



Résolution réelle de systèmes polynomiaux flous

Philippe Aubry, Jérémy Marrez, Annick Valibouze

► **To cite this version:**

Philippe Aubry, Jérémy Marrez, Annick Valibouze. Résolution réelle de systèmes polynomiaux flous. 2018. hal-01766553

HAL Id: hal-01766553

<https://hal.sorbonne-universite.fr/hal-01766553>

Submitted on 13 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

RÉSOLUTION RÉELLE DE SYSTÈMES POLYNOMIAUX FLOUS

PHILIPPE AUBRY, JÉRÉMY MARREZ, ANNICK VALIBOUZE

RÉSUMÉ. Cet article présente un algorithme efficace appelé SolveFuzzySystem, ou SFS, permettant de trouver les solutions réelles des systèmes algébriques dont les coefficients sont des nombres flous à support fini et de fonctions de dispersion bijectives. Les solutions réelles d'un tel système sont déduites des solutions de systèmes algébriques à coefficients réels. L'algorithme est basé sur de nouveaux résultats universels puisqu'indépendants des fonctions de dispersion. Ces résultats théoriques incluent la gestion des signes des solutions des systèmes flous. Une implantation dans le paquetage Fuzzy du logiciel libre de calcul formel SageMath et une version parallèle de l'algorithme sont décrites.

1. INTRODUCTION

La modélisation de problèmes avec des données incertaines a d'importantes applications en ingénierie, économie et sciences sociales ([4]). Dans ce cadre, de nombreux chercheurs se sont intéressés à la recherche des solutions réelles d'une équation polynomiale à coefficients flous. Cette question est en particulier centrale pour l'interpolation de fonctions floues ([1]). Les méthodes de résolution se sont d'abord appuyées sur des techniques locales ([2], [16], [3], [15]). Puis ces dernières années s'est développée une approche globale dans laquelle nous nous inscrivons et faisant appel à des techniques algébriques classiques ([14, 7]). Celles-ci peuvent effectivement être utilisées avec les nombres flous, qui, en dépit d'une dénomination pouvant prêter à confusion, bénéficient d'une définition tout-à-fait formelle.

Nous revisitons cette approche et la renforçons. Jusqu'alors, les méthodes antérieures tant locales que globales se sont focalisées sur les nombres flous dits *triangulaires*. Les résultats présentés ici s'appliquent plus généralement aux nombres flous à support fini et de fonctions de dispersion inversibles, que sont en particulier les nombres flous triangulaires de fonctions de dispersion linéaires, mais aussi, par exemple, les nombres flous quadratiques, de fonctions de dispersion quadratiques.

Pour la résolution des systèmes algébriques, chaque coefficient flou émanant de l'expérimentation est donné sous une certaine représentation, dite en tuple, qui, bien que formelle, est non traitable par des méthodes algébriques usuelles (bases de Gröbner [6], décomposition triangulaire [5], représentation univariée rationnelle [17], ...). Néanmoins, pour tout nombre flou à support fini et de fonctions de dispersion inversibles, cette représentation en tuple est transformable dans une autre représentation, dite paramétrique, à coefficients non plus flous mais réels. Nous donnons l'expression de la représentation paramétrique en fonction des inverses de ses fonctions de dispersion (voir Proposition 2.9).

Date: 21 février 2018.

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

Nous montrons, comment dans le cadre des nombres flous à support fini et de fonctions de dispersion inversibles, la résolution d'un système (S) de s équations à k variables et à coefficients flous se ramène à la recherche des solutions positives de 2^k systèmes, chacun de $3s$ équations à k variables et à coefficients réels (Théorèmes 3.9 et 4.1). Ce nouveau système noté $\mathcal{T}(S)$ est appelé la *transformée réelle* de (S). Nous étendons ce résultat aux nombres flous dits *trapézoïdaux* pour obtenir alors $4s$ équations à la place de $3s$ (Section 3.6). Après la description d'un algorithme de base nécessitant la résolution de 2^k systèmes, nous proposons l'algorithme optimisé `SolveFuzzySystem` qui réduit le nombre de systèmes à résoudre.

La section 2 est consacrée aux nombres flous, leurs différentes représentations ainsi que le passage de l'une à l'autre dans le cas d'un nombre flou à support fini. Dans la section 3 nous définissons la transformée réelle et les équations induites d'une équation algébrique floue. Nous montrons que, dans le cas flou triangulaire, la transformée réelle est un système équivalent à la forme tranchée collectée telle qu'elle est calculée par des méthodes antérieures. L'extension des résultats au cas trapézoïdal clôt la section. La section 4 regroupe les résultats de la section précédente pour les appliquer à un système d'équations. Elle établit le théorème principal 4.1 et le traduit en un premier algorithme appelé BA-SFS. Dans la section 5, celui-ci est optimisé en l'algorithme `SolveFuzzySystem`, ou SFS. On s'intéresse alors aux moyens de paralléliser SFS. L'implantation de l'algorithme SFS dans le package Fuzzy en SageMath (voir [9]) est succinctement présentée dans la section 6. Pour finir, des exemples illustrent les versions séquentielles et parallèles de l'algorithme.

2. NOMBRES FLOUS

Nous présentons ci-dessous les généralités sur la théorie des ensembles et des nombres flous, puis les deux représentations de nombres flous classiques : par tuple et paramétrique. Pour aller plus loin, le lecteur pourra s'intéresser à [10]. Nous donnons les formules exprimant la représentation paramétrique d'un nombre flou à partir de sa représentation en tuple (proposition 2.9). Ces formules constituent la pierre angulaire de notre méthode algébrique pour déterminer les solutions dans le corps de réels \mathbb{R} de systèmes polynomiaux flous.

2.1. Généralités.

Les nombres flous se définissent à partir de la notion d'*ensemble flou*. Dans le cadre classique, l'appartenance à un sous-ensemble E de l'ensemble universel X est booléenne. La fonction caractéristique de E est à valeurs dans $\{0, 1\}$, signifiant si l'élément appartient ou pas à E . La théorie des ensembles flous généralise ce concept classique. Ainsi que l'a décrit Zadeh dans son article original [20], un ensemble flou \tilde{E} est une classe d'objets avec un continuum de degrés d'appartenance. Un élément x appartient à l'ensemble flou \tilde{E} avec un certain degré de validité représenté par une fonction à valeurs comprises entre 0 et 1.

Définition 2.1. Un *ensemble flou* \tilde{E} est une paire formée d'un sous-ensemble E de X et de sa *fonction d'appartenance*, $\mu_{\tilde{E}} : X \rightarrow [0, 1]$.

Pour chaque élément x de X , la valeur $\mu_{\tilde{E}}(x)$ est appelé le *degré d'appartenance* de x à \tilde{E} .

Le support $\text{Supp}(f)$ d'une fonction f définie sur un ensemble A est l'ensemble des $a \in A$ tels que $f(a) \neq 0$. Le support de \tilde{E} se définit comme celui de sa fonction d'appartenance : $\text{Supp}(\tilde{E}) := \text{Supp}(\mu_{\tilde{E}})$. Les nombres flous sont définis à partir des ensembles flous à l'aide de

la notion de r -coupe, qui est aussi sous -jacente à la représentation paramétrique qui sera introduite dans la section 2.3.

Définition 2.2. Soit \tilde{E} un ensemble flou et r un réel dans $]0, 1[$. La r -coupe de \tilde{E} est l'ensemble suivant

$$\tilde{E}_r = \{x \in X \mid \mu_{\tilde{E}}(x) \geq r\}$$

La 0-coupe \tilde{E}_0 est la fermeture de $\text{Supp}(\tilde{E})$. Un ensemble flou est dit *convexe* si toutes ses r -coupes le sont, où $r \in [0, 1]$.

Quelques exemples de r -coupes apparaissent pour un support fini comme infini dans les figures 1 à 3. Ils représentent des ensembles flous particuliers de \mathbb{R} définis ci-après :

Définition 2.3. Soit n fixé dans \mathbb{R} . Un *nombre flou* \tilde{n} est un ensemble flou convexe de réels dont la fonction d'appartenance $\mu_{\tilde{n}}$ de \mathbb{R} dans $[0, 1]$ est continue et vérifie $\mu_{\tilde{n}}^{-1}(\{1\}) = \{n\}$; i.e. n est le seul réel qui possède 1 comme degré d'appartenance. On dit que n est la *mode* du nombre flou \tilde{n} .

D'après la définition d'un nombre flou \tilde{n} , le degré d'appartenance à \tilde{n} d'un réel x augmente quand x se rapproche du mode n . Deux nombres flous sont égaux s'ils ont les mêmes fonctions d'appartenance (voir [20]).

Remarque 2.4. La littérature fait état de définitions plus larges que celle donnée ci-dessus en intégrant les nombres flous dits *trapézoïdaux*, c'est-à-dire ceux pour lesquels le degré d'appartenance est égal à 1 sur un intervalle de X contenant le mode. La définition 2.3 les exclut par souci de clarté de notre étude. Notons que la plupart des applications se situent dans le cadre non trapézoïdal. Cependant, une fois établie notre analyse, nous montrerons qu'elle s'étend simplement au cas trapézoïdal (cf section 3.6) .

Fixons \tilde{n} un nombre flou de fonction d'appartenance $\mu_{\tilde{n}}$. On appelle *restriction gauche* (resp. *restriction droite*) et on note $\mu_{\tilde{n}_-}$ (resp. $\mu_{\tilde{n}_+}$) la restriction de $\mu_{\tilde{n}}$ à gauche (resp. à droite) de son mode n . On distingue suivant la nature des fonctions $\mu_{\tilde{n}_-}$ et $\mu_{\tilde{n}_+}$ certaines familles de nombres flous. Leur dénomination dérive directement de la nature de ces restrictions. La famille la plus populaire est celle des nombres flous dits *triangulaires* pour lesquels, sur $\text{Supp}(\tilde{n})$, les fonctions $\mu_{\tilde{n}_-}$ et $\mu_{\tilde{n}_+}$ sont linéaires. De façon similaire on parle de nombres flous *gaussiens*, *quadratiques*,... On notera que les deux restrictions ne sont pas forcément de même nature. Ainsi un nombre flou triangulaire-quadratique désigne un nombre flou \tilde{n} pour lequel $\mu_{\tilde{n}_-}$ est linéaire sur $\text{Supp}(\tilde{n}) \cap]-\infty, n[$ et $\mu_{\tilde{n}_+}$ est quadratique sur $\text{Supp}(\tilde{n}) \cap]n, +\infty[$.

La figure 1 représente un nombre flou gaussien $\tilde{3}$, i.e. de mode $n = 3$. Le support est infini et la 0-coupe \tilde{E}_0 de $\tilde{E} = \tilde{n}$ est égale à \mathbb{R} tout entier.

2.2. Représentation en tuple.

La représentation en tuple d'un nombre flou à support fini fut proposée par D. Dubois et H. Prade en 1977 dans [11]. Elle a l'avantage de faciliter considérablement les opérations arithmétiques sur les nombres flous dès lors qu'ils appartiennent à une même famille type décrite dans la définition 2.5 de cette section et basée sur les *fonctions de dispersion*.

Une fonction H définie sur $[0, +\infty[$ à valeurs dans l'intervalle fermé $[0, 1]$ sera dite *fonction de dispersion* si $H(0) = 1$, $H(1) = 0$ et que H est continue décroissante sur son domaine de définition.

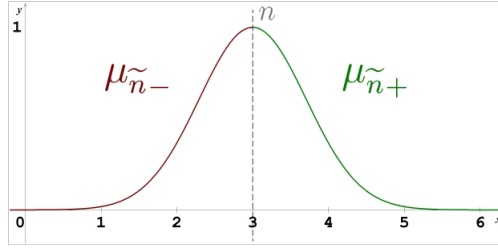


FIGURE 1. Support infini : nombre flou gaussien $\tilde{3}$ de mode $n = 3$: $\tilde{E}_0 = \mathbb{R}$.

