Thick Gradual Intervals: An Alternative Interpretation of Type-2 Fuzzy Intervals and its Potential Use in Type-2 Fuzzy Computations

[(a)] Reda Boukezzoula, [(b)] Luc Jaulin and [(a)] Laurent Foulloy

[(a)] LISTIC - Université Savoie Mont Blanc, 5 Chemin de Bellevue, 74940 Annecy-le-Vieux

[(b)] Lab-STICC, ENSTA-Bretagne, 2 rue François Verny; 29806 Brest

Abstract

This paper proposes a new interpretation of type-2 fuzzy intervals (T2FIs) through the joint use of gradual intervals (GIs) and thick intervals (TIs). In this framework, a T2FI is viewed as a thick gradual interval (TGI). This new representation gives an original concept for the manipulation of T2FIs according to the thick gradual representation. Furthermore, this vision allows an extension of the interval arithmetic arsenal and reasoning to the framework of T2FIs. The proposed approach can be regarded as more computationally viable, which will make T2FIs computations more useful in applied scenarios. As an illustration, the proposed concept is used to implement the elementary arithmetic operations on T2FIs. The potentialities of the TGI approach have been validated in the frameworks of T2FI aggregation operators and T2FI regression.

Keywords: Type-1 and Type-2 Fuzzy Intervals, Thick Intervals and Thick Gradual Intervals, Fuzzy Arithmetic and Gradual Arithmetic, Interval Arithmetic, T2FI Regression, T2FI Aggregation Operators.

I. Introduction

Zadeh proposed fuzzy set theory [74], which provides original mathematical tools for dealing with vague or imprecise information in the form of membership functions. The vagueness is mostly due to the approximate characteristics that are expressed in a linguistic form through a natural language. For clarity, conventional fuzzy subsets (sets) will be referred to as type-1 fuzzy sets (T1FSs). In the fuzzy literature, a T1FS is sometimes called "type-1 fuzzy interval (T1FI)". Indeed, a T1FI is a T1FS such that all its α-cuts are intervals. Furthermore, a T1FI can be considered as a stack of nested intervals that are defined by the concept of α-cuts [4]. The term T1FS, which was initially proposed by Zadeh [74], is used to distinguish these sets from other fuzzy extensions, such as type-2 fuzzy sets (T2FSs), which were initially proposed in [75] and analyzed in, e.g., [39][40][41].

To extend standard interval arithmetic (SIA) [43][44] methods, which were initially proposed over conventional intervals (CIs), to T1FIs, the concept of gradual numbers (GNs) has been proposed [23][26]. A GN is defined by an assignment function that can represent the essence of graduality. In this context, a T1FI can be represented as a pair of GNs [23][26]. These lower and upper bounds are called left and right profiles. This approach provides a new outlook on T1FI representation, arithmetic and reasoning. More generally, according to the GN representation, the concept of a gradual interval (GI) has been proposed [7][8][23][26]. A CI is a GI if its boundaries are GNs. Conversely, a GI can be interpreted as a T1FI if its left and right profiles are injective and, respectively, nondecreasing and nonincreasing. Furthermore, if a T1FI is a particular case of a GI, the inverse is false insofar as no monotony constraint is associated with the GI bounds (the profiles). Therefore, the GI representation encompasses the T1FI representation [7][8][23][26].

To deal with uncertainties in T1FSs, the concept of T2FSs was introduced by Zadeh [74] as an extension of T1FSs. A T2FS is a fuzzy set whose membership grades are T1FSs. They are highly useful in scenarios where it is difficult to determine the “exact” membership function of a T1FS. Furthermore, T2FSs possess many advantages over T1FSs because their membership functions are fuzzy, thereby making it possible to deal with the effects of uncertainty in T1FSs. Due to the computational complexity of T2FSs, most researchers have concentrated on interval T2FSs (IT2FSs). For simplicity and when no confusion is possible, in this paper, normalized IT2FSs are referred to as type-2 fuzzy intervals (T2FIs). Furthermore, according to Mendel *et al.* [40], T2FIs are practical due to their manageable computational complexity.

In the fuzzy literature, T2FSs and T2FIs are applied to numerous application domains, such as image processing and pattern recognition [27][38], multicriteria decision-making and aggregation operators [14][53][54][67][68][73][76], automatic control [10][11][13][55][56] and fuzzy regression [2][36][46][71]. In such applications, the computation over T2FIs is of paramount importance. Typically, fuzzy arithmetic is applied to mathematical models that include T2FIs by using one of the two approaches that have been introduced in the literature: the α-cut approach and the Zadeh extension principle. The main contribution of this paper is the proposal of a new interpretation of T2FIs and their arithmetic through a new kind of interval, namely, a thick gradual interval (TGI). Using this new type of interval, a T2FI is not represented by lower and higher T1FIs but by left and right GIs. Conceptually, the proposed computational method differs from existing methods in the literature and is distinguished by its ability to extend all approaches that are based on CIs and T1FIs to T2FIs in the fields of control, multicriteria decision making, regression and modeling, and aggregation operators, among others.

To examine the TGI strategy, the standard operations $\{+,-,\times,\div\}$ are investigated. According to [45]: "these operations are of course a basis for more complicated problems of interval arithmetic. Therefore, they are very important. If the elementary operations are formulated imprecisely or incorrectly, then using them for solving problems can sometimes lead to controversial results". Moreover, based on the TGI concept, all the SIA operations on CIs and T1FIs [43][44] can be extended to T2FIs. While the primary objective of this paper is to develop the concept of TGIs, potential applications in computing T2FI aggregation operators and in constructing T2FI regression models are proposed to demonstrate the applicability of the proposed concept, namely, we are clarifying that the TGI concept is not merely a theoretical challenge but has real-world applications.

The remainder of this paper is organized as follows: Section II discusses concepts regarding conventional intervals (CIs) and their arithmetic. Thick intervals (TIs) and their arithmetic are detailed in Section III. In Section IV, gradual intervals (GIs) and their associated arithmetic are presented. In Section V, thick gradual intervals (TGIs) are developed, thereby providing a direct extension of SIA operations to T2FIs. Section VI is devoted to a TGI computational example with comparative results according to the α-plane methodology. In Section VII, potential applications of the TGI approach in the fields of T2FI aggregation operators and T2FI regression have been investigated. Finally, concluding remarks are presented in Section VII.

II. RELATED WORKS AND MOTIVATION

In the literature, systems that are based on T2FSs and T2FIs have stimulated a major interest in various application fields (see [10][11][13][14][27][38][55][56][57]). However, fuzzy arithmetic through T2FIs remains a little-studied field. In the fuzzy context, the extension of the typical arithmetic operations on real numbers to T1FIs is not a new problem. It is well known that the computation using the Zadeh's extension principle is computationally expensive. Considering a T1FI as a collection of α–cuts is a conventional approach (e.g., [4][28][48][59]). While the extension principle leads to NP-hard computations, approximation *via* α-cuts (and its hybridizations) provides a feasible method for computing. The literature contains extensive discussion on the mathematical aspects of the two fuzzy arithmetic implementation approaches (see [24] for a review). Due to its simplicity and to the availability of computational methods, fuzzy arithmetic that is based on the α-cut principle is the most common approach for implementing fuzzy arithmetic in various applications. However, the literature is unanimous in the fact that the α-cut approach is time-consuming and depends on the number of α-cuts that are used. Conceptually, the α-cut strategy for computation is simply a generalization of interval arithmetic, where rather than considering CIs at one level only, several levels are considered in [0, 1].

