, a term $\boldsymbol{x}^{\boldsymbol{d}}$ is represented by rep($\boldsymbol{x}^{\boldsymbol{d}}$) = $x_1 ** d_1 \cdots x_k ** d_k$;
- a monomial $\mathrm{M} = \widetilde{n}\,\boldsymbol{x}^{\boldsymbol{d}}$ is represented by the pair rep(M) = (rep($\widetilde{n}$), rep($\boldsymbol{x}^{\boldsymbol{d}}$));
- a polynomial p with fuzzy coefficients is represented by rep(p), the list of the rep(M) where M ∈ Supp(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

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

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 $\mathrm{I} \in \{-1,1\}^k$, returns the real transform of the induced fuzzy system S(I), that is the system of $3\,s$ equations with real coefficients obtained by transforming each of its equations E(I) as in Corollary 3.10. 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 [19]. 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(`Sr`) = $\bigcup_{\mathrm{B}\in\mathrm{Z}} \mathrm{V(B)} \setminus \mathrm{V}(\mathrm{I_B})$ où $\mathrm{I_B} = \prod_{b\in\mathrm{B}} init(b)$. One can find the definition of the initial $init(b)$ of a polynomial b in ([8]) or in [7], 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 : As in [7] (example 6.1), the call `resolution_reelle_systemes_flous(F,2)` returns the variety $V(F) = \{(x = -1, y \pm 1)\}$, solution of the system F given above. 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 of the support as follows: $x, 1, y^2$, the matrix of signs M(F) is

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

Note that in the performance of the algorithm below, a column 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 $I \in \{I_1, \ldots, I_4\}$; 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 the function `resolution_reelle_systemes_flous(F,2)` by linking the real parameters to formal parameters: `S=F` et `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 $\mathrm{V(F)} = \{(x=-1, y\pm 1)\}$.

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

$$\mathrm{F}_1 : \begin{cases} (2,1,1)xy + (3,1,1)x^2y^2 + (2,1,1)x^3y^3 = (7,3,3), \\ (5,1,1)xy + (2,3,1)x^2y^2 + (2,2,1)x^3y^3 = (9,6,3) \end{cases}$$

The solution returned by the function `resolution_reelle_systemes_flous(F`$_1$`,2)` of the Fuzzy package is the variety $\mathrm{V}(\mathrm{F}_1) = \{(x = \frac{1}{y}, y) \mid y \in \mathbb{R}\backslash\{0\})\}$, like in [7]. 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

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

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 $\mathrm{F}_1(\mathrm{I})$. 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 $\mathrm{V}(\mathrm{F}_1)$ 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 support of F_1 is $\{x^3y^3, x^2y^2, xy, 1\}$. The matrix of signs of F_1 is the following:

$$\begin{array}{c|cccc} & I_0 & I_1 & I_2 & I_3 \\ \hline x^3y^3 & 1 & -1 & -1 & 1 \\ x^2y^2 & 1 & 1 & 1 & 1 \\ xy & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{array}$$

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 will be found 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 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. The four results are added in the variable `sol` containing the real solutions of the fuzzy system F_1. Concretely, this gives:

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+')])

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

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

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 $\mathscr{T}(\mathrm{S})$ of the system (S). As a main property, it has the same positive solutions as (S) (Theorem 3.9).

Unlike the previous methods that were restricted to triangular fuzzy numbers, our results apply to any family $\mathfrak{F}(\mathrm{L},\mathrm{R})$ 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 fuzzy coefficients, 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 previous methods. 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, and 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 also thank Binh-Minh Bui-Xuan for his remarks following his careful reading of the paper.

REFERENCES

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

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

[3] 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.

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

[5] 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.

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

[7] 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.

[8] 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.

[9] D. Dubois, E. Kerre, R. Mesiar, and H. Prade. *Fuzzy Interval Analysis*, 483–581. Springer US, Boston, MA, 2000.
[10] D. Dubois and H. Prade. Operations on fuzzy numbers. *International Journal of systems science*, 9(6):613–626, 1978.
[11] R. Goetschel and W. Voxman. Elementary fuzzy calculus. *Fuzzy Sets Syst.*, 18(1):31–43, January 1986.
[12] J. Marrez. *Bibliothèque Fuzzy en SageMath, Documentation. Preprint HAL-CNRS*, HAL-ID hal-01663476, 2017. http://hal.upmc.fr/hal-01663476 .
[13] 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.
[14] N.A. Rahman and L. Abdullah. Multi-solutions of fuzzy polynomials: a review. *Int. Journal of Engineering and Applied sciences*, 3(3):8 – 16, 2013.
[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] F. Rouillier. Solving zero dimensional systems through the Rational Univariate Representation. *AAECC*, 9(5):433–461, 1999.
[17] The Sage Developers. *SageMath, the Sage Mathematics Software System*, 2017. http://www.sagemath.org.
[18] L. Stefanini and L. Sorini. Fuzzy arithmetic with parametric lr fuzzy numbers. In *IFSA/EUSFLAT Conf.*, 600–605, 2009.
[19] W. T. Wu. A zero structure theorem for polynomial equations solving. *MM Research Preprints*, 1:2–12, 1987.
[20] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965.

Philippe Aubry, LIP6, SORBONNE UNIVERSITÉ, CNRS, 75252 PARIS CEDEX 05, FRANCE
E-mail address, `philippe.aubry@@lip6.fr`

Jérémy Marrez, LIP6, SORBONNE UNIVERSITÉ, CNRS, 75252 PARIS CEDEX 05, FRANCE
E-mail address, `jeremy.marrez@@lip6.fr`

Annick Valibouze, LIP6 AND LSTA, SORBONNE UNIVERSITÉ, CNRS, 75252 PARIS CEDEX 05, FRANCE
E-mail address, `annick.valibouze@@lip6.fr`