Définition 2.5. Soient G et D deux fonctions de dispersion. Un nombre flou \tilde{n} à support fini est dit *de type G-D* si $G(1) = D(1) = 0$ et s'il existe deux réels positifs α et β tels que la fonction d'appartenance $\mu_{\tilde{n}}$ de \tilde{n} est de la forme suivante :

$$\mu_{\tilde{n}}(x) = \begin{cases} G\left(\frac{n-x}{\alpha}\right) & \text{pour } n-\alpha \leq x < n \quad \text{si } \alpha \neq 0 \\ 1 & \text{pour } x = n \\ D\left(\frac{x-n}{\beta}\right) & \text{pour } n < x \leq n+\beta \quad \text{si } \beta \neq 0 \\ 0 & \text{pour } x \in]-\infty, n-\alpha[\cup]n+\beta, +\infty[\end{cases}$$

En particulier, lorsque $\alpha = \beta = 0$, le nombre flou \tilde{n} est le réel n avec $\mu_{\tilde{n}}(x) = \delta_{n,x}$, de support le singleton $\{n\}$.

On dit que le triplet (n, α, β) est la *représentation en tuple* du nombre flou \tilde{n} . Les réels α et β sont respectivement appelés les *dispersions* à gauche et à droite du mode n . Les fonctions G et D sont appelées les *fonctions de dispersion* de la famille des nombres flous de type G-D, notée $\mathfrak{F}(G, D)$, et par extension les fonctions de dispersion du nombre flou lui-même.

Le support de \tilde{n} est celui de sa fonction d'appartenance $\mu_{\tilde{n}}$. Ainsi, le support de \tilde{n} est $]n-\alpha, n+\beta[$ si $\alpha \neq 0$ et $\beta \neq 0$, il est $[n, n+\beta[$ si $\alpha = 0$ et $\beta \neq 0$, il est $]n-\alpha, n]$ si $\alpha \neq 0$ et $\beta = 0$ et il est le singleton $\{n\}$ si $\alpha = \beta = 0$.

La *famille triangulaire*, composée des nombres flous triangulaires, est la famille $\mathfrak{F}(G, D)$ dont les fonctions de dispersion G et D sont toutes deux linéaires. Dans ce cas, comme $g(0) = D(0) = 1$ and $G(1) = D(1) = 0$, on a $G = D$, et ces fonctions de dispersion sont égales à la fonction F définie par $F(x) = -x + 1$.

Exemple 2.6. Choisissons un nombre flou triangulaire $\tilde{3}$ de mode $n = 3$ et de représentation en tuple $(3, 2, 2)$. Les figures 2 et 3 font apparaître respectivement la 1/2-coupe $\tilde{E}_{1/2} = [2, 4]$ et la 0-coupe $\tilde{E}_0 = [3-\alpha, 3+\beta] = [1, 5]$, où $\tilde{E} = \tilde{3}$.

Au sein d'une famille donnée $\mathfrak{F}(G, D)$, un triplet (n, α, β) représente un unique élément \tilde{n} . L'addition est une loi interne à $\mathfrak{F}(G, D)$ définie par

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

La famille des nombres flous de type G-D forme un semi-groupe commutatif pour l'addition, l'élément neutre étant le triplet $(0, 0, 0)$. Il existe en outre un produit approché qui munit la famille $\mathfrak{F}(G, D)$ d'une structure de semi-groupe commutatif avec le triplet $(1, 0, 0)$ comme élément neutre. Ce produit est distributif sur l'addition. Du fait que nous traitons uniquement d'équations à indéterminées réelles, notre étude ne fera intervenir que des produits

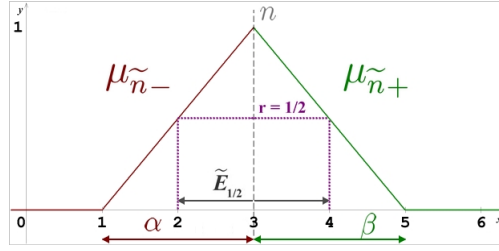


FIGURE 2. Support fini $]1, 5[$: r -coupe $\tilde{E}_{1/2} = [2, 4]$ du nombre flou triangulaire $\tilde{\mathfrak{Z}} = (3, 2, 2)$.

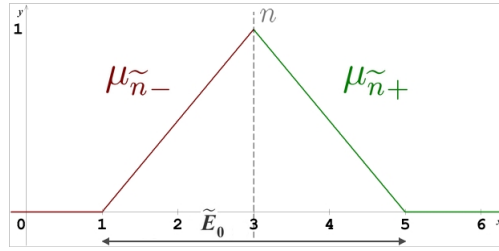


FIGURE 3. Support fini $]1, 5[$: 0-coupe $\tilde{E}_0 = [1, 5]$ du nombre flou triangulaire $\tilde{\mathfrak{Z}} = (3, 2, 2)$.

avec un opérande réel q , i.e. $\tilde{q} = (q, 0, 0)$. Dans ce cas, le produit décrit par Dubois et Prade devient exact, et il est défini comme suit :

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

Noter l'inversion des deux dispersions lorsque q est négatif : En particulier, $-q\beta$ et $-q\alpha$ sont respectivement les dispersions gauche et droite de $q \cdot (n, \alpha, \beta)$.

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

2.3. Représentation paramétrique.

La représentation suivante, introduite en 1986 par Goetschel et Voxman, leur permet de plonger l'ensemble des nombres flous trapézoïdaux dans un espace vectoriel topologique. La définition ci-dessous est adaptée de [12] pour les nombres flous à mode unique.

Définition 2.7. La *forme paramétrique* d'un nombre flou \tilde{n} est une paire ordonnée $[\underline{n}, \overline{n}]$ de fonctions $\underline{n}(r)$ et $\overline{n}(r)$ pour $r \in [0, 1]$, qui satisfont les conditions suivantes :

- (i) \overline{n} est une fonction bornée, continue à gauche et décroissante sur $[0, 1]$,
- (ii) \underline{n} est une fonction bornée, continue à gauche et croissante sur $[0, 1]$,
- (iii) $\underline{n}(1) = \overline{n}(1)$.

Le nombre flou \tilde{n} défini par les fonctions \underline{n} et \overline{n} a pour fonction d'appartenance $\mu : \mathbb{R} \rightarrow [0, 1]$ définie par $\mu(x) = \sup\{r \mid \underline{n}(r) \leq x \leq \overline{n}(r)\}$.

Une arithmétique floue, décrite dans le lemme ci-dessous, opère sur la représentation paramétrique. Elle est cohérente avec celle de la représentation en tuple de la section 2.2. Elle est rappelée dans le lemme 2.8 ci-dessous. L'addition et la multiplication par un scalaire $q \in \mathbb{R}$ sont conformes à l'arithmétique des intervalles.

Lemme 2.8. [18] *Etant donné un nombre réel q et deux nombres flous \tilde{m} et \tilde{n} sous forme paramétrique, les propriétés suivantes s'appliquent :*

$$(1) \quad \tilde{m} = \tilde{n} \text{ si et seulement si } \underline{m}(r) = \underline{n}(r) \text{ et } \overline{m}(r) = \overline{n}(r) \text{ pour chaque } r \in [0, 1],$$

$$(2) \quad \tilde{m} + \tilde{n} = [\underline{m} + \underline{n}, \overline{m} + \overline{n}],$$

$$(3) \quad q \cdot \tilde{n} = \begin{cases} [q \cdot \underline{n}, q \cdot \overline{n}] & \text{si } q \geq 0, \\ [q \cdot \overline{n}, q \cdot \underline{n}] & \text{si } q \leq 0 \end{cases}$$

où, pour toute fonction f de \mathbb{R} vers \mathbb{R} , le produit $g = q \cdot f$ représente la fonction définie par $g(r) = qf(r)$ pour tout $r \in \mathbb{R}$.

2.4. Expression de la représentation paramétrique d'un nombre flou de type G-D.

Notre méthode de résolution s'appliquera à des équations polynomiales dont les coefficients sont des nombres flous d'une même famille $\mathfrak{F}(G, D)$ où G et D sont des fonctions de dispersion nécessairement inversibles. Elle repose sur le changement de représentation donné par les formules de la proposition 2.9 ci-dessous. Celui-ci est fondamental puisque nous devons réécrire algébriquement chaque coefficient flou $\tilde{n} \in \mathfrak{F}(G, D)$ de sa représentation en tuple vers sa représentation paramétrique.

La représentation paramétrique de \tilde{n} a un fort lien avec ses r -coupes puisque les fonctions \underline{n} et \overline{n} définies par

$$\underline{n}(r) = \inf \tilde{n}_r \quad \text{et} \quad \overline{n}(r) = \sup \tilde{n}_r \quad \text{pour tout } r \in [0, 1]$$

satisfont les conditions de la définition 2.7. Ce lien apparaît graphiquement sur la figure 4 où $x_1 = \underline{n}(1/2)$ et $x_2 = \overline{n}(1/2)$ pour un nombre flou triangulaire $\tilde{3} = (3, 2, 3)$.

La représentation graphique des fonctions \underline{n} et \overline{n} est obtenue par une rotation plane du graphe de la fonction d'appartenance suivie d'une symétrie verticale. Cette transformation est illustrée dans la figure 4 et, pour un nombre flou quadratique $\tilde{3} = (3, 2, 3)$, dans la figure 5. Formellement elle est décrite par les formules suivantes.

Proposition 2.9. *Soit $\tilde{n} = (n, \alpha, \beta)$ un élément de $\mathfrak{F}(G, D)$ où G et D sont des fonctions bijectives. La représentation paramétrique $[\underline{n}, \overline{n}]$ de \tilde{n} est donnée par*

$$(3) \quad \begin{cases} \underline{n}(r) = n - \alpha G^{-1}(r) \\ \overline{n}(r) = n + \beta D^{-1}(r) \end{cases} .$$

En particulier, lorsque \tilde{n} est un nombre flou triangulaire, on a :

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

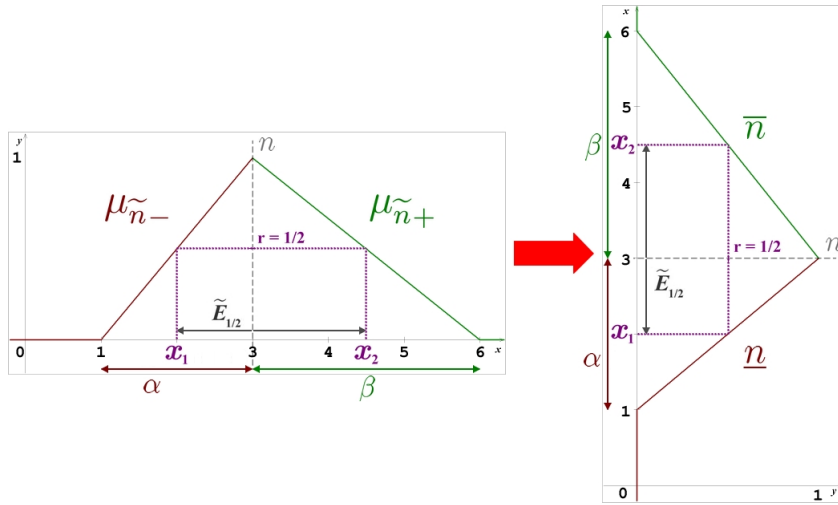


FIGURE 4. Graphe des fonctions $\underline{\tilde{z}}$ et $\overline{\tilde{z}}$ à partir du graphe d'une fonction d'appartenance linéaire

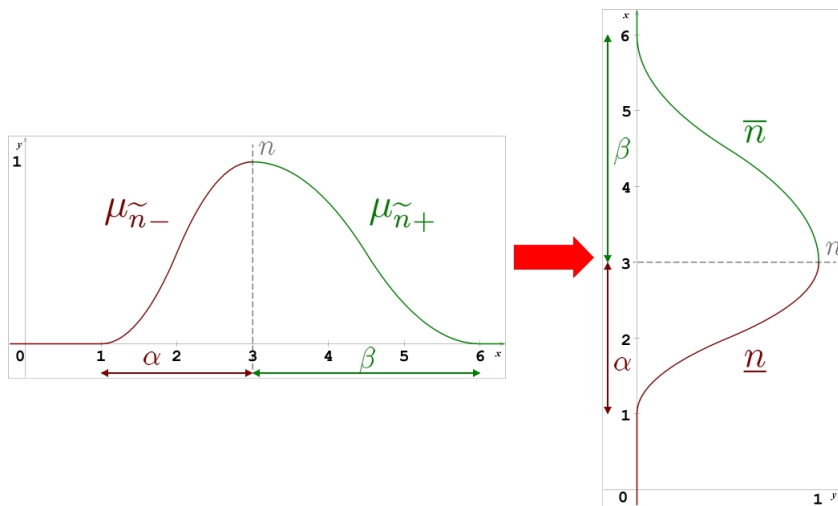


FIGURE 5. Graphe des fonctions $\underline{\tilde{z}}$ et $\overline{\tilde{z}}$ à partir du graphe d'une fonction d'appartenance quadratique

Démonstration. Pour $r \in [0, 1]$, il résulte de la définition 2.5 que $r = G\left(\frac{n-\underline{n}(r)}{\alpha}\right) = D\left(\frac{\overline{n}(r)-n}{\beta}\right)$. Puisque G et D sont inversibles, $\underline{n}(r)$ et $\overline{n}(r)$ satisfont à l'identité (3) de la proposition. Dans le cas triangulaire, on obtient la formule (4) car $G = D = F$ où $F : x \mapsto 1 - x$ est bijective avec $F^{-1} = F$ □

Dans la suite de l'article, les coefficients flous d'une équation ou d'un système seront toujours supposés tous appartenir à une même famille $\mathfrak{F}(G, D)$.