The computational philosophy over T1FIs has been extended to T2FIs. Zadeh [75] was the pioneer who defined operations for T2FIs using α-cuts. Since then, several new representations and computational methods have been proposed (e.g., [32][51][60][61][69][72][73]), such as the α-plane representation [51], the Zslices representation [69] and the conjoint use of α-cuts and the extension principle [30]. For instance, in [72][73], an aggregation operation that uses the α-cut principle is defined. In [29][30][32][51], the concepts of α-cuts and α-planes are applied for various operations. Simply, the α-plane concept is a representation that is comparable to the α-cut concept for T1FSs. In [69], the Zslices principle has been proposed and used to compute the centroids of T2FSs. Based on the α-plane and α-cut concepts, Stefanini *et al.* [60][61] proposed operations on T2FIs according to the F-transform. For additional details on the α-cut and α-plane principles, see [32][51]. In this framework, regardless of which method is used, computing operations on T2FIs remains computationally expensive due to the complex 3D nature of the T2FIs. Although this strategy, namely, the α-cut and α-plane approaches and their associated discretization procedure, was sound and useful in various scenarios, it was computationally expensive and required significant preliminary computations. According to [31], the implementation of arithmetic operations on T2FIs sometimes requires the use of massively parallel processing units such as graphical processing units (GPUs). Moreover, methods such as the α-plane approach underestimate the results and cannot always guarantee rigorous enclosures for the ranges of operations, namely, the T2FIs that are obtained via an operation do not always contain all possible values of the operands.

To provide an overview of the work that is presented here and to explain the reasoning behind our approach, we begin by detailing the motivation for the proposed approach. The developments that are inherent to our methodology will be detailed in the next sections. In this paper, for utilizing T2FIs in an analytically tractable way, a more computationally and analytically viable approach is proposed. Thus, as SIA operations on real numbers have been extended to T1FIs thorough the concept of GIs, our motivation is to extend these operations to T2FIs. To realize this objective, a new alternative T2FI representation is developed. This representation is based on the joint application of the GI [7][8][23][26] and TI [15][20][21] concepts. A TI is an interval for which the bounds are uncertain and are represented by CIs. As a T1FI is regarded as a pair of GNs (left and right profiles), a T2FI can be represented by a pair of left and right GIs. This approach, which utilizes the new concept of thick gradual intervals (TGIs), enables the extension of the mathematical arsenal of interval arithmetic and reasoning using CIs and T1FIs to T2FIs while avoiding the discretization procedure, which is necessary for implementing the α-cut principle and its hybridizations. A T2FI is represented and manipulated as a TGI through the SIA. The TGI concept can be transposed in almost all applications based on T2FIs where guaranteed and analytical computations are possible. For instance, the TGI concept can be used in big-data clustering to model fuzzy uncertainties [57], in fuzzy multi-objective reliability–redundancy allocation problems [1], in the restricted Boltzmann machine (RBM) to model its uncertain parameters [58] and in automatic control applications to represent the inputs, outputs and/or parameters of dynamical systems [10][11][13]. Furthermore, the T2FI aggregation operator methods [53][54][67][68][76] and the T2FI regression approaches [3][5][12][16][25][36][46][71] can find a new breath thanks to the TGI strategy. The proposed strategy does not suffer from any restriction on the class of functions to which it can be applied and provides analytical definitions for efficiently computing functions of T2FIs. However, in this paper, the proposed method is restricted to normal T2FIs but can be generalized through thick gradual sets (see [20] for the definition of thick sets).

Although this approach can take advantage of the flexibility, the rigor and the guaranteed results of interval arithmetic and reasoning, it can be sometimes criticized for its accumulation of fuzziness. This phenomenon causes overestimation of the uncertainties in the resulting T2FIs. This overestimation originates from the decorrelation phenomenon of the SIA, which is also known as the dependency problem. Indeed, because SIA guarantees the containment of the set of all possible results, the pessimistic independence property between the intervals is implicitly assumed. Furthermore, this overestimation problem can be reduced by implementing extensions and hybridizations of SIA, such as the arithmetic Kaucher [47] and Stefanini [59]. These arithmetic's have been applied to T1FIs [7][8][59] and can be naturally extended to T2FIs *via* TGIs.

III. Conventional intervals (CIs) and their arithmetic

II.1. Conventional interval (CI) representation

A real interval $[a]$ is defined as a closed compact and bounded subset of $\Re$ such that:

$$[a] = [a^-, a^+] = \{a \in \Re \mid a^- \le a \le a^+\};\ \text{where } a^- \le a^+ \quad (1)$$

The real numbers $a^- = \inf([a])$ and $a^+ = \sup([a])$ are regarded as the endpoints (the lower and upper bounds) of the interval $[a]$. Throughout this paper, the set $\mathbb{I} = \{[a^-, a^+] \mid a^- \le a^+;\ a^-, a^+ \in \Re\}$ denotes the set of proper intervals and $\mathbb{Z}$ represents the subset of intervals in $\mathbb{I}$ that contain zero in their interiors. In (1), the bracket notation can be regarded as an operator that associates to an unknown and crisp value a an interval domain $[a]$ that contains it. Furthermore, as discussed by Kulpa in [49][50], the real interval $[a]$ can be interpreted as an imprecise or uncertain real number, for which it is possible to guarantee that its precise value is located somewhere between the endpoints a^- and a^+. Usually, the interval notation $[a]$, which is typically used in the interval arithmetic literature, is of an epistemic nature and refers to an arbitrary real number a in the interval $[a]$.

II.2. Conventional interval (CI) arithmetic

CIs are manipulated using the well-known SIA, which was developed by Sunaga, Warmus and Moore [43][44][64][70]. When considering two CIs, namely, $[a] = [a^-, a^+]$ and $[b] = [b^-, b^+]$, the four SIA operations of $+, -\ \times$, and $\div$ are defined by the following expressions:

$$\text{Addition: } \forall [a], [b] \in \mathbb{I}: [a]+[b] = [a^-+b^-, a^++b^+] \quad (2)$$

$$\text{Subtraction: } \forall [a], [b] \in \mathbb{I}: [a]-[b] = [a^- - b^+, a^+ - b^-] \quad (3)$$

$$\text{Multiplication: } \forall [a], [b] \in \mathbb{I}: [a]\times[b] = [min(\phi), max(\phi)];\ \phi = \{a^-b^-, a^-b^+, a^+b^-, a^+b^+\} \quad (4)$$

$$\text{Division: } \forall [a] \in \mathbb{I}, [b] \in \mathbb{I}\backslash\mathbb{Z}: [a]\div[b] = [a]\times(1\div[b]);\ \text{where } 1\div[b] = [1/b^+, 1/b^-] \quad (5)$$

If $0 \in [b]$, it is assumed that $[a]\div[b] = \Re$.

II.3. Example 1: Is the CI representation always sufficient?

The literature is unanimous regarding the utility of and the interest in the CI representation and its associated arithmetic (SIA). This strategy has several advantages and it permits rigorous enclosures for the ranges of operations and functions. In this representation, what is imprecise (or uncertain) is not the interval $[a]$ but the content of the information that is instantiated in $[a]$. Indeed, the interval $[a]$ is "precise" and its bounds, namely, a^- and a^+, are assumed to be known with certainty. However, in some practical applications, the interval bounds can be uncertain [15][20][21] and they cannot be expressed as crisp values.

We present a simple example to illustrate this case. Let us consider a robot that is moving on a one-dimensional path and is at position x. This robot can see for a distance d. The position x is acquired via GPS. Therefore, it is associated with an uncertainty. The robot vision system depends on the weather conditions and provides the distance d with an uncertainty. The visibility zone, which would be a conventional CI $[z] = [x - d, x + d]$ if x and d were precise information, becomes a more complex piece of information since the bounds are themselves CIs. This phenomenon of uncertainty gives rise to the concept of thick intervals (TIs), of which the bounds are CIs. This concept of TIs, which extends the CI representation in an uncertain environment, is detailed in the next section [20].