3. LA TRANSFORMÉE RÉELLE D'UNE ÉQUATION

Nous allons utiliser la terminologie suivante : une variable est dite *réelle* si elle représente n'importe quel nombre réel ; une variable réelle est dite *positive* si elle représente n'importe quel nombre réel positif, c'est-à-dire appartenant à \mathbb{R}^+ ; un k -uplet (b_1, \dots, b_k) de variables réelles ou de nombres réels est dit positif si chaque composante b_i est positive.

Dans cette section nous considérons une équation algébrique (E) ci-dessous, à coefficients flous et k variables réelles. Lorsqu'on calcule avec des variables réelles et des nombres flous, le problème est de traiter les nombres flous de la forme $q \cdot \tilde{n}$ car leurs dispersions dépendent du signe de $q \in \mathbb{R}$ (voir lemme 2.8). Dans l'équation (E) chaque monôme $m = x_1^{d_1} \cdots x_k^{d_k}$ ($d_i \in \mathbb{N}$) dont le signe est a priori inconnu, joue le rôle de q ; On notera que si chaque exposant d_i est pair alors m est positif. Quand le k -uplet $\mathbf{x} = (x_1, \dots, x_k)$ est positif, le monôme m est forcément positif.

Pour contourner ce problème de signe, la section 3.2 cherche à obtenir les solutions réelles de (E) dans \mathbb{R}^k à partir des solutions *positives* d'équations auxiliaires floues E(I) où $I \in \{-1, 1\}^k$. Nous appellerons ces équations les équations *induites* de (E).

En considérant seulement des variables réelles positives, dans la section 3.3 une *forme tranchée* de (E) est construite afin d'obtenir une *forme tranchée collectée* de (E) ; c'est-à-dire un système d'équations polynomiales à coefficients réels dont les solutions positive sont celles de (E). Dans la littérature cette forme tranchée collectée est obtenue par un algorithme valable seulement avec des coefficients flous triangulaires. Elle est composée de quatre équations. Dans cet article, la seule condition posée sur les coefficients est qu'ils soient à support fini et à fonctions de dispersion bijective.

De plus, la section 3.4 établit une formule qui fournit une forme tranchée collectée de (E) particulière, formée de seulement trois équations. Nous appelons ce système *la transformée réelle* de (E), désignée par $\mathcal{F}(E)$.

Pour obtenir les solutions réelles de (E), il est nécessaire et suffisant de regrouper les solutions réelles positives des 2^k transformées réelles $\mathcal{F}(E(I))$. Cette question est l'objet de la section 4.

La section 3.5 compare la transformée réelle $\mathcal{F}(E)$ au système tranché collecté qu'on trouve dans les travaux antérieurs pour le cas de coefficients flous triangulaires. Enfin la section 3.6 étend nos résultats aux nombres flous trapézoïdaux.

3.1. Notations et données.

Soit $\mathbf{d} = (d_1, \dots, d_k) \in \mathbb{N}^k$ et k indéterminées x_1, \dots, x_k notées \mathbf{x} . Un *monôme* en x_1, \dots, x_k de multidegré \mathbf{d} est un produit de la forme $x_1^{d_1} \cdots x_k^{d_k}$ qu'on notera $\mathbf{x}^{\mathbf{d}}$. De la même manière, pour $\mathbf{a} = (a_1, \dots, a_k) \in \mathbb{R}^k$ on note $\mathbf{a}^{\mathbf{d}}$ le produit $a_1^{d_1} \cdots a_k^{d_k}$. On note $\mathbf{x} \times \mathbf{y}$ le produit composante par composante $(x_1 y_1, \dots, x_k y_k)$ où $\mathbf{y} = (y_1, \dots, y_k)$. Remarquons que $(\mathbf{x} \times \mathbf{y})^{\mathbf{d}} = \mathbf{x}^{\mathbf{d}} \mathbf{y}^{\mathbf{d}}$.

Dans toute cette section, nous considérons l'équation polynomiale (E) ci-dessous, à coefficients $\tilde{n}_{\mathbf{d}}$ et \tilde{m} dans $\mathfrak{F}(G, D)$, où G et D sont des fonctions bijectives. Ces nombres sont donnés par leur représentation en tuple $\tilde{n}_{\mathbf{d}} = (n_{\mathbf{d}}, \alpha_{\mathbf{d}}, \beta_{\mathbf{d}})$ et $\tilde{m} = (m, \alpha, \beta)$. Soit

$$(5) \quad (E) : \sum_{\mathbf{d} \in \text{Supp}(E)} \tilde{n}_{\mathbf{d}} \mathbf{x}^{\mathbf{d}} = \tilde{m} \quad ,$$

où $\text{Supp}(E)$ désigne le support de (E) i.e. l'ensemble fini des \mathbf{d} dans \mathbb{N}^k tels que $\tilde{n}_{\mathbf{d}} \neq 0$ (nous ne prenons pas en compte le côté droit de (E) qui peut être indifféremment nul ou non).

L'ensemble $\text{Sol}(E)$ des solutions réelles de cette équation est donné par :

$$\text{Sol}(E) = \{ \mathbf{a} \in \mathbb{R}^k \mid \sum_{\mathbf{d} \in \text{Supp}(E)} \tilde{n}_{\mathbf{d}} \mathbf{a}^{\mathbf{d}} = \tilde{m} \} .$$

L'ensemble $\text{Sol}^+(E)$ des solutions dites *positives* de (E) est donné par :

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

Nous cherchons toutes les solutions notre équation floue (E) en passant par les r -coupes afin de nous ramener à des coefficients réels et utiliser les méthodes connues de résolutions de systèmes algébriques. Cependant, d'après l'identité (3) du lemme 2.8, la r -coupe de $\tilde{n}_{\mathbf{d}} x^{\mathbf{d}}$ dépend du signe de $x^{\mathbf{d}}$ alors que ce signe est tout autant inconnu que x . C'est pourquoi nous nous ramenons au cas où toutes les solutions sont positives. Mais nous voulons toutes les racines réelles de (E), pas seulement les positives.

Dans la section 3.2, nous montrons comment construire toutes les solutions de (E) à partir des solutions positives des équations floues induites de (E), que nous noterons $E(I)$ où I parcourt $\{-1, 1\}^k$. Il restera ensuite à voir comment déterminer les 2^k ensembles $\text{Sol}^+(E(I))$. Dans cette perspective, les sections 3.3 et 3.4 supposent que les variables sont toutes réelles et positives. Dans ce cadre, nous cherchons la formule de la transformée réelle $\mathcal{F}(E)$ de (E). Les solutions positives du système polynomial réel $\mathcal{F}(E)$ sont exactement les solutions positives de (E). Dans ces deux parties, l'équation floue (E) jouera le rôle de chacune des équations induites $E(I)$.

3.2. Solutions de (E) en fonction de solutions positives de ses équations induites.

Soit $\mathbf{a} = (a_1, \dots, a_k) \in \mathbb{R}^k$. Nous posons $|\mathbf{a}| = (|a_1|, \dots, |a_k|)$ où $|b|$ désigne la valeur absolue de $b \in \mathbb{R}$. Remarquons que $|\mathbf{a}^{\mathbf{d}}| = |\mathbf{a}|^{\mathbf{d}}$. On adopte la notation $\varepsilon(\mathbf{a}) = (\text{sign}(a_1), \dots, \text{sign}(a_k)) \in \{-1, 1\}^k$ pour le k -uplet des signes des composantes de \mathbf{a} . La valeur $\varepsilon(\mathbf{a})^{\mathbf{d}} = \prod_{i=1}^k \text{sign}(a_i)^{d_i} \in \{-1, 1\}$ est le signe du réel $\mathbf{a}^{\mathbf{d}}$ et on a

$$\mathbf{a}^{\mathbf{d}} = \varepsilon(\mathbf{a})^{\mathbf{d}} |\mathbf{a}^{\mathbf{d}}| = \varepsilon(\mathbf{a})^{\mathbf{d}} |\mathbf{a}|^{\mathbf{d}} .$$

Evaluer en \mathbf{a} le terme $\tilde{n}_{\mathbf{d}} x^{\mathbf{d}}$ de notre équation (E) s'effectue en remplaçant le monôme $x^{\mathbf{d}}$ par sa valeur $\mathbf{a}^{\mathbf{d}} = \varepsilon(\mathbf{a})^{\mathbf{d}} |\mathbf{a}|^{\mathbf{d}}$. D'après l'identité (3) du lemme 2.8, lorsque le signe $\varepsilon(\mathbf{a})^{\mathbf{d}}$ de $\mathbf{a}^{\mathbf{d}}$ est positif, le nombre flou $\tilde{n}_{\mathbf{d}} \mathbf{a}^{\mathbf{d}}$ a pour r -coupe

$$[\underline{n}_{\mathbf{d}}(r) |\mathbf{a}|^{\mathbf{d}}, \overline{n}_{\mathbf{d}}(r) |\mathbf{a}|^{\mathbf{d}}]$$

et si ce signe est -1 alors sa r -coupe est

$$[-\overline{n}_{\mathbf{d}}(r) |\mathbf{a}|^{\mathbf{d}}, -\underline{n}_{\mathbf{d}}(r) |\mathbf{a}|^{\mathbf{d}}] .$$

Il est par conséquent impossible de transformer $\tilde{n}_{\mathbf{d}} x^{\mathbf{d}}$ en sa représentation paramétrique sans connaître le signe de $x^{\mathbf{d}}$. Notre idée pour contourner ce problème de signe de $x^{\mathbf{d}}$, est d'introduire un k -uplet artificiel $I \in \{-1, 1\}^k$ pour les signes des variables. Le réel $I^{\mathbf{d}}$ peut alors être mélangé avec le coefficient $\tilde{n}_{\mathbf{d}}$ et x peut être supposé positif. Cela conduit aux 2^k équations floues induites $E(I)$ définies ci-dessous.

Définition 3.1. Soit (E) une équation polynomiale floue à k variables donnée par (5). Les équations induites de (E) sont les équations suivantes :

$$(6) \quad E(I) : \sum_{d \in \text{Supp}(E)} I^d \widetilde{n}_d x^d = \widetilde{m} \quad \text{où } I \in \{-1, 1\}^k.$$

On notera que (E) = E((1, 1, ..., 1)). Dans E(I), le k -uplet I joue le rôle du k -uplet $\varepsilon(a)$ des signes d'une solution potentielle a de (E). Pour chaque terme $I^d \widetilde{n}_d x^d$ du membre gauche de E(I), la valeur $I^d \in \{-1, 1\}$ joue le rôle du signe et le signe de x^d est quant à lui considéré comme positif. Il n'y a pas de problème pour exprimer la r -coupe de $I^d \widetilde{n}_d x^d$ dans E(I) avec x positive alors que c'est impossible pour $\widetilde{n}_d x^d$ dans (E) avec x non nécessairement positive.

Pour décrire les relations entre les solutions des équations (E) et E(I), pour tout k -uplet de nombres réels $q = (q_1, \dots, q_k)$ et pour toute partie \mathcal{S} de \mathbb{R}^k , nous introduisons la notation suivante :

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

Lemme 3.2. Soit $I \in \{-1, 1\}^k$ et E(I) une équation induite de (E). Alors

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

Démonstration. Comme $I \times I = (1, \dots, 1)$, il suffit de prouver la première égalité. Pour une solution $b \in \mathbb{R}^k$ de E(I), nous avons

$$\widetilde{m} = \sum_d I^d \widetilde{n}_d b^d = \sum_d \widetilde{n}_d (I \times b)^d .$$

Par conséquent $I \times b$ est une solution de (E).