IV. Thick intervals (TIs) and their arithmetic

III.1. Thick interval (TI) representation

To represent and manipulate intervals whose bounds are uncertain and are represented by CIs, the concept of thick intervals (TIs) has been proposed by Desrochers and Jaulin [20]. A TI, which is denoted $[\![a]\!]$, is a subset of $\mathbb{I}$ and can be expressed in the following form (see Fig. 1):

$$[\![a]\!] = [\![[a^-], [a^+]]\!] = \{[a] = [a^-, a^+] \in \mathbb{I} \mid a^- \in [a^-] \text{ and } b^- \in [b^-]\} \quad (6)$$

In (6), $[a^-]$ and $[a^+]$ are two CIs that contain the uncertain lower bound $a^- \in \Re$ and the uncertain upper bound $a^+ \in \Re$, respectively.

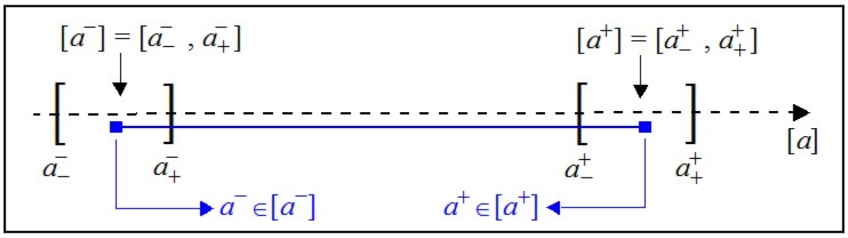


Fig. 1: Representation of a thick interval $\llbracket a \rrbracket$

TI $\llbracket a \rrbracket$ represents all the CIs $[a^-, a^+]$, where $a^- \in [a^-]$ and $b^- \in [b^-]$ (see Fig. 1). A CI is a TI for which the bounds are certain, namely,

$$[a] = [a^-, a^+] = \llbracket [a^-, a^-], [a^+, a^+] \rrbracket$$

III.2. Thick interval (TI) arithmetic

The SIA operations on CIs can be directly extended to TIs [20][21]. For instance, when considering two TIs $\llbracket a \rrbracket = \llbracket [a^-], [a^+] \rrbracket$ and $\llbracket b \rrbracket = \llbracket [b^-], [b^+] \rrbracket$, the thick operations +,– ×, and ÷ can be obtained via an extension of SIA on CIs to TIs [20][21]. These operations are defined by the following expressions:

$$\text{Addition: } \forall \llbracket a \rrbracket, \llbracket b \rrbracket \in \mathbb{I}: \llbracket a \rrbracket + \llbracket b \rrbracket = \llbracket [a^-], [a^+] \rrbracket + \llbracket [b^-], [b^+] \rrbracket = \llbracket [a^-]+[b^-], [a^+]+[b^+] \rrbracket \quad (7)$$

$$\text{Subtraction: } \forall \llbracket a \rrbracket, \llbracket b \rrbracket \in \mathbb{I}: \llbracket a \rrbracket - \llbracket b \rrbracket = \llbracket [a^-], [a^+] \rrbracket - \llbracket [b^-], [b^+] \rrbracket = \llbracket [a^-]-[b^+], [a^+]-[b^-] \rrbracket \quad (8)$$

$$\text{Multiplication: } \forall \llbracket a \rrbracket, \llbracket b \rrbracket \in \mathbb{I}: \llbracket a \rrbracket \times \llbracket b \rrbracket = \llbracket [a^-], [a^+] \rrbracket \times \llbracket [b^-], [b^+] \rrbracket = \llbracket min(\Phi), max(\Phi) \rrbracket \quad (9)$$

where $\Phi = \{[a^-]\times[b^-], [a^-]\times[b^+], [a^+]\times[b^-], [a^+]\times[b^+]\}$ and the *min* and *max* operators between two CIs $[a] = [a^-, a^+]$ and $[b] = [b^-, b^+]$ are expressed as follows:

$$min([a], [b]) = [min(a^-, b^-), min(a^+, b^+)]; \; max([a], [b]) = [max(a^-, b^-), max(a^+, b^+)]$$

Due to the commutative and associative properties of the *min* and *max* operators, in the presence of several CIs, the computational mechanism is applied to pairs of intervals.

$$\text{Division: } \forall \llbracket a \rrbracket, \llbracket b \rrbracket \in \mathbb{I} \backslash \mathbb{Z}: \llbracket a \rrbracket \div \llbracket b \rrbracket = \llbracket [a^-], [a^+] \rrbracket \div \llbracket [b^-], [b^+] \rrbracket = \llbracket a \rrbracket \times (1 \div \llbracket b \rrbracket) \quad (10)$$

$$\text{where: } 1 \div \llbracket b \rrbracket = 1 \div \llbracket [b^-], [b^+] \rrbracket = \llbracket 1 \div [b^+], 1 \div [b^-] \rrbracket$$

In (7)-(10), each operation $\odot \in \{+,-,\times,\div\}$ between TIs is interpreted as follows: if $[a] \in \llbracket a \rrbracket$ and $[b] \in \llbracket b \rrbracket$, then the interval $[x] = [a] \odot [b]$ satisfies $[x] \in \llbracket a \rrbracket \odot \llbracket b \rrbracket$. Furthermore, each operation $[a] \odot [b]$ is performed via conventional SIA operations (2)-(5).

According to the definitions (7)-(10), the arithmetic operations over TIs are direct transpositions of SIA expressions (2)-(5), where the operations on real number bounds are replaced by operations on CI bounds. To show the methodology and the specificities of the computation over TIs, an illustrative computational example is presented in Appendix *A*.

III.3. Example 2: Is the TI representation useful?

Let us reconsider the robot example that was presented in Section II.2. When the position x and the distance d are crisp values, the visibility zone is specified by the CI:

$$[z] = [z^-, z^+] = x + [-d, d] = [x-d, x+d]$$

However, as discussed in Section II.2, the position x and the distance d are assumed to be uncertain and are represented by their likelihood CIs, namely, $x \in [x] = [7, 12]$ and $d \in [d] = [3, 5]$. According to the TI representation, the robot visibility zone can be formulated as follows:

$$\llbracket z \rrbracket = \llbracket [z^-], [z^+] \rrbracket = [x] + \llbracket -[d], [d] \rrbracket = [7, 12] + \llbracket -[3, 5], [3, 5] \rrbracket$$

Expressing the conventional interval [7, 12] as the TI $\llbracket [7, 7], [12, 12] \rrbracket$ yields the following:

$$\llbracket z \rrbracket = \llbracket [7, 7], [12, 12] \rrbracket + \llbracket -[3, 5], [3, 5] \rrbracket = \llbracket [7, 7] - [3, 5], [12, 12] + [3, 5] \rrbracket = \llbracket [2, 4], [15, 17] \rrbracket$$

The concepts of CI and TI have been detailed. The next sections are devoted to their extension to the fuzzy case by introducing a vertical dimension, which is related to the relevance degrees and is limited to the unit interval [0, 1]. From a methodological perspective, a CI is extended to a GI (T1FI) and a TI is extended to a TGI (T2FI).

V. GRADUAL INTERVALS (GIS) VERSUS T1FIS AND THEIR ARITHMETIC

In the paper, for simplicity and without loss of generality, some examples are carried out using unimodal and linear GIs. However, the proposed concepts remain transposable regardless of the shape of the considered GIs.