Réciproquement, soit $a \in \text{Sol}(E)$. Alors $b = I \times a$ est une solution de E(I) puisque

$$\widetilde{m} = \sum_d \widetilde{n}_d (I \times I \times a)^d = \sum_d I^d \widetilde{n}_d (I \times a)^d \quad \square$$

Pour tout $a \in \mathbb{R}^k$, il existe $I \in \{-1, 1\}^k$ tel que $\varepsilon(a) = I$ et pour lequel l'équation E(I) évaluée en $x = |a|$ est identique à l'équation (E) évaluée en a . La question de savoir comment retrouver toutes les solutions de l'équation (E) à partir des solutions positives des équations E(I) est résolue par le théorème suivant :

Théorème 3.3. Considérons une équation algébrique floue (E). Alors l'ensemble de ses solutions est formé des $I \times b$ où I parcourt $\{-1, 1\}^k$ et b parcourt l'ensemble des solutions positives de E(I). En d'autres termes,

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

Démonstration. D'après le lemme 3.2 on a $I \otimes \text{Sol}^+(E(I)) \subset \text{Sol}(E)$. Réciproquement, si $a \in \text{Sol}(E)$ alors $a^d = \varepsilon(a)^d |a|^d = \varepsilon(a)^d |a|^d$. Il en résulte que

$$\widetilde{m} = \sum_d a^d \widetilde{n}_d = \sum_d \varepsilon(a)^d |a|^d \widetilde{n}_d .$$

En posant $I = \varepsilon(a)$ et $b = |a| \in \mathbb{R}^{+k}$, nous avons bien $I \times b = \varepsilon(a) \times (\varepsilon(a) \times a) = a$, où b est solution de E(I). \square

Remarque 3.4. Les coefficients de (E) étant donnés par leur représentation en tuple, les équations $E(I)$ se déduisent directement de (E). En effet, pour tout $\mathbf{d} \in \text{Supp}(E)$ le coefficient $I^{\mathbf{d}} \tilde{n}_{\mathbf{d}}$ est égal à $\tilde{n}_{\mathbf{d}}$ lui-même lorsque $I^{\mathbf{d}} = 1$, sinon il est égal à $(-n_{\mathbf{d}}, \beta_{\mathbf{d}}, \alpha_{\mathbf{d}})$ lorsque $I^{\mathbf{d}} = -1$, d'après l'égalité (2). Cela assure que pour trouver toutes les solutions réelles d'une équation, il suffit de disposer de savoir calculer les solutions réelles positives d'une équation quelconque.

Dans la pratique, on n'a pas à résoudre 2^k équations floues distinctes $E(I)$. En particulier, dans le cas où chacune des composantes de tout $\mathbf{d} = (d_1, \dots, d_k) \in \text{Supp}(E)$ est paire, les équations induites $E(I)$ sont toutes identiques à (E).

Exemple 3.5. Prenons, par exemple, $k = 3$ et

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

Il y a $8 = 2^3$ triplets dans $\{-1, 1\}^3$: $I_1 = (1, 1, 1), I_2 = (1, 1, -1), I_3 = (-1, 1, 1), I_4 = (-1, 1, -1), I_5 = (1, -1, 1), I_6 = (1, -1, -1), I_7 = (-1, -1, 1)$ et $I_8 = (-1, -1, -1)$. Il n'y a que deux équations $E(I_j)$ distinctes. En effet, puisque $\text{Supp}(E) = \{\mathbf{d}_1 = (2, 1, 0), \mathbf{d}_2 = (0, 0, 4)\}$, on a $I_j^{\mathbf{d}_2} = 1$ pour tout $j \in \llbracket 1, 8 \rrbracket$ et par conséquent les I_j se séparent en deux groupes : celui des $j \in \llbracket 1, 4 \rrbracket$ pour lesquels $I_j^{\mathbf{d}_1} = 1$ et celui des $j \in \llbracket 5, 8 \rrbracket$ pour lesquels $I_j^{\mathbf{d}_1} = -1$. Ainsi $E(I_1) = E(I_j)$ pour tout $j \in \llbracket 1, 4 \rrbracket$ et $E(I_5) = E(I_j)$ pour tout $j \in \llbracket 5, 8 \rrbracket$. D'après le théorème 3.3, nous avons

$$\text{Sol}(E) = \bigcup_{j=1}^4 I_j \otimes \text{Sol}^+(E(I_1)) \quad \cup \quad \bigcup_{j=5}^8 I_j \otimes \text{Sol}^+(E(I_5)) \quad .$$

Pour $j \in \llbracket 5, 8 \rrbracket$ on a $E(I_j) = E(I_5)$. Cette équation est déterminée à l'aide de la remarque 3.4. Avec $\tilde{3} = (3, \alpha_1, \beta_1), \tilde{1} = (1, \alpha_2, \beta_2)$ et $\tilde{6} = (6, \alpha, \beta)$, on a $-\tilde{3} = (-3, \beta_1, \alpha_1)$ d'où

$$E(I_5) : (-3, \beta_1, \alpha_1) x_1^2 x_2 + (1, \alpha_2, \beta_2) x_3^4 = (6, \alpha, \beta) \quad .$$

Les sections suivantes sont consacrées à la recherche des solutions positives de (E) que nous appliquerons à chaque équation $E(I)$ en tenant compte de la remarque 3.4 illustrée dans l'exemple précédent.

3.3. La forme tranchée de (E) pour déterminer $\text{Sol}^+(E)$.

La résolution algébrique de l'équation floue (E) se base sur le passage de la représentation $G-D$ des nombres flous à leur représentation paramétrique. Dans la présentation ci-dessous nous renforçons significativement cette méthode classique dans le cas de coefficients flous triangulaires en l'appliquant à un système générique et en l'étendant à des coefficients flous bien plus généraux.

Conformément au lemme 2.8, l'égalité de l'équation (E) est équivalente à deux égalités sur les r -coupes des membres de gauche et de droite de (E) si tous les $x^{\mathbf{d}}, \mathbf{d} \in \text{Supp}(E)$, sont supposés représenter des réels de même signe. En effet, d'après la règle (3) de ce lemme, la multiplication d'un nombre flou par un scalaire q se sépare en deux cas : $q \leq 0$ et $q \geq 0$. C'est pour cette raison que nous nous plaçons dans le cas où tous les scalaires sont de même signe; Choisissons ce signe positif (que nous aurions pu choisir négatif). Ainsi, nous cherchons uniquement les solutions $\mathbf{a} \in \mathbb{R}^{+k}$ puisqu'alors pour tout $\mathbf{d} \in \mathbb{N}^k$, le réel $q := \mathbf{a}^{\mathbf{d}}$ est positif. Nous appliquerons les résultats obtenus à chaque équation floue induite $E(I)$.

Pour $\mathbf{a} \in \mathbb{R}^{+k}$, d'après le lemme 2.8, nous avons l'équivalence suivante :

$$(7) \quad \mathbf{a} \in \text{Sol}^+(E) \iff \left[\sum_{d \in \text{Supp}(E)} \frac{n_d(r)}{\overline{n_d(r)}} \mathbf{a}^d, \sum_{d \in \text{Supp}(E)} \overline{n_d(r)} \mathbf{a}^d \right] = [\underline{m}(r), \overline{m}(r)] \quad .$$

Cette équivalence, nous amène à considérer le système $\mathcal{C}(E)$ suivant de deux équations à $k+1$ variables x_1, \dots, x_k, r , appelé appelé *forme tranchée* de (E) (*crisp form* en anglais) :

$$\mathcal{C}(E) : \begin{cases} \sum_{d \in \text{Supp}(E)} \frac{n_d(r)}{\overline{n_d(r)}} \mathbf{x}^d = \underline{m}(r) \\ \sum_{d \in \text{Supp}(E)} \overline{n_d(r)} \mathbf{x}^d = \overline{m}(r) \quad . \end{cases}$$

Soit F un ensemble d'équations de $\mathbb{R}[x_1, \dots, x_k, r]$. Nous posons

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

où $\text{Sol}(F)$ est l'ensemble des solutions de F dans \mathbb{R}^{k+1} . Soit $\mathbf{a} = (a_1, \dots, a_k) \in \mathbb{R}^{+k}$. D'après l'équivalence (7), \mathbf{a} est une solution de (E) si et seulement si, pour tout $r \in [0, 1]$, le $(k+1)$ -uplet (a_1, \dots, a_k, r) est une solution du système $\mathcal{C}(E)$. En d'autres termes, les solutions positives de l'équation floue (E) sont aussi celles de sa forme tranchée $\mathcal{C}(E)$:

$$(8) \quad \text{Sol}^+(E) = \text{Sol}^+(\mathcal{C}(E)) \quad .$$

Exemple 3.6. Reprenons l'équation floue de l'exemple 3.5 avec des nombres flous triangulaires. Nous rappelons que leurs r -coupes sont données par la formule (4) et que les triplets du support commun à $E(I_1) = (E)$ et $E(I_5)$ sont $\mathbf{d}_1 = (2, 1, 0)$ et $\mathbf{d}_2 = (0, 0, 4)$. Les représentations paramétriques des coefficients de l'équation s'écrivent alors comme suit :

- Pour à la fois $E(I_1)$ et $E(I_5)$, nous avons $\widetilde{n_{\mathbf{d}_2}} = \widetilde{1} = [\alpha_2 r + 1 - \alpha_2, -\beta_2 r + 1 - \beta_2]$ et $\widetilde{6} = [\alpha r + 6 - \alpha, -\beta r + 6 + \beta]$
- pour $E(I_1)$, nous avons $\widetilde{n_{\mathbf{d}_1}} = \widetilde{3} = [\alpha_1 r + 3 - \alpha_1, -\beta_1 r + 3 - \beta_1]$
- pour $E(I_5)$, nous avons $\widetilde{n_{\mathbf{d}_1}} = -\widetilde{3} = [\beta_1 r - 3 - \beta_1, -\alpha_1 r - 3 - \alpha_1]$.

Les formes tranchées respectives des deux équations floues distinctes induites de (E) nécessaires et suffisantes au calcul des solutions de (E) sont donc

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

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

Dans le cas spécifique de coefficients triangulaires, illustré dans l'exemple 3.6, la forme tranchée est composée de deux équations avec, dans chaque membre, des expressions linéaires par rapport à la variable r . C'est une conséquence des formules (4). Le cas triangulaire est aisé à traiter car les fonctions de dispersion sont égales à la fonction $F : x \mapsto 1 - x$ qui vérifie $F^{-1} = F$. Le cas général où les fonctions de dispersions sont simplement bijective, nécessite l'utilisation des formules d'inversion (3) avec deux indéterminées au lieu d'une seule. Ceci conduit à la forme tranchée à deux paramètres du théorème suivant :

Théorème 3.7. Soit L et R deux fonctions de dispersion et

$$(E) : \sum_{d \in \text{Supp}(E)} \widetilde{n}_d x^d = \widetilde{m},$$

une équation polynomiale à coefficients dans la famille $\mathfrak{F}(G, D)$ et donnés par leurs représentations en tuple comme suit : $\widetilde{m} = (m, \alpha, \beta)$ et $\widetilde{n}_d = (n_d, \alpha_d, \beta_d)$ pour $d \in \text{Supp}(E)$. Si les fonctions de dispersion G et D sont bijectives alors la forme tranchée de (E) est donnée par :

$$(9) \quad \mathcal{C}(E) : \begin{cases} \sum_d n_d x^d - m + (\alpha - \sum_d \alpha_d x^d) u & = 0 \\ \sum_d n_d x^d - m + (-\beta + \sum_d \beta_d x^d) v & = 0, \end{cases}$$

où $u = L^{-1}(r)$ et $v = R^{-1}(r)$ pour tout $r \in [0, 1]$. Pour $\mathbf{a} \in \mathbb{R}^{+k}$, on a $\mathbf{a} \in \text{Sol}^+(E)$ si et seulement si, pour tout $r \in [0, 1]$, le système (9) est satisfait par le $(k+2)$ -uplet $(a_1, \dots, a_k, L^{-1}(r), R^{-1}(r))$.