VI.1. Gradual interval (GI) representation

A CI $[a]$ can be represented by an indicator function that takes the value 1 over the interval and 0 anywhere else. To represent the graduality in CIs and to improve their specificity, the concept of GNs has been proposed [23][26]. In scenarios in which the bounds of a CI represent a gradual transition over this interval, they can be represented by GNs. A GN is a real-valued number that is parameterized by a degree of relevance $\lambda \in (0,1]$. Furthermore, it is modeled by a function from (0,1] to $\Re$. In this framework, a CI $[a]$ becomes a GI $[a(\lambda)]$ if its bounds are GNs [6][7][8][23][26]. Similar to a CI, a GI is represented by the ordered pair of its two bounds, which are called left and right profiles. The GI $[a(\lambda)] = [a^-(\lambda), a^+(\lambda)]$, where $a^-(\lambda) \leq a^+(\lambda)$ and $a^-(\lambda)$ and $a^+(\lambda)$ are GNs. The latter are assumed to be continuous and their domains are extended to [0, 1], namely, $a^-(0)$ and $a^+(0)$ are defined. Via a similar approach, as for the T1FI representation, in the gradual representation two dimensions are considered. The horizontal dimension is similar to that in the CI representation. The vertical dimension is related to the relevance degrees and is limited to the unit interval [0,1]. For example, Fig. 2 shows two GIs, namely, $[a(\lambda)]$ and $[b(\lambda)]$, where no monotony constraint is necessarily imposed on the GNs (the profiles).

GI $[a(\lambda)]$ can be interpreted as a T1FI if its profiles $a^-(\lambda)$ and $a^+(\lambda)$ are injective and respectively nondecreasing and nonincreasing [6][7][8][23][26]. If a T1FI is a particular case of a GI, the reciprocal is false insofar as no monotony constraint is associated with the GNs (the profiles). The concept of GIs is more general and encompasses that of T1FIs. For the remainder of this paper, a GI in which the profiles $a^-(\lambda)$ and $a^+(\lambda)$ are, respectively, nondecreasing and nonincreasing is called a monotone (consonant) GI (or T1FI). A nonmonotone (nonconsonant) GI that cannot be represented by a T1FI is called a "pure GI". Moreover, the set of GIs is denoted by $\mathbb{GI}$. For instance, the GI $[a(\lambda)]$ in Fig. 2 is consonant and can be regarded as a T1FI. However, the GI $[b(\lambda)]$ is a pure GI and cannot be represented by a T1FI. Via a simplified approach, if an FI can be regarded as a stack of nested CIs, a pure GI is characterized by ill-nested CIs through the vertical dimension.

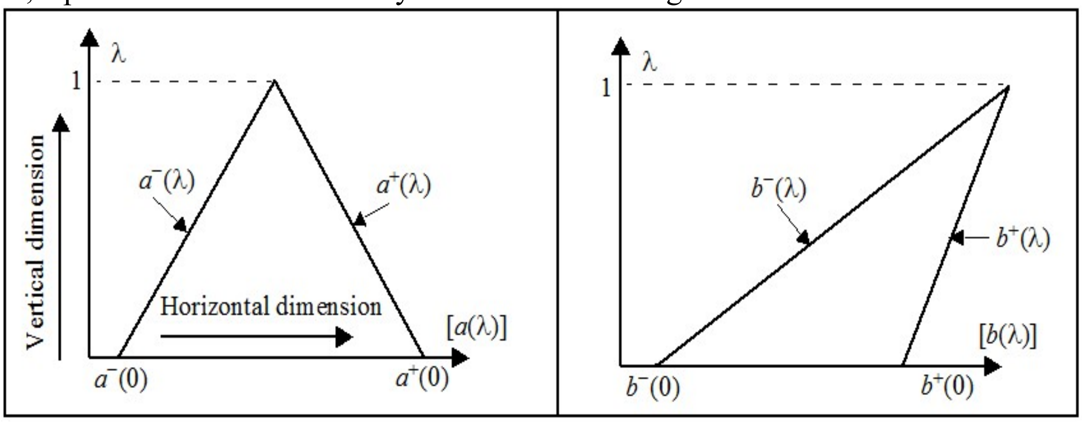


Fig. 2: Two gradual intervals $[a(\lambda)]$ and $[b(\lambda)]$

VI.2. Gradual interval (GI) arithmetic

The SIA operations on CIs can be extended to GIs where the CIs in (2)-(5) are replaced by gradual CIs. Thus, the arithmetic operations over the GIs are defined by the following expressions [7][8]:

$$\text{Addition: } \forall [a(\lambda)], [b(\lambda)] \in \mathbb{GI} : [a(\lambda)]+[b(\lambda)] = [a^-(\lambda)+b^-(\lambda), a^+(\lambda)+b^+(\lambda)] \quad (11)$$

$$\text{Subtraction: } \forall [a(\lambda)], [b(\lambda)] \in \mathbb{GI} : [a(\lambda)]-[b(\lambda)] = [a^-(\lambda)-b^+(\lambda), a^+(\lambda)-b^-(\lambda)] \quad (12)$$

$$\text{Multiplication: } \forall [a(\lambda)], [b(\lambda)] \in \mathbb{GI} : [a(\lambda)]\times[b(\lambda)] = [min(\phi(\lambda), max(\phi(\lambda)] \quad (13)$$

$$\text{where: } \phi(\lambda) = \{a^-(\lambda)b^-(\lambda), a^-(\lambda)b^+(\lambda), a^+(\lambda)b^-(\lambda), a^+(\lambda)b^+(\lambda)\}$$

$$\text{Division: } \forall[a(\lambda)] \in \mathbb{GI},\ [b(\lambda)] \in \mathbb{GI} \setminus \mathbb{Z}: [a(\lambda)] \div [b(\lambda)] = [a(\lambda)] \times (1 \div [b(\lambda)]) \quad (14)$$
$$\text{where } 1 \div [b(\lambda)] = [1/b^{+}(\lambda), 1/b^{-}(\lambda)]$$

As a T1FI is a special case of a GI, the operations on T1FIs can be implemented by using expressions (11)-(14) over GIs.

IV.3. Example 3: Is the GI (or T1FI) representation always sufficient?

We reconsider the robot example that was discussed in Section II.3. Let us assume that the position x and the distance d are still uncertain and are instantiated in their likelihood CIs, to which degrees of confidence are associated. In this case, $x \in [x(\lambda)]$ and $d \in [d(\lambda)]$, where $[x(\lambda)]$ and $[d(\lambda)]$ are distributions of possibility that are given by the T1FIs (monotone GIs): $[x(\lambda)] = [7+3\lambda, 12-2\lambda]$ and $[d(\lambda)] = [3+\lambda, 5-\lambda]$ (see Fig. 3).

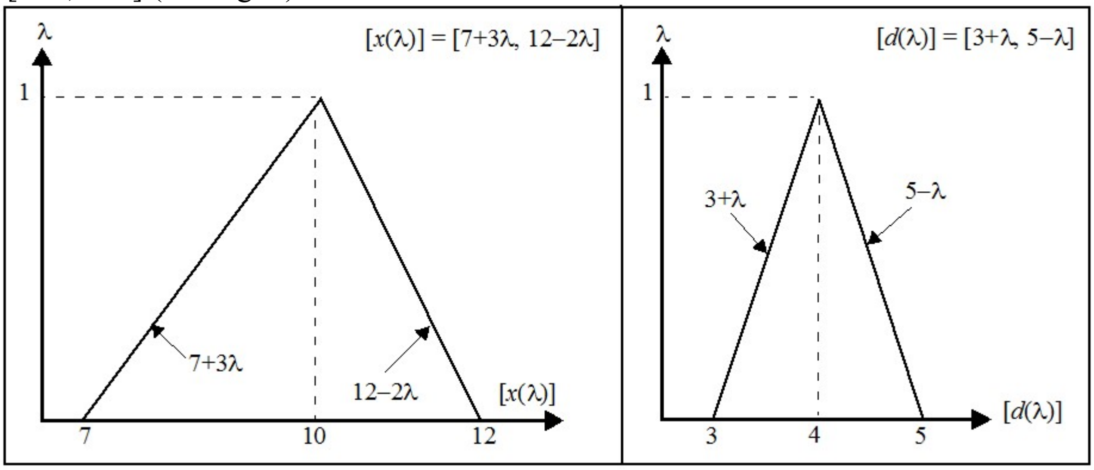


Fig. 3: T1FI representations of $[x(\lambda)]$ and $[d(\lambda)]$

According to this representation, the robot visibility zone when $x \in [x(\lambda)]$ and $d \in [d(\lambda)]$ cannot be correctly expressed as a single GI (T1FI) since its bounds are also GIs. In this case, a new kind of GIs, which can be qualified as thick, can be envisioned, where the bounds are not GNs but GIs. This concept of a TGI, which can represent a T2FI, is detailed in the next section.

VI. Thick gradual intervals (TGIs) versus T2FIs and their arithmetic

V.1. Thick gradual interval (TGI) representation

To represent and manipulate GIs (or T1FIs) whose bounds (profiles) are uncertain, the concept of TGIs is proposed in this paper. This concept is useful in scenarios in which a GI is uncertain and it is difficult to determine its profiles with certainty (see Fig. 4).

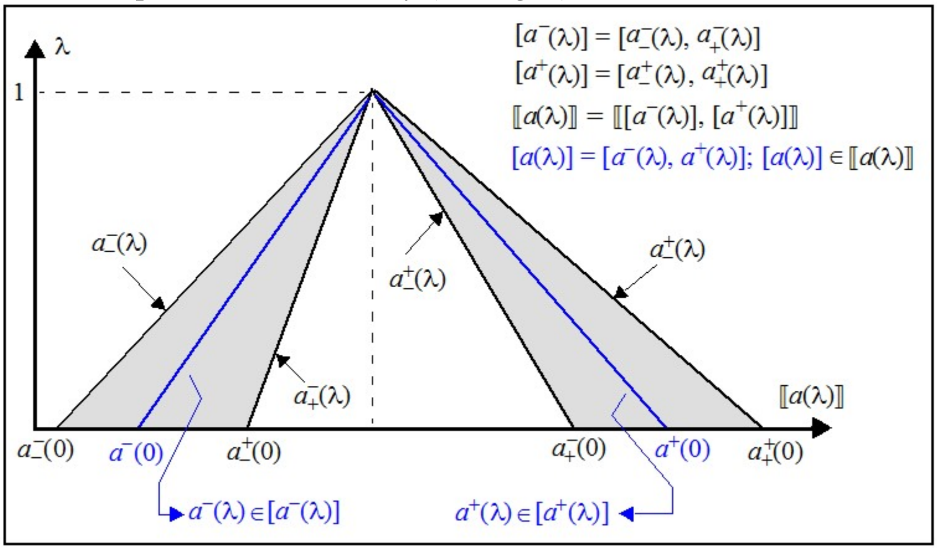


Fig. 4: TGI representation as an uncertain T1FI

In this case, according to the TI formalism, a GI $[a(\lambda)]$ becomes a TGI $[\![a(\lambda)]\!] = [\![[a^-(\lambda)], [a^+(\lambda)]]\!]$, where the bounds $[a^-(\lambda)]$ and $[a^+(\lambda)]$ are GIs. Thus, a TGI is defined by the following expression:

$$[\![a(\lambda)]\!] = [\![[a^-(\lambda)], [a^+(\lambda)]]\!] = \{[a(\lambda)] = [a^-(\lambda), a^+(\lambda)] \in \mathbb{GI} \mid a^-(\lambda) \in [a^-(\lambda)] \text{ and } b^-(\lambda) \in [b^-(\lambda)]\}$$

The TGI $[\![a(\lambda)]\!]$ represents an uncertain GI $[a(\lambda)]$, where its profiles $a^-(\lambda)$ and $a^+(\lambda)$ are located in their likelihood GIs $[a^-(\lambda)]$ and $[a^+(\lambda)]$, respectively (see Fig. 4). By analogy with the GI formalism, the bounds $[a^-(\lambda)]$ and $[a^+(\lambda)]$ are respectively called left and right GIs. This new representation enables a direct extension of the thick SIA operations (7)-(10) to TGIs.

The bounds $[a^-(\lambda)]$ and $[a^+(\lambda)]$ are pure GIs and cannot be represented by T1FIs, thereby giving the TGI concept its full meaning.

V.2. Thick gradual interval (TGI) arithmetic

The elementary operations over TGIs can be implemented via the following expressions:

$$\text{Addition: } \forall [\![a(\lambda)]\!], [\![b(\lambda)]\!] \in \mathbb{GI} : [\![a(\lambda)]\!] + [\![b(\lambda)]\!] = [\![[a^-(\lambda)]+[b^-(\lambda)], [a^+(\lambda)]+[b^+(\lambda)]]\!] \quad (15)$$

$$\text{Subtraction: } \forall [\![a(\lambda)]\!], [\![b(\lambda)]\!] \in \mathbb{GI} : [\![a(\lambda)]\!] - [\![b(\lambda)]\!] = [\![[a^-(\lambda)]-[b^+(\lambda)], [a^+(\lambda)]-[b^-(\lambda)]]\!] \quad (16)$$

$$\text{Multiplication: } \forall [\![a(\lambda)]\!], [\![b(\lambda)]\!] \in \mathbb{GI} : [\![a(\lambda)]\!] \times [\![b(\lambda)]\!] = [\![min(\Phi(\lambda)), max(\Phi(\lambda))]\!] \quad (17)$$

$$\text{where: } \Phi(\lambda) = \{[c_1(\lambda)], [c_2(\lambda)], [c_3(\lambda)], [c_4(\lambda)]\}; \text{ with:}$$

$$[c_1(\lambda)] = [a^-(\lambda)]\times[b^-(\lambda)], [c_2(\lambda)] = [a^-(\lambda)]\times[b^+(\lambda)], [c_3(\lambda)] = [a^+(\lambda)]\times[b^-(\lambda)]; [c_4(\lambda)] = [a^+(\lambda)]\times[b^+(\lambda)]\}$$

As for CIs, the *min* and *max* between two GIs $[a(\lambda)]$ and $[b(\lambda)]$ are expressed as follows:

$$min([a(\lambda)], [b(\lambda)]) = [min(a^-(\lambda), b^-(\lambda)), min(a^+(\lambda), b^+(\lambda))];$$

$$max([a(\lambda)], [b(\lambda)]) = [max(a^-(\lambda), b^-(\lambda)), max(a^+(\lambda), b^+(\lambda))];$$

$$\text{Division: } \forall [\![a(\lambda)]\!] \in \mathbb{GI}, [\![b(\lambda)]\!] \in \mathbb{GI} \setminus \mathbb{Z}: [\![a(\lambda)]\!] \div [\![b(\lambda)]\!] = [\![a(\lambda)]\!] \times (1 \div [\![b(\lambda)]\!]) \quad (18)$$

$$\text{where: } 1 \div [\![b(\lambda)]\!] = 1 \div [\![[b^-(\lambda)], [b^+(\lambda)]]\!] = [\![1\div[b^+(\lambda)], 1\div[b^-(\lambda)]]\!]$$

From a methodological perspective, expressions (15)-(18) are gradual versions of expressions (7)-(10). All the TIs are replaced by TGIs. However, from a practical perspective, there are various differences in the implementations of the multiplication and the division operations, where *min* and *max* operations between GIs are necessary. Indeed, in contrast to CIs, where only a single horizontal dimension is considered, GIs are represented by two dimensions (horizontal and vertical). In this framework, attention must be paid to the points of intersection between ascending (descending) profiles (see Fig. 5).