Démonstration. Par définition, une fonction de dispersion H envoie l'intervalle $[0, 1]$ sur lui-même. Si, de plus, H est bijective alors son inverse H^{-1} est continue et décroissante, et vérifie $H^{-1}(1) = 0$ et $H^{-1}(0) = 1$. Supposons que les fonctions de dispersion G et D sont bijectives. Puisque chaque r est dans $[0, 1]$, on peut poser $u = G^{-1}(r)$ et $v = D^{-1}(r)$. Lorsque r parcourt $[0, 1]$ de manière croissante, les paramètres u et v parcourent l'intervalle $[0, 1]$ en décroissant de 1 à 0. D'après les relations (3), la forme paramétrique des coefficients de l'équation est donnée par

$$(10) \quad \begin{aligned} \overline{n}_d(r) &= n_d - \alpha_d u & , & \quad \overline{n}_d(r) = n_d + \beta_d v & \quad \text{pour } d \in \text{Supp}(E) \\ \overline{m}(r) &= m - \alpha u & , & \quad \overline{m}(r) = m + \beta v & . \end{aligned}$$

La forme tranchée $\mathcal{C}(E)$ de (E) présentée dans (8) s'écrit alors comme un système de deux équations à $k+2$ variables x_1, \dots, x_k, u, v , où u et v sont dépendantes entre elles :

$$\mathcal{C}(E) : \begin{cases} \sum_d n_d x^d - \alpha_d u x^d = m - \alpha u \\ \sum_d n_d x^d + \beta_d v x^d = m + \beta v \end{cases} .$$

En regroupant les facteurs respectifs de u et v ainsi que le facteur constant en à la fois u et v , la forme tranchée s'exprime bien sous la forme (9). La dernière affirmation du théorème relative à $\text{Sol}^+(E)$ découle de l'égalité (8). \square

Remarquons que l'application du théorème 3.7 ne nécessite pas de connaître les fonctions inverses des restrictions sur $[0, 1]$ de G et D , mais simplement de savoir que ces restrictions sont bijectives.

Notre approche permet à la fois d'améliorer et de généraliser les méthodes connues jusqu'alors. Par exemple, les résultats de [14] et [7] se restreignent aux seuls nombres flous triangulaires. En effet, la forme tranchée de (E) avec deux paramètres $u = L^{-1}(r)$ and $v = R^{-1}(r)$ donnée dans l'identité (9) est une généralisation de la forme tranchée connue dans le cas triangulaire avec un seul paramètre r appartenant à $[0, 1]$.

Dans les articles sus-cités, pour chaque problème à résoudre, l'algorithme calcule le système $\mathcal{C}(E)$ en les variables x_1, \dots, x_k, r , et linéaire en r . Il est alors réécrit en un système équivalent de quatre équations algébriques à coefficients réels en x_1, \dots, x_k , appelé *système tranché collecté* de (E).

Dans la partie 3.4, nous montrerons comment obtenir une forme tranchée collectée réduite à trois équations, ceci pour toute famille $\mathfrak{F}(G, D)$ telle que les fonctions de dispersion

$G - D$ sont bijectives sur leur supports respectifs. Ce système est la transformée réelle de (E). De plus, nous en donnerons la formulation explicite à partir de (E).

3.4. La transformée réelle et les solutions réelles positives de (E).

Ci-dessous nous définissons la transformée réelle d'une équation floue (E) et montrons que ses solutions positives sont celles de (E).

Définition 3.8. Soient G et D deux fonctions de dispersion inversibles et

$$(E) : \sum_{d \in \text{Supp}(E)} \tilde{n}_d x^d = \tilde{m},$$

une équation polynomiale à coefficients dans la famille $\mathfrak{F}(G, D)$ donnés par leurs représentations en tuple $\tilde{n}_d = (n_d, \alpha_d, \beta_d)$ et $\tilde{m} = (m, \alpha, \beta)$. La transformée réelle de l'équation floue (E) est le système à coefficients réels $\mathcal{T}(E)$ suivant :

$$(11) \quad \mathcal{T}(E) : \begin{cases} \sum_{d \in \text{Supp}(E)} n_d x^d = m \\ \sum_{d \in \text{Supp}(E)} \alpha_d x^d = \alpha \\ \sum_{d \in \text{Supp}(E)} \beta_d x^d = \beta. \end{cases}$$

Cette définition s'étend naturellement à un système (S) d'équations floues telles que (E). Nous noterons $\mathcal{T}(S)$ sa transformée réelle, i.e. le système composé par la réunion des transformées réelles des équations de (S).

Théorème 3.9. Sous les hypothèses de la définition 3.8, si les fonctions G et D sont bijectives alors l'ensemble des solutions réelles positives de l'équation (E) est égal à celui de sa transformée réelle, autrement dit

$$\text{Sol}^+(E) = \text{Sol}^+(\mathcal{T}(E)) \quad .$$

Démonstration. Comme G et D sont des bijections, nous pouvons appliquer le théorème 3.7. Soit $\mathbf{a} \in \mathbb{R}^{+k}$. D'après ce théorème $\mathbf{a} \in \text{Sol}^+(E)$ si et seulement si pour tout $r \in [0, 1]$ la forme tranchée de (E) donnée par (9) est satisfaite par les $(k+2)$ -uplets $(a_1, \dots, a_k, G^{-1}(r), D^{-1}(r))$.

Avec $r = 1$, nous avons $u = G^{-1}(1) = 0$. Alors tout $\mathbf{a} \in \text{Sol}^+(E)$ satisfait l'équation $\sum_d n_d x^d = m$. Notons que lorsque $r = 1$, nous avons aussi $v = 0$ puisque $D(0) = 1$. Nous retrouvons ainsi la même équation et non une deuxième. C'est pour cette raison que nous aboutirons à trois équations seulement. Puis, en prenant $r = 0$, nous avons $u = G^{-1}(0) = 1$ et $v = D^{-1}(0) = 1$. En remplaçant dans (9) l'expression $\sum_d n_d x^d - m$ par 0 et chaque variable u et v par 1, on en déduit que toute solution positive de (E) est une solution positive de la transformée réelle $\mathcal{T}(E)$.

Réciproquement, considérons la forme tranchée $\mathcal{C}(E)$ comme un système polynomial en les variables x et à coefficients dans l'anneau $\mathbb{R}[u, v]$. Toute solution $(a_1, \dots, a_k) \in \mathbb{R}^k$ de $\mathcal{T}(E)$ est aussi une solution de $\mathcal{C}(E)$ dans \mathbb{R}^k quels que soient les paramètres u et v , donc en particulier dans l'intervalle $[0, 1]$ et reliés par la contrainte $G^{-1}(u) = D^{-1}(v) \in [0, 1]$. Par conséquent, toute solution positive de $\mathcal{T}(E)$ est aussi une solution positive de (E). \square

Le théorème 3.9 assure que trouver les racines réelles positives de (E) revient à trouver les racines réelles positives de sa transformée réelle $\mathcal{T}(E)$. Il est donc inutile de développer

des calculs intermédiaires sur les représentations paramétriques comme le faisaient les méthodes antérieures dans le cas particulier des coefficients flous triangulaires.

A partir de l'expression de la transformée réelle $\mathcal{T}(E)$, nous pouvons déduire les expressions de ses équations induites $E(I)$. En faisant référence à la remarque 3.4, nous obtenons le corollaire ci-dessous qui se passe de démonstration.

Corollary 3.10. *Sous les hypothèses de la définition 3.8, soit $I \in \{-1, 1\}^k$ et $E(I)$ l'équation induite de E définie dans (6). Alors la transformée réelle de $E(I)$ s'écrit comme suit :*

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

De plus, lorsque les fonctions de dispersion G et D sont bijectives, on a $\text{Sol}^+(E(I)) = \text{Sol}^+(\mathcal{T}(E(I)))$.

3.5. Relations avec des méthodes antérieures dans le cas triangulaire.

Considérons un système (S) formé de s équations polynomiales à coefficients flous triangulaires. Dans ce cas particulier, les auteurs de [14] et [7], calculent un système tranché collecté de (S) contenant $4s$ équations algébriques à coefficients réels. de mêmes solutions positives que (E) . Nous nous intéressons dans cette partie aux relations entre ce système tranché collecté à $4s$ équations et le système tranché collecté particulier à $3s$ équations qu'est notre transformée réelle. Les solutions réelles positives de l'un et l'autre sont les mêmes que celles du système initial (S) . C'est le principe de toute forme tranchée collectée de (S) .

Considérons ci-dessous le système F_1 de la section 6 dans [7] :

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

Pour la première équation, la forme tranchée collectée produite par l'algorithme de [7] est le système suivant

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

Pour la seconde équation, c'est le système

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

Appelons T_1 le système formé des huit équations précédentes.

Par ailleurs, en appliquant à F_1 la formule (11) définissant la transformée réelle, nous obtenons $\mathcal{F}(F_1)$, le système de six équations suivant :

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

Un calcul facile montre l'équivalence des systèmes T_1 et $\mathcal{F}(F_1)$, dont l'ensemble des solutions est $\{(x, y) \in \mathbb{R} \mid xy = 1\}$. Ce phénomène d'équivalence entre les deux approches s'explique de façon tout à fait générale comme nous le montrons ci-dessous en considérant le calcul usuel de la forme tranchée collectée par application de l'algorithme de [7] sur l'équation générique (E) de (5) .

Soient respectivement $\tilde{m} = (n, \alpha, \beta)$ et $\tilde{n}_d = (n_d, \alpha_d, \beta_d)$ ($d \in \text{Supp}(E)$) les représentations en tuple des coefficients flous triangulaires du système (E). D'après les relations (4), les r -coupes s'expriment dans ce cas par

$$\begin{aligned} \tilde{n}_d(r) &= [\alpha_d r + n_d - \alpha_d, -\beta_d r + n_d + \beta_d] \text{ pour } d \in \text{Supp}(E), \text{ et} \\ \tilde{m}(r) &= [\alpha r + m - \alpha, -\beta r + m + \beta] . \end{aligned}$$

A la différence de notre traitement du cas général, la transformation ne nécessite pas le recours à $u = G^{-1}(r)$ et $v = D^{-1}(r)$ où les fonctions de dispersion G et D sont bijectives mais non connues. Pour un nombre flou triangulaire, puisque $G = D = L$ où $L(x) = L^{-1}(x) = 1 - x$, les méthodes antérieures remplacent directement dans les équations $G^{-1}(r)$ et $D^{-1}(r)$ par leur expression en la variable r . C'est ainsi qu'elles aboutissent à la forme tranchée de (E) ci-dessous, exprimée sous forme de deux polynômes en la variable r .

$$\mathcal{C}(E) : \begin{cases} (\sum_d \alpha_d x^d - \alpha)r + \sum_d (n_d - \alpha_d) x^d - m + \alpha = 0 \\ (\beta - \sum_d \beta_d x^d)r + \sum_d (n_d + \beta_d) x^d - m - \beta = 0 . \end{cases}$$

Un k -uplet $(x_1, \dots, x_k) \in \mathbb{R}^k$ est une solution de $\mathcal{C}(E)$ pour tout $r \in [0, 1]$ si et seulement si chaque coefficient en la variable r de ces équations indépendantes est nul. La forme tranchée collectée de (E) s'écrit par conséquent

$$\begin{cases} \sum_d \alpha_d x^d &= \alpha \\ \sum_d (n_d - \alpha_d) x^d &= m - \alpha \\ \sum_d \beta_d x^d &= \beta \\ \sum_d (n_d + \beta_d) x^d &= m + \beta . \end{cases}$$

En appliquant cette transformation à chaque équation du système F_1 , on retrouve la forme tranchée collectée T_1 de notre exemple. Pour l'équation générique (E), en injectant la première relation dans la deuxième et en constatant que la dernière équation est la somme des trois autres, la forme tranchée collectée se réduit en la transformée $\mathcal{F}(E)$ en trois équations dont l'expression définie dans (11).

3.6. Extension à d'autres catégories de nombres flous.

Comme annoncé dans la remarque 2.4, les résultats obtenus s'adaptent facilement aux nombres flous trapézoïdaux à support fini. Ceux-ci élargissent la définition 2.3 que nous donnons des nombres flous en autorisant $\mu_{\tilde{n}}^{-1}(\{1\})$ à être un intervalle $[a, b]$. Dans ce cadre, un nombre flou \tilde{n} à support fini est de type G – D si sa fonction d'appartenance $\mu_{\tilde{n}}$ est de la forme suivante :

$$\mu_{\tilde{n}}(x) = \begin{cases} G\left(\frac{a-x}{\alpha}\right) & \text{pour } a - \alpha \leq x < a \text{ si } \alpha \neq 0 \\ 1 & \text{pour } x \in [a, b] \\ D\left(\frac{x-n}{\beta}\right) & \text{pour } b < x \leq b + \beta \text{ si } \beta \neq 0 \\ 0 & \text{pour } x \in] -\infty, a - \alpha[\cup] b + \beta, +\infty[\end{cases} .$$

Le nombre flou \tilde{n} a alors comme représentation en tuple le quadruplet (a, b, α, β) .

L'expression de la représentation paramétrique donnée dans la proposition 2.9 se décline sous la forme suivante pour un nombre flou trapézoïdal de type G – D, où G et D sont des bijections :

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

Lorsque l'équation (E) : $\sum_{d \in \text{Supp}(E)} \tilde{n}_d x^d = \tilde{m}$ est à coefficients flous trapézoïdaux de type G – D, où $\tilde{n}_d = (a_d, b_d, \alpha_d, \beta_d)$ et $\tilde{m} = (a, b, \alpha, \beta)$ sont les représentations en tuple des coefficients, les formes paramétriques données dans la preuve du théorème 3.7 deviennent

$$\begin{aligned} \underline{n}_d(r) &= a_d - \alpha_d u & , & & \overline{n}_d(r) &= b_d + \beta_d v & \text{pour } d \in \text{Supp}(E), \\ \underline{m}(r) &= a - \alpha u & , & & \overline{m}(r) &= b + \beta v . \end{aligned}$$

En appliquant le raisonnement de la section 3.4 (ici aussi $G(1) = D(1) = 0$ et $G(0) = D(0) = 1$), on obtient de la même manière une transformée réelle de (E), mais à quatre équations cette fois-ci :

$$(14) \quad \mathcal{T}(E) : \begin{cases} \sum_d a_d x^d = a \\ \sum_d b_d x^d = b \\ \sum_d \alpha_d x^d = \alpha \\ \sum_d \beta_d x^d = \beta . \end{cases}$$

En conséquence l'exploitation de la transformée réelle pour résoudre des systèmes polynomiaux se transposera directement à des systèmes à coefficients flous trapézoïdaux. Les algorithmes proposés dans cet article seront valables pour ces systèmes aussi. Seule la fonction algorithmique **Transformée Réelle**, introduite dans la section suivante, devra être légèrement adaptée - en appliquant la formule (14) à chaque équation de (S) - pour retourner la transformée réelle $\mathcal{T}(S)$ avec $4s$ équations au lieu de $3s$.

4. RÉOLUTION RÉELLE D'UN SYSTÈME POLYNOMIAL FLOU

La résolution d'un système flou (S) de s équations se déduit directement des résultats ci-dessus pour une seule équation. Suivant la section 3, à partir d'une équation (E) on construit 2^k équations floues $E(I)$ où I parcourt les 2^k k -uplets de $\{-1, 1\}^k$. D'après le théorème 3.3, les solutions de (E) se déduisent des solutions positives de chaque équation floue induite $E(I)$.

Et selon le théorème 3.9, les solutions positives de chaque équation $E(I)$ sont les solutions positives des trois équations algébriques formant sa transformée réelle $\mathcal{T}(E(I))$.

De la même façon, nous allons considérer 2^k transformées réelles $\mathcal{T}(S(I))$ des systèmes induits de (S) , avec 3 s équations, où I parcourt les 2^k k -uplets de $\{-1, 1\}^k$.

Cette section établit le théorème principal 4.1 exprimant les solutions réelles d'un système à partir des solutions positives des 2^k transformées réelles de ses systèmes induits. Un premier algorithme est issu de l'application directe de ce théorème. On s'intéresse ensuite aux moyens de réduire le nombre de branches de calculs en dérivant certains des 2^k ensembles de solutions réelles de (S) des solutions positives d'un système $\mathcal{T}(S(I))$ traité précédemment par l'algorithme. La discussion conduira à l'algorithme optimisé `SolveFuzzySystem` de la section suivante.

4.1. Fondements.

Soient $(E_1), \dots, (E_s)$ les s équations du système (S) . Pour chaque $I \in \{-1, 1\}^k$, nous notons $S(I)$ le système flou induit de (S) formé des s équations floues induites $E_1(I), \dots, E_s(I)$, selon la notation de la section 3.2. Pour chaque $I \in \{-1, 1\}^k$ la transformée réelle $\mathcal{T}(S(I))$ du système induit $S(I)$ est le système formé des 3 s équations provenant des transformées réelles des équations induites $E_1(I), \dots, E_s(I)$. Le théorème principal ci-dessous est une conséquence directe du théorème 3.3 et du corollaire 3.10.

Théorème 4.1. *Soit (S) un système système flou à coefficient dans $\mathfrak{F}(G, D)$. Si les fonctions G et D sont bijectives alors l'ensemble des solutions de (S) est l'union des $I \times \mathbf{b}$ pour tous les k -uplets \mathbf{b} qui sont solutions positives d'une transformée réelle $\mathcal{T}(S(I))$ où I parcourt $\{-1, 1\}^k$. En d'autres termes,*

$$\text{Sol}(S) = \bigcup_{I \in \{-1, 1\}^k} \{I \times \mathbf{b} \mid \mathbf{b} \in \text{Sol}^+(\mathcal{T}(S(I)))\} = \bigcup_{I \in \{-1, 1\}^k} I \otimes \text{Sol}^+(\mathcal{T}(S(I))).$$

Un premier algorithme pour la résolution réelle de systèmes flous, dérive naturellement du théorème 4.1 et des résultats précédents. Nommé BA-SFS, il est présenté ci-dessous. Il prend en entrée un système polynomial dont les coefficients flous sont donnés par une représentation en tuple et sont tous supposés appartenir à une même famille $\mathfrak{F}(G, D)$ à fonctions de dispersion bijectives. Il est basé sur les fonctions suivantes :

- La fonction **Multisigne**(j) est la bijection naturelle entre l'intervalle de 2^k entiers $[[0, 2^k - 1]]$ et les k -uplets de $\{-1, 1\}^k$ qui à $j := \sum_{i=0}^{2^k-1} b_i 2^i$ associe $I = (c_0, \dots, c_{2^k-1})$ avec $c_i = 2b_i - 1$.
- La fonction **SolPos**(SR) retourne les solutions positives du système d'équations algébriques SR. Elle peut être basée sur toute méthode de résolution connue. L'implantation décrite dans la section 6 fait appel à la méthode de décomposition triangulaire de Wu ([19]).
- La fonction **TransforméeRéelle**(S,I) applique la formule (12) pour renvoyer les 3 s équations de la transformée réelle du système flou à s équations $S(I)$. (ce sera 4 s équations si les nombres flous sont trapézoïdaux).

Algorithm 1 Algorithme basique de résolution réelle

Require: S , un système polynomial à coefficients flous
 k , la dimension, i.e. le nombre de variables
Ensure: sol , l'ensemble des solutions réelles de (S)

```

 $sol := \{\}$ 
for  $j := 0$  to  $2^k - 1$  do
   $I := \text{MultiSigne}(j)$ 
   $\text{TRSI} := \text{TransforméeRéelle}(S(I))$ 
   $\text{PR} := \text{SolPos}(\text{TRSI})$ 
   $sol := sol \cup I \otimes \text{PR}$ 
end for

return  $sol$ 

```

4.2. Réduction du nombre de systèmes algébriques à résoudre.

Par extension de la notation employée pour une équation, nous notons $\text{Supp}(S)$ le support de (S) , c'est-à-dire l'ensemble des \mathbf{d} dans \mathbb{N}^k tels que $x^{\mathbf{d}}$ apparait dans (S) avec un coefficient non nul.

Dans la boucle de l'algorithme BA-SFS, le système $S(I)$ peut être identique à celui d'une étape précédente. Dans ce cas, il est redondant de calculer à nouveau PR , l'ensemble des solutions réelles positives de $S(I)$. Nous cherchons ci-dessous à l'éviter.

L'algorithme optimisé `SolveFuzzySystem` qui est proposé dans la section 5 fournit toutes les solutions de (S) en évitant des calculs inutiles. À cette fin, nous cherchons d'abord à identifier, à partir du support de (S) , des systèmes induits $S(I)$ identiques. L'exemple 3.5 portant sur une seule équation montre comment cette situation peut se produire puisque les 8 équations floues induites se réduisent à seulement 2 distinctes. Lorsque, par exemple encore, toutes les composantes de chaque k -uplet \mathbf{d} du support de (S) sont paires, les 2^k systèmes $S(I)$ induits par (S) sont identiques. Notre objectif est d'automatiser la reconnaissance de systèmes induits identiques.

Pour cela, nous construisons ci-dessous la matrice $M(S)$ de signes où les colonnes sont indicées par les 2^k k -uplets I de $\{-1, 1\}^k$ et les lignes par les k -uplets \mathbf{d} de $\text{Supp}(S)$. L'entrée à la ligne \mathbf{d} et à la colonne I vaut $I^{\mathbf{d}} \in \{-1, 1\}$, qui remplace le signe de $x^{\mathbf{d}}$ dans $S(I)$:

$$M(S) := \mathbf{d} \begin{pmatrix} & I \\ & \vdots \\ & \vdots \\ \dots & I^{\mathbf{d}} \end{pmatrix}$$

Fixons deux k -uplets distincts $I_1, I_2 \in \{-1, 1\}^k$ et considérons $C(I_1)$ et $C(I_2)$ les deux colonnes de $M(S)$ indicées respectivement par I_1 et I_2 . Si $C(I_1) = C(I_2)$ alors $S(I_1) = S(I_2)$. En

effet, pour toute équation (E) du système (S), le coefficient $I_1^d \widetilde{n}_d$ de chaque monôme x^d apparaissant à gauche de chaque équation $E(I_1)$ est identique au coefficient $I_2^d \widetilde{n}_d$ dans $E(I_2)$ d'après la définition des équations induites donnée dans (6).

Le nombre des systèmes $S(I)$ distincts parmi les 2^k induits par (S) est donc au plus égal au nombre de colonnes distinctes dans $M(S)$. La matrice des signes permet d'éviter bon nombre de systèmes induits à résoudre. On peut mettre en oeuvre d'autres reconnaissances rapides de systèmes induits identiques. En effet, deux systèmes $S(I_1)$ et $S(I_2)$ peuvent être identiques à permutation près des équations alors que $C(I_1) \neq C(I_2)$. Nous pouvons encore repérer les équations de la forme " $a = 0$ " et " $-a = 0$ " car elles ont les mêmes solutions.

5. ALGORITHME `SolveFuzzySystem`

Dans cette section nous présentons l'algorithme optimisé `SolveFuzzySystem` - SFS en abrégé - pour calculer les solutions réelles d'un système flou polynomial. Les coefficients du système sont des nombres flous donnés par leur représentation en tuple et sont supposés appartenir à une même famille $\mathfrak{F}(G, D)$ dont les fonctions de dispersion sont bijectives. Après la description de l'algorithme séquentiel SFS dans la section 5.1, nous discutons sa parallélisation dans la section 5.2.

5.1. L'algorithme séquentiel `SolveFuzzySystem`.

Comme le premier algorithme BA-SFS, l'algorithme `SolveFuzzySystem` consiste à parcourir itérativement les colonnes de la matrice de signes $M(S)$ décrite à la section 4.2. Pour chaque colonne $C(I)$ indiquée par un k -uplet I de signes, il cherche les solutions réelles du système (S) issues de $S(I)$ en évitant des calculs inutiles si l'une des colonnes précédentes le permet.

A cet effet, nous employons les trois vecteurs suivants, indicés de 0 à $2^k - 1$, vides au début de l'algorithme :

- *ColonnesDistinctes* contiendra les colonnes distinctes de la matrice des signes; L'indice *cpt* est incrémenté lorsqu'une colonne distincte des précédentes est détectée;
- *SystemesDistincts* contiendra les systèmes flous distincts dont les solutions positives ont été calculées; L'indexation correspond à celle de *ColonnesDistinctes*;
- *lb*, indexé aussi comme *ColonnesDistinctes*, contiendra les solutions réelles positives de $S(I)$ telles que la colonne $C(I)$ apparait dans *ColonnesDistinctes*.