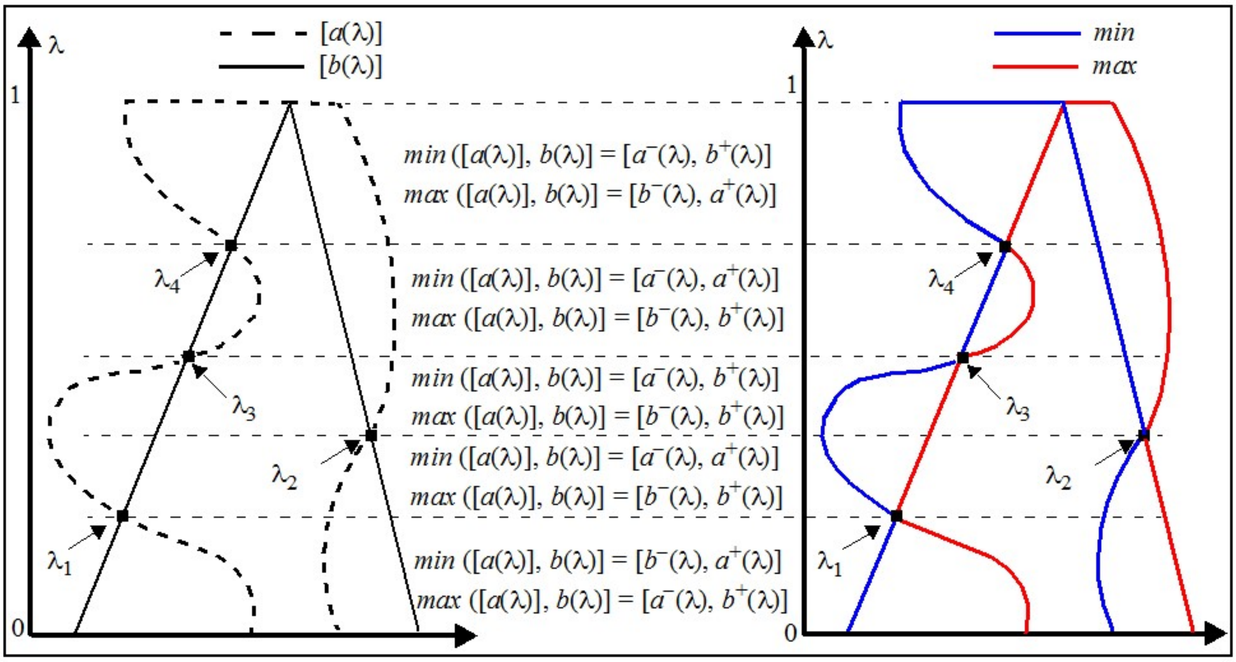


Fig. 5: Example of *min* and *max* operations between two GIs

The break points that delimit areas on the vertical dimension must be determined beforehand to compute the *min* and *max* expressions. A point of intersection consists of a cross between two ascending (descending) profiles. By traversing λ from 0 to 1, at each intersection between two left or right ascending (descending) profiles, the *min* and *max* expressions change (see Fig. 5 for an illustration). A more complete analysis with a generic expression for computing the *min* and *max* operators between GIs can be found in [6].

V.3. Example 4: Is the TGI representation useful?

Here, the robot example of Section III.3 is considered again. The visibility zone is specified by the following TGI:

$$\forall\lambda\in[0,1]: [\![z(\lambda)]\!] = [\![[z^-(\lambda)], [z^+(\lambda)]]\!] = [x(\lambda)] + [\![-[d(\lambda)], [d(\lambda)]]\!]$$
$$= [7+3\lambda, 12-2\lambda] + [\![-[3+\lambda, 5-\lambda], [3+\lambda, 5-\lambda]]\!]$$

The GI $[7+3\lambda, 12-2\lambda]$ is a particular case of a TGI, namely,

$$[7+3\lambda, 12-2\lambda] = [\![[7+3\lambda, 7+3\lambda], [12-2\lambda, 12-2\lambda]]\!]$$

Hence,

$$[\![z(\lambda)]\!] = [\![[7+3\lambda, 7+3\lambda]-[3+\lambda, 5-\lambda], [12-2\lambda, 12-2\lambda]+[3+\lambda, 5-\lambda]]\!]$$
$$= [\![[2+4\lambda, 4+2\lambda], [15-\lambda, 17-3\lambda]]\!]$$

The gradual visibility zone $[\![z(\lambda)]\!]$ is illustrated in Fig. 6.

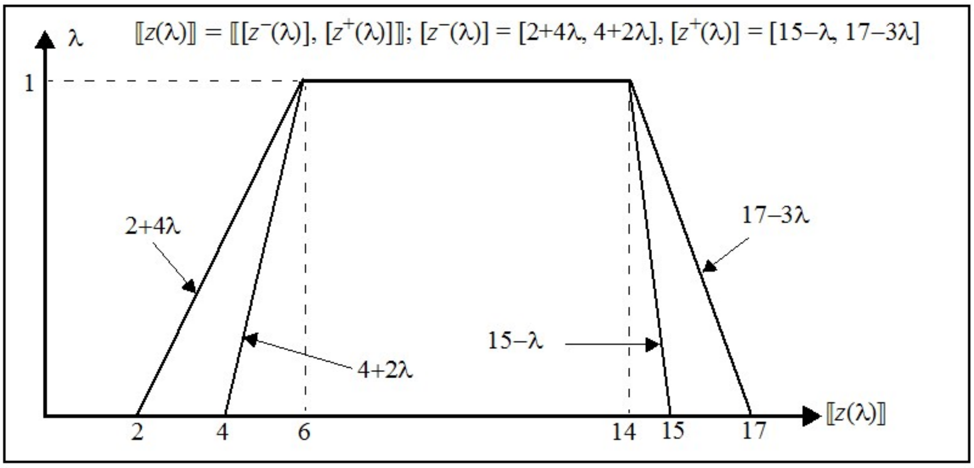


Fig. 6: Visibility zone as a thick gradual interval

V.4. T2FIs *versus* TGIs

A T2FI is characterized by a type-2 membership function, which is represented by two type-1 membership functions, namely, the lower function (inf) and the upper function (sup). In this context, a T2FI, which is denoted $\tilde{A}$, is completely defined by these two T1FIs $A^{\inf}$ and $A^{\sup}$, which are defined by their membership functions $\mu_A^{\inf}(x)$ and $\mu_A^{\sup}(x)$ subject to the constraint $\mu_A^{\inf}(x) < \mu_A^{\sup}(x)$ (see Fig. 7).