Comme expliqué dans la section 4.2, deux cas de figure permettent d'éviter des calculs redondants pour une colonne courante I :

- (1) Si $C(I)$ est identique à une colonne précédente $C(J)$ dans $M(S)$ alors $S(I) = S(J)$. Dans ce cas, l'algorithme n'ajoute pas la colonne $C(I)$ dans le vecteur *ColonnesDistinctes*. Il utilise les solutions positives de $S(J)$ qui ont déjà été calculées et conservées dans *lb* lors d'une étape précédente. Cela constitue la première étape de l'algorithme (voir l'instruction **4** :).
- (2) Si $S(I)$ est identique à un système $S(J)$ où $C(J)$ est une colonne précédant $C(I)$ tout en lui étant distincte, alors l'algorithme passe à l'indice suivant. Il ajoute dans le vecteur *ColonnesDistinctes* la colonne $C(I)$ et dans le vecteur *lb* les solutions positives précédemment calculées de $S(J)$. Comme il n'est pas adéquat d'ajouter $S(I)$ au vecteur

SystemesDistincts, l'algorithme laisse l'entrée vide à l'indice *cpt*. Nous pourrons alors appliquer les tests (1) et (2) aux colonnes suivantes de $M(S)$ (voir l'instruction **7** :).

En dehors des situations (1) et (2), les solutions positives du système flou $S(I)$ sont calculées, comme dans l'algorithme de base BA-SFS, avec les fonctions **TransforméeRéelle** et **SolPos**. L'algorithme incrémente l'index *cpt* et stocke dans les vecteurs *ColonnesDistinctes*, *SystemesDistincts* et *lb* respectivement $C(I)$, $S(I)$ et les solutions positives de $S(I)$. Il sera ainsi possible d'appliquer les tests (1) et (2) aux colonnes suivantes de $M(S)$ (voir les instructions **9** :).

À chaque étape de la boucle, c'est-à-dire pour chaque k -uplet I de signes, le théorème 4.1 est appliqué pour obtenir les solutions réelles de (S) associées aux solutions positives de $S(I)$.

L'algorithme `SolveFuzzySystem` fait appel aux mêmes fonctions que l'algorithme BA-SFS, complétées de celles ci-dessous :

- La fonction **ColonneSignes**(j, S) renvoie la $(j + 1)$ -ième colonne de la matrice $M(S)$ des signes (la valeur de j débute avec 0 et pas 1);
- La fonction **EstDans**($e, Distinct$) retourne -1 si e n'est pas un élément du vecteur *Distinct*, sinon elle renvoie l'indice de la première occurrence de e dans *Distinct*. Cette fonction est utilisée dans l'algorithme à la fois sur les colonnes signées de $M(S)$ et sur les systèmes de polynômes. Pour une recherche efficace, quand il s'agit de polynômes, nous les ordonnons selon un ordre total sur les monômes et en assignant le signe + au monôme dominant.

5.2. Une version parallèle de l'algorithme `SolveFuzzySystem`,

Les solutions positives de chaque système flou induit $S(I)$ se calculent indépendamment de celles des autres systèmes flous induits (avec successivement les fonctions **TransforméeRéelle** puis **SolPos**). Cette partie étant la plus coûteuse de l'algorithme `SolveFuzzySystem`, il est opportun de la paralléliser. Ceci nécessite de modifier l'algorithme pour d'abord identifier les systèmes induits $S(I)$ distincts à résoudre, mais sans entreprendre leur résolution, tout en conservant les informations permettant d'appliquer ultérieurement le théorème 4.1.

A cette fin, nous modifions le rôle du vecteur *lb*. Il sera indicé de 0 à $2^k - 1$ comme les colonnes de $M(S)$, et $lb[j]$ contiendra l'indice dans *SystemesDistincts* du système correspondant à la colonne **ColonneSignes**(j). Lorsque la résolution d'un système aura été effectuée, ses solutions positives seront stockées dans un nouveau vecteur *SPos*, de mêmes indices que *SystemesDistincts*.

L'algorithme parallèle général se déroule en trois étapes comme suit.

- (1) On repère efficacement les systèmes distincts à résoudre en parallèle. Et, pour chaque colonne I de $M(S)$, i.e. chaque j entre 0 et $2^k - 1$, on enregistre dans $lb[j]$ l'indice du premier système antérieurement traité qui est identique à $\mathcal{F}(S(I))$ (ce peut-être j lui-même si $S(I)$ est nouveau). Les systèmes distincts sont stockés dans le vecteur *SystemesDistincts*.

Cela est réalisé avec l'ossature de l'algorithme séquentiel `SolveFuzzySystem` en y apportant les modifications ci-dessous :

- dans **4** : remplacer l'instruction $sol := sol \cup I \otimes lb[i]$ par $lb[j] := lb[i]$.

Algorithm 2 Algorithme de résolution réelle optimisé `SolveFuzzySystem`, ou SFS

Require: S , un système polynomial à coefficients flous sous leur représentation en tuple k , la dimension, i.e. le nombre de variables
Data: $cpt := -1$, le compteur (à une unité près) du nombre de colonnes distinctes de $M(S)$
 $ColonnesDistinctes := []$, les colonnes distinctes de $M(S)$ déjà parcourues
 $SystemesDistincts := []$, la liste des systèmes $S(I)$ distincts déjà rencontrés
 $lb := []$, $lb[i]$ contiendra les solutions positives de $SystemesDistincts[i]$
 $sol := \{\}$
Ensure: sol , l'ensemble des solutions réelles de (S)

```

# l'indice  $j$  parcourt toutes les colonnes de  $M(S)$ 
# lorsque  $j$  parcourt  $[0, 2^k - 1]$ ,  $MultiSigne(j)$  parcourt les multi-signes  $I \in \{-1, 1\}^k$ 

1: for  $j := 0$  to  $2^k - 1$  do
2:    $I := MultiSigne(j)$ 
    $C := ColonneSignes(j)$  #  $C$  est la colonne courante dans  $M(S)$ 

   # on teste si  $C$  est une nouvelle colonne de  $M(S)$ 
    $i := EstDans(C, ColonnesDistinctes)$ 
3:   if  $i \neq -1$  then
4:     # les solutions positives de  $S(I)$  sont dans  $lb[i]$  : application du théorème 4.1
      $sol := sol \cup I \otimes lb[i]$ 
     # on passe à la colonne suivante dans  $M(S)$  et  $cpt$  n'est pas incrémenté
     go to 1 :
5:   end if

   # ici,  $C$  est une nouvelle colonne de  $M(S)$ 
    $cpt := cpt + 1$ 
    $ColonnesDistinctes[cpt] := C$ 

   # on teste si  $S(I)$  est un nouveau système
    $i := EstDans(S(I), SystemesDistincts)$ 
6:   if  $i \neq -1$  then
7:     # Le  $i$ -ième système est identique à  $S(I)$ . Ses solutions positives rangées dans  $lb[i]$ 
     sont affectées à  $lb[cpt]$  car toute colonne précédant  $C = C(I)$  lui est distincte
      $lb[cpt] := lb[i]$ 
8:   else
9:     # on stocke le nouveau système et on calcule ses solutions positives qu'on stocke
     dans  $lb[cpt]$ 
      $SystemesDistincts[cpt] := S(I)$ 
      $TRSI := TransforméeRéelle(S(I))$ 
      $lb[cpt] := SolPos(TRSI)$ 
10:  end if

   # on applique le théorème 4.1
    $sol := sol \cup I \otimes lb[cpt]$ 

11: end for

12: return  $sol$ 

```

- dans **7** : remplacer l'instruction $lb[cpt] := lb[i]$ par $lb[j] := lb[i]$.
 - dans **9** : remplacer les instructions $TRSI := \mathbf{TransforméeRéelle}(S(I))$ et $lb[cpt] := \mathbf{SPos}(TRSI)$ par $lb[j] := cpt$.
- (2) En parallèle, on résout chacun des systèmes distincts du vecteur *SystemesDistincts*. Pour chaque système $SI := \mathbf{SystemesDistincts}[cpt]$, s'il existe, nous appliquons les deux instructions suivantes :
- ```

TRSI := TransforméeRéelle(SI)
SPos[cpt] := SolPos(TRSI)

```
- Comme dans l'algorithme SFS, quand  $C(I)$  est une nouvelle colonne mais que  $S(I)$  n'est pas un nouveau système, on n'enregistre rien dans *SystemesDistincts[cpt]*.
- (3) On parcourt le vecteur *lb* afin de construire toutes les solutions du système (S) à partir des résolutions de l'étape précédente; il s'agit essentiellement d'appliquer le théorème 4.1. Cette dernière étape est réalisable également en parallèle. Chaque  $lb[j]$  a pour valeur l'indice *cpt* des solutions positives de  $S(I)$  dans le vecteur *SPos* où  $I = \mathbf{MultiSigne}(j)$ . Cette étape se réalise par la boucle suivante :
- ```

for  $j := 0$  to  $2^k - 1$  do
     $sol := \{\mathbf{MultiSigne}(j) \otimes b \mid b \in \mathbf{SPos}[lb[j]]\} \cup sol$ 
end for

```

Nous détaillons l'exécution de la version parallèle dans un exemple de la section 6.2.

6. IMPLANTATION ET EXEMPLES

Nous disposons d'un package Fuzzy sous SageMath ([9]) écrit par Jérémy Marrez ([13]). Une fonction `ResolutionReelleSystemesFlous` implante l'algorithme `SolveFuzzySystem` de la section 5. Rappelons que les nombres flous doivent être à support borné et à fonctions de dispersion inversibles.

6.1. Description des structures et principales fonctions.

La représentation externe des données dans Fuzzy est indiquée ci-dessous :

- une fonction de dispersion H possède une représentation $\text{rep}(H)$; par exemple, si H est quadratique, c'est "Quad";
- un nombre flou $\tilde{n} = (n, a, b)$ de $\mathfrak{F}(G, D)$ est représenté par $\text{rep}(\tilde{n}) = \mathbf{NombreFlouRed}((n, a, b, \text{rep}(G), \text{rep}(D)))$;
- un terme x^d est représenté de façon usuelle par $\text{rep}(x^d) = x_1 ** d_1 \cdots x_k ** d_k$;
- un monôme $M = \tilde{n} x^d$ est représenté par le couple $\text{rep}(M) = (\text{rep}(\tilde{n}), \text{rep}(x^d))$;
- un polynôme p à coefficients flous est représenté par la liste $\text{rep}(p)$ de ses monômes à coefficient non nul; i.e. avec une représentation creuse;
- une équation (E) : $p = q$, où p, q sont deux polynômes, est représentée par le couple $\text{rep}(E) = (\text{rep}(p), \text{rep}(q))$;
- un système (S) formé de s équations $(E_1), \dots, (E_s)$ est représenté par la liste $\text{rep}(S) = [\text{rep}(E_1), \dots, \text{rep}(E_s)]$.

Par exemple, avec des nombres flous quadratiques, le système

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

est représenté comme suit :

```
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)]
```

Remarque 6.1. Les équations floues du système F ci-dessus n'ont pas exactement la même forme que l'équation générique (E) étudiée dans l'article. Le membre droit de (E) ne se limite pas à un nombre flou \tilde{m} . Cependant il est clair que nos résultats s'étendent aux équations de cette forme.

La fonction `resolution_reelle_systemes_flous(S, k)` implantant l'algorithme SFS prend en paramètres la représentation d'un système flou (S) à s équations ainsi que le nombre k de variables. Elle renvoie l'ensemble des k -uplets solutions réelles de(S). Pour cela, elle utilise principalement les fonctions ci-dessous :

- La fonction `transformee_reelle(S, I)`, à partir d'un système flou S à s équations et d'une liste de signes $I \in \{-1, 1\}^k$, construit la transformée réelle du système flou induit $S(I)$, c'est-à-dire le système à $3s$ équations à coefficients réels obtenu en transformant chacune des ses équations $E(I)$ comme au corollaire 3.10. Elle implante donc notre fonction **Transformée Réelle** des algorithmes des sections 4 et 5.
- La fonction `SolPos(Sr)` renvoie l'ensemble des solutions réelles positives d'un système Sr à coefficients réels. Elle fait d'abord appel à la fonction `Wu(Sr)` qui implante l'algorithme de Wu ([8]). Celle-ci renvoie une famille Z d'ensembles de polynômes dits *caractéristiques* telle que la variété $V(Sr)$ des zéros de Sr admet une décomposition en systèmes polynomiaux triangulaires de la forme $V(Sr) = \bigcup_{B \in Z} V(B) \setminus V(I_B)$ où $I_B = \prod_{b \in B} \text{init}(b)$. On peut retrouver la définition de l'initial $\text{init}(b)$ d'un polynôme b dans [8] ou dans [7], où la méthode de Wu est décrite dans le cadre de la résolution de systèmes à coefficients flous triangulaires. A partir de l'ensemble Z calculé, une fonction `get_zeros(Z)` utilisant la structure triangulaire des ensembles caractéristiques B , renvoie les éléments de \mathbb{R}^{+k} qui sont dans $V(Sr)$.

6.2. Exemples.

Exemple 1 : La fonction `resolution_reelle_systemes_flous(F, 2)` retourne comme dans [7] (exemple 6.1) la variété $V(F) = \{(x = -1, y \pm 1)\}$, solution du système F donné plus haut. Nous décrivons ci-dessous les calculs intermédiaires.

Avec $k = 2$ variables x, y , nous avons $2^2 = 4$ multisignes : $I_0 = [-1, -1], I_1 = [-1, 1], I_2 = [1, -1], I_3 = [1, 1]$. En les prenant dans cet ordre qui correspond respectivement à $j = 0, 1, 2, 3$

dans l'algorithme STS et en considérant les monômes du support dans l'ordre $x, 1, y^2$, la matrice des signes $M(F)$ de F est

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

On précise que dans la description ci-dessous de l'exécution de l'algorithme, une colonne sera représentée par un vecteur ligne.

La variable Sr est la transformée réelle du système flou induit $F(I)$ de $s = 2$ équations, avec $I \in \{I_0, \dots, I_3\}$; Sr possède donc en théorie $3s = 6$ équations. Mais en pratique certaines sont identiques. Par exemple, lorsque $I = [-1, -1]$, la transformée réelle de $F(I)$ est le système $[2y^2 - x - 1, -y^2 + 1, -y^2 + 1, -2y^2 - x + 3, -y^2 + 1, -y^2 + 1]$ (nous ne faisons pas apparaître les membres de droite d'une équation lorsqu'ils sont nuls). Celui-ci se réduit à $Sr = [-y^2 + 1, 2y^2 - x - 1, -2y^2 - x + 3]$ en enlevant les doublons. C'est le premier système à résoudre dans le programme. Pour des raisons pratiques évidentes, c'est ce système qui est enregistré dans la variable `SystemesDistincts` et non le système flou $F(I)$ comme indiqué dans l'algorithme SFS.

Présentons le déroulé de la fonction `resolution_reelle_systemes_floos(F, 2)` en liant les paramètres effectifs aux paramètres formels : $S=F$ et $k=2$. Au départ, nous avons :

```
cpt = -1
ColonnesDistinctes = []
SystemesDistincts = []

j=0 :
  I=I_{0}=[-1,-1], C=[-1,1,1]
  La colonne C est nouvelle donc cpt=cpt+1 ; i.e. cpt=0
  ColonnesDistinctes[0] = C
  Sr = transformee_reelle(S,I) retourne Sr= [-y^2+1,2*y^2 -x -1,-2*y^2 -x+3]
  Le système réel Sr est un nouveau système :
  SystemesDistincts[0] = Sr
  SolPos(Sr) retourne set([(1, 1)]), les solutions positives de Sr
  lb[0]= set([(1, 1)])
  On ajoute dans sol le produit de I avec les solutions présentes dans lb[0]:
  sol = set([(-1, -1)])

  ColonnesDistinctes = [[1, -1, 1]]
  SystemesDistincts = [[y^2 - 1, 2*y^2 - x - 1, 2*y^2 + x - 3]]

j=1 :
  I=I_{1}=[-1,1], C=[-1,1,1]
  La colonne C est identique à ColonnesDistinctes[0].
  On ajoute dans sol le produit de I avec les solutions présentes dans lb[0]:
  sol = set([(-1, 1), (-1, -1)])

j=2 :
```

```

I=I_{2}=[1,-1], C=[1,1,1]
La colonne C est nouvelle donc cpt=cpt+1 ; i.e. cpt=1
ColonnesDistinctes[1] = C
Sr = transformee_reelle(S,I) retourne Sr= [2*y^2+x-1, -2*y^2+x+3, -y^2+1]
Le système réel Sr est un nouveau système :
SystemesDistincts[1] = Sr
SolPos(Sr) retourne set([]), les solutions positives de Sr
lb[1]= set([])
On ajoute dans sol le produit de I avec les solutions présentes dans lb[1]:
sol = set([(-1, 1), (-1, -1)])

ColonnesDistinctes = [[-1, 1, 1], [1, 1, 1]]
SystemesDistincts = [[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]
La colonne C est identique à ColonnesDistinctes[1].
On ajoute dans sol le produit de I avec les solutions présentes dans lb[1]:
sol = set([(-1, 1), (-1, -1)])

```

Nous trouvons donc bien la variété $V(F) = \{(x = -1, y \pm 1)\}$.

Exemple 2 : Illustration de la version parallèle de l'algorithme SFS sur le système flou F_1 suivant :

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

La solution retournée par la fonction `resolution_reelle_systemes_floos(F1, 2)` est, comme dans [7], la variété $V(F_1) = \{(x = \frac{1}{y}, y) \mid y \in \mathbb{R} \setminus \{0\}\}$. Nous remarquons que dans la première équation (E) les dispersions à gauche et à droite sont identiques. Ainsi les deux équations $\sum_{d \in \text{Supp}(E)} \alpha_d x^d = \alpha$ et $\sum_{d \in \text{Supp}(E)} \beta_d x^d = \beta$ de la transformée réelle de (E) sont identiques et égales à $x^3y^3 + x^2y^2 + xy = 3$. On retrouve encore cette équation avec les dispersions droites de la seconde équation de F_1 . C'est pour cette raison que nous aurons 4 et non $6 = 3s$ équations dans la transformée réelle de toute équation floue induite $F_1(I)$. Lors d'une implantation, il est possible d'en tenir compte afin de repérer les équations identiques une fois pour toutes sur le système flou initial et d'utiliser le résultat ensuite sur chacun de ses systèmes induits.

Nous allons trouver $V(F_1)$ en appliquant la version parallèle de notre algorithme. Les multisignes sont les mêmes que pour le système F du premier exemple et le support de F_1 est $\{x^3y^3, x^2y^2, xy, 1\}$. La matrice des signes de F_1 est la suivante :

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

Nous voyons qu'il suffit de résoudre seulement les systèmes correspondant aux deux premières colonnes; c'est-à-dire que nous avons les identités suivantes sur le systèmes flous induits : $F_1(I_0) = F_1(I_3)$ and $F_1(I_1) = F_1(I_2)$. Dans l'exécution de l'algorithme cette information se retrouve dans le vecteur $1b$. Ainsi, le nombre de systèmes à résoudre est réduit de moitié.

Lors de la première étape de l'algorithme parallèle, après $j=0$ puis $j=1$, nous avons :

```
ColonnesDistinctes = [[1, 1, 1, 1], [-1, 1, -1, 1]]
SystemesDistincts =
[[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]]
1b = [0,1]
```

Après les tours de boucle $j=2$, puis $j=3$, seul le vecteur $1b$ est modifié comme suit :

$$1b = [0, 1, 1, 0].$$

Puisque $1b[2]=1b[1]$, les solutions réelles positives du troisième système flou induit $F_1(I_2)$ correspondant à $1b[2]$ sont celles du deuxième système $F_1(I_1)$ correspondant à $1b[1]$; ces solutions positives seront dans $SPos[1]$ après la seconde étape. De la même façon on aura $1b[3]=1b[0]$.

Lors de la seconde étape, la fonction $SolPos$ est appelée en parallèle sur chacun des deux systèmes de $SystemesDistincts$ pour calculer leurs solutions positives respectives. Ces solutions sont récupérées et rangées dans le vecteur $SPos$. On a alors :

$$SPos = [\text{set}([(1/y, 'R+')]), \text{set}([])].$$

La dernière étape calcule en parallèle, pour $j=0, 1, 2, 3$, les produits $I \otimes SPos[1b[j]]$ correspondant à la dernière colonne "solutions retournées" ci-dessous. Les quatre résultats sont rangés dans la variable sol contenant les solutions du système flou F_1 . Concrètement, cela donne :

j	I	1b[j]	SPos[j]	solutions retournées
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+')])

$$sol = \text{set}([(1/y, 'R+'), (1/y, 'R-')])$$

Comme avec la version séquentielle, deux résolutions de systèmes seulement sont effectuées, et nous trouvons la variété $V(F_1) = \{(x = \frac{1}{y}, y) \mid y \in \mathbb{R} \setminus \{0\}\}$.

7. CONCLUSION

Pour résoudre un système flou (S) de s équations et k variables, jusqu'à maintenant les méthodes algébriques existantes exécutaient des calculs avec la forme paramétrique des coefficients pour obtenir la forme tranchée collectée de (S) composée de $4s$ équations. Nous montrons que ces calculs sont superflus en exhibant une formule qui définit un système équivalent à $3s$ équations, que nous appelons la transformée réelle $\mathcal{T}(S)$ du système (S). Une propriété fondamentale est qu'elle possède les mêmes solutions réelles positives que (S) (théorème 3.9).

Alors que les méthodes antérieures ne traitaient que les systèmes à coefficients flous triangulaires, nos résultats s'appliquent pour des coefficients dans une famille $\mathfrak{F}(G, D)$ vérifiant simplement que ses fonctions de dispersions G et D sont bijectives. De plus il n'y a pas besoin de calculer l'inverse de ces fonctions puisque la transformée réelle est une formule universelle indépendante de G et D .

Pour résoudre des équations à coefficients flous on est confronté à la question du signe des solutions. Cette question est intrinsèque aux nombres flous et provient du fait que la multiplication d'un nombre flou par un scalaire réel s'exprime différemment selon le signe de ce scalaire. Notre stratégie a consisté à nous concentrer sur les solutions positives en reportant a priori sur les coefficients flous ce problème de signes. Elle a été rendue possible en établissant le théorème 4.1 qui exprime l'ensemble des solutions réelles de (S) en fonction des solutions positives d'au plus 2^k systèmes flous induits de (S). Ce théorème conduit immédiatement à un premier algorithme qui automatise la recherche des solutions de (S) en évitant les études de signes indispensables avec les méthodes antérieures. Notre approche est tout à fait indépendante du choix de la méthode employée pour calculer les solutions positives d'un système de polynômes à coefficients réels.

Parmi les 2^k systèmes induits à résoudre, certains d'entre eux sont identiques. Les différents exemples de l'article montrent qu'il n'est pas rare de réduire substantiellement le nombre de systèmes induits à résoudre. Nous décrivons une stratégie pour éviter des branches de calcul redondantes. Elle conduit à un algorithme optimisé, `SolvingFuzzySystem`, qui a été implanté dans le package `Fuzzy` du logiciel de calcul formel `SageMath`.

La partie coûteuse de cet algorithme réside dans la recherche des solutions positives de chaque transformée réelle d'un système flou induit de (S). Nous décrivons dans la section 5.2 une parallélisation de l'algorithme `SolvingFuzzySystem` permettant d'exécuter ces calculs mutuellement indépendants en parallèle, et nous en illustrons le fonctionnement sur le second exemple de la section 6.

RÉFÉRENCES

- [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*, pages 207–220. Springer, 1990.
- [9] The Sage Developers. *SageMath, the Sage Mathematics Software System*, 2017. <http://www.sagemath.org>.

- [10] D. Dubois, E. Kerre, R. Mesiar, and H. Prade. *Fuzzy Interval Analysis*, pages 483–581. Springer US, Boston, MA, 2000.
- [11] D. Dubois and H. Prade. Operations on fuzzy numbers. *International Journal of Systems Science*, 9(6) :613–626, 1978.
- [12] R. Goetschel and W. Voxman. Elementary fuzzy calculus. *Fuzzy Sets Syst.*, 18(1) :31–43, 1986.
- [13] J. Marrez. Bibliothèque Fuzzy en SageMath, documentation. Preprint HAL-CNRS, HAL-ID hal-01663476. <http://hal.upmc.fr/hal-01663476>, 2017.
- [14] 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.
- [15] 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.
- [16] H. Rouhparvar. Solving fuzzy polynomial equation by ranking method. First Joint Congress on Fuzzy and Intelligent Systems, Ferdowsi University of Mashhad, Iran, 2007.
- [17] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Applicable Algebra in Engineering, Communication and Computing*, 9(5) :433–461, 1999.
- [18] L. Stefanini and L. Sorini. Fuzzy arithmetic with parametric lr fuzzy numbers. In *IFSA/EUSFLAT Conf.*, pages 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@sorbonne-universite.fr

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

Annick Valibouze, LIP6 AND LPSM, SORBONNE UNIVERSITÉ, CNRS, 75252 PARIS CEDEX 05, FRANCE
E-mail address, annick.valibouze@sorbonne-universite.fr