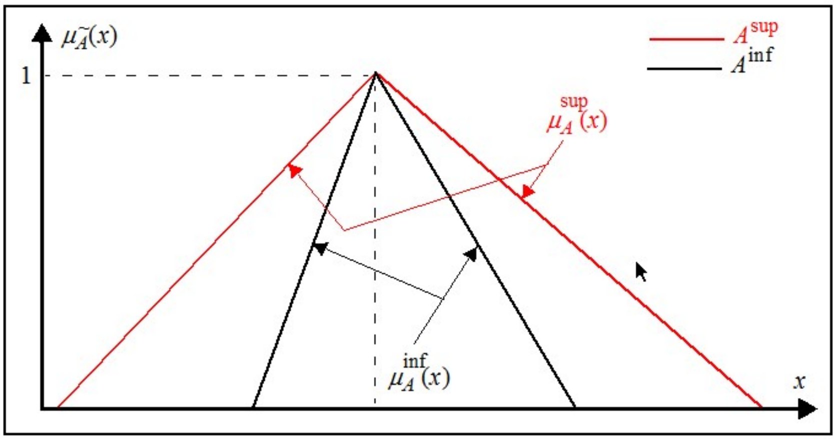


Fig. 7: Conventional representation of a T2FI

If this conventional representation remains useful, its manipulation through fuzzy arithmetic operations is relatively difficult. For instance, according to the α-plane principle, it is possible to define a T2FI as a collection of CI sets. Each of these CI sets may be independently processed for implementing the arithmetic operations. However, this is computationally expensive and sometimes requires the use of massively parallel processing units such as graphical processing units (GPUs) [31]. This high computational cost has hindered the progress of T2FI arithmetic. Furthermore, the α-plane concept underestimates the results and cannot always guarantee rigorous enclosures for the ranges of T2FI operations. In this paper, through the GI and TI concepts, a new interpretation of T2FIs is proposed. Thus, a T2FI is composed of two pure GIs, namely, $[a^-(\lambda)]$ and $[a^+(\lambda)]$: one that represents the left part and the other the right part (see Fig. 8). In this framework, according to the TI formalism, a T2FI can be regarded as a TGI (and *vice versa*): $[\![a(\lambda)]\!] = [\![[a^-(\lambda)], [a^+(\lambda)]]\!]$. The passage between the GI and membership function representations is straightforward via the inverse functions.

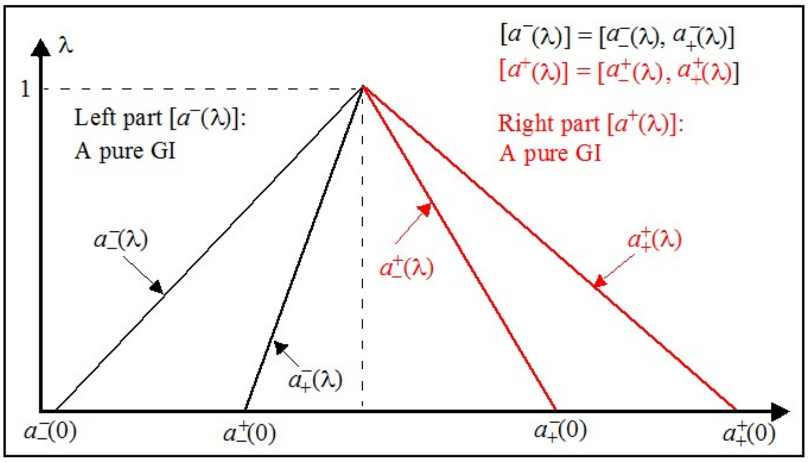


Fig. 8: Representation of a T2FI as a TGI

The quantities $[a^-(\lambda)]$ and A^{inf} are not equivalent; $[a^+(\lambda)]$ and A^{sup} are also not equivalent. For instance, the TGI $[\![z(\lambda)]\!]$ that represents the visibility zone (see Fig. 6) can be regarded as a T2FI, which is denoted as $\tilde{Z}$ and represented by its type-1 membership functions $\mu_Z^{inf}(x)$ and $\mu_Z^{sup}(x)$:

$$\mu_Z^{inf}(x) = \begin{cases} (x-4)/2\ ; & 4 \le x \le 6 \\ 1\ ; & 6 \le x \le 14\ ; \\ (-x+15)\ ; & 14 \le x \le 15 \end{cases} \quad \mu_Z^{sup}(x) = \begin{cases} (x-2)/4\ ; & 2 \le x \le 6 \\ 1\ ; & 6 \le x \le 14 \\ (-x+17)/3\ ; & 14 \le x \le 17 \end{cases}$$

Fig. 9 interprets the TGI $[\![z(\lambda)]\!]$, which is represented in Fig. 6, as a T2FI with its type-2 membership function.

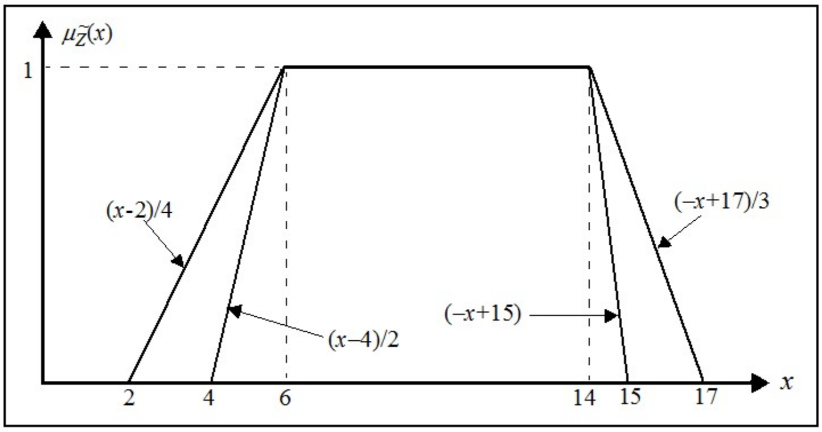


Fig. 9: T2FI representation and equivalence

This new TGI approach enables the extension of the interval arithmetic arsenal to T2FIs. Moreover, it renders the T2FIs computations more tractable and guarantees rigorous enclosures for the ranges of operations and functions. The proposed approach is analyzed in the next section through an illustrative example.

VI. Illustrative example of the use of TGI arithmetic

In this section, a computational example is presented to emphasize important points and show the advantages of the proposed concepts. Furthermore, comparative results between our approach and the α-plane methodology are presented.

VI.1. Thick gradual interval arithmetic

Let us consider two triangular T2FIs: $\tilde{A}$ and $\tilde{B}$. Each T2FI is completely defined by its two upper (sup) and lower (inf) T1FIs (see Fig. 10).

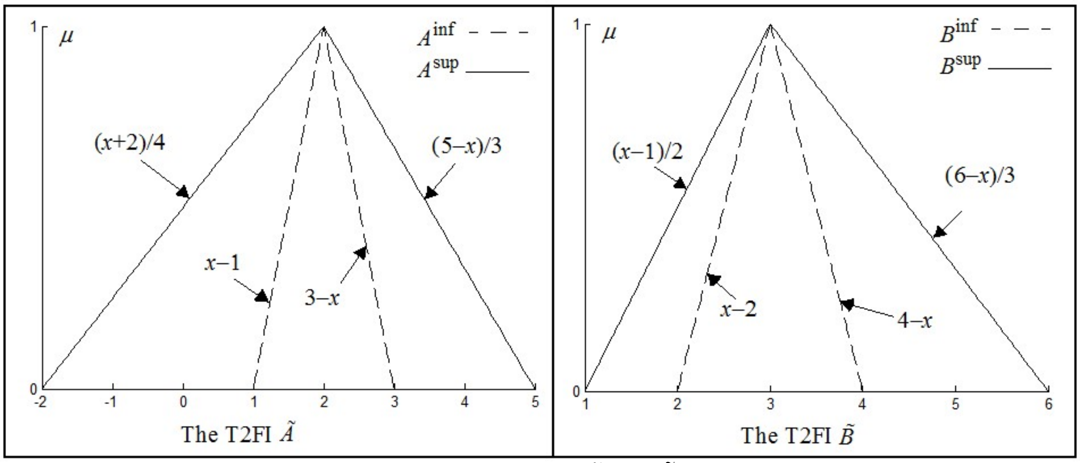


Fig. 10: T2FIs $\tilde{A}$ and $\tilde{B}$

For example, T2FI $\tilde{A}$ is defined by A^{inf} and A^{sup}, which are characterized by their membership functions $\mu_A^{inf}(x)$ and $\mu_A^{sup}(x)$ subject to the constraint $\mu_A^{inf}(x) < \mu_A^{sup}(x)$. T2FI $\tilde{B}$ is defined similarly (see Fig. 9):

$$\mu_A^{sup}(x) = \begin{cases} (x+2)/4 \; ; \; -2 \le x \le 2 \\ (-x+5)/3 \; ; \; 2 \le x \le 5 \end{cases} \; ; \; \mu_A^{inf}(x) = \begin{cases} x-1 \; ; \; 1 \le x \le 2 \\ -x+3 \; ; \; 2 \le x \le 3 \end{cases} \text{; and:}$$

$$\mu_B^{sup}(x) = \begin{cases} (x-1)/2 \; ; \; 1 \le x \le 3 \\ (-x+6)/3 \; ; \; 3 \le x \le 6 \end{cases} \; ; \; \mu_B^{inf}(x) = \begin{cases} x-2 \; ; \; 2 \le x \le 3 \\ -x+4 \; ; \; 3 \le x \le 4 \end{cases}$$

T2FIs $\tilde{A}$ and $\tilde{B}$ can be expressed as TGIs ⟦$a(\lambda)$⟧ and ⟦$b(\lambda)$⟧, as illustrated in Fig. 11 and expressed as follows:

$$[\![a(\lambda)]\!] = [\![[a^-(\lambda)], [a^+(\lambda)]]\!] = [\![[-2+4\lambda, 1+\lambda], [3-\lambda, 5-3\lambda]]\!] \text{; and:}$$

$$[\![b(\lambda)]\!] = [\![[b^-(\lambda)], [b^+(\lambda)]]\!] = [\![[1+2\lambda, 2+\lambda], [4-\lambda, 6-3\lambda]]\!]$$

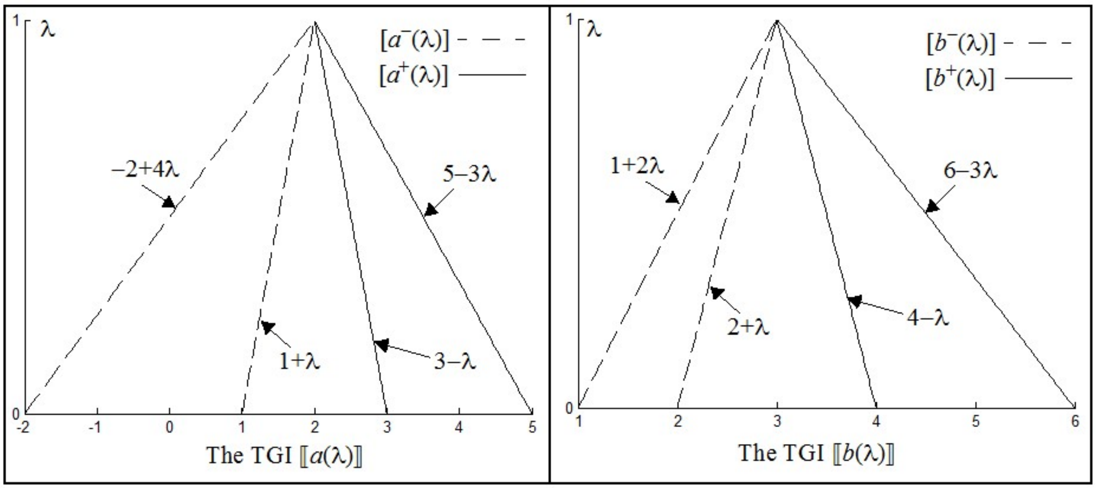


Fig. 11: TGIs ⟦a⟧ and ⟦b⟧

The SIA operations over the TGIs are expressed as follows:

- Addition: $[\![a(\lambda)]\!]+[\![b(\lambda)]\!] = [\![[-2+4\lambda,\ 1+\lambda]+[1+2\lambda,\ 2+\lambda],\ [3-\lambda,\ 5-3\lambda]+[4-\lambda,\ 6-3\lambda]]\!]$
  $= [\![[-1+6\lambda,\ 3+2\lambda],\ [7-2\lambda,\ 11-6\lambda]]\!]$
- Subtraction: $[\![a(\lambda)]\!]-[\![b(\lambda)]\!] = [\![[-2+4\lambda,\ 1+\lambda]-[4-\lambda,\ 6-3\lambda],\ [3-\lambda,\ 5-3\lambda]-[1+2\lambda,\ 2+\lambda]]\!]$
  $= [\![[-8+7\lambda,\ -3+2\lambda],\ [1-2\lambda,\ 4-5\lambda]]\!]$

The TGIs that result from addition and subtraction of TGIs $[\![a(\lambda)]\!]$ and $[\![b(\lambda)]\!]$ are illustrated in Fig. 12.

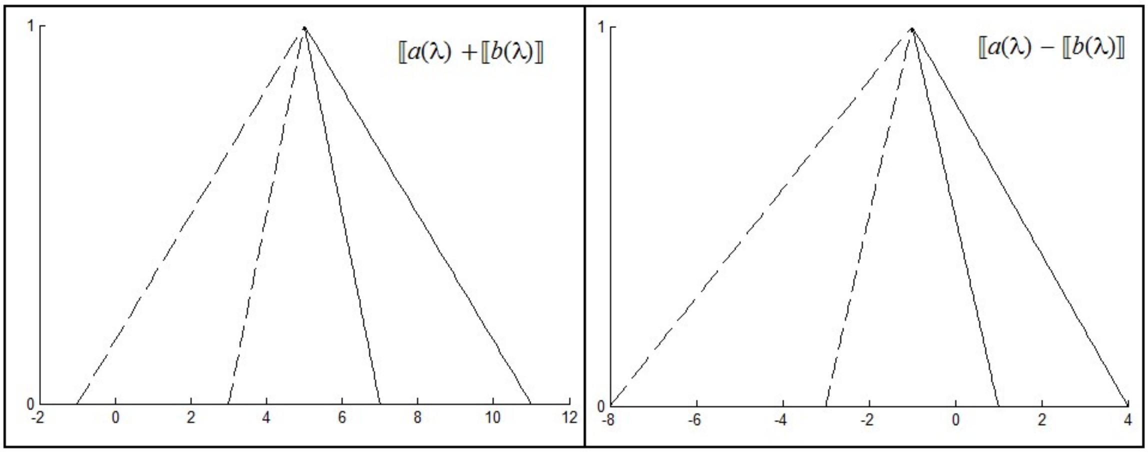


Fig. 12: Addition and subtraction results between $[\![a(\lambda)]\!]$ and $[\![b(\lambda)]\!]$

- Multiplication: $[\![a(\lambda)]\!]\times[\![b(\lambda)]\!] = [\![min(\Phi_1(\lambda)),\ max(\Phi_1(\lambda))]\!]$; $\Phi_1(\lambda)=\{[c_1(\lambda)],\ [c_2(\lambda)],\ [c_3(\lambda)],\ [c_4(\lambda)]\}$

The computation of the GIs $[c_1(\lambda)]$, $[c_2(\lambda)]$, $[c_3(\lambda)]$ and $[c_4(\lambda)]$ yields the following:

$$[c_1(\lambda)] = [a^-(\lambda)]\times[b^-(\lambda)] = \begin{cases}[(-2+4\lambda)(2+\lambda),(1+\lambda)(2+\lambda)];\ \text{if: } 0\le\lambda\le 0.5\\ [(-2+4\lambda)(1+2\lambda),(1+\lambda)(2+\lambda)];\ \text{if: } 0.5<\lambda\le 1\end{cases}$$

$$[c_2(\lambda)] = [a^-(\lambda)]\times[b^-(\lambda)] = \begin{cases}[(-2+4\lambda)(6-3\lambda),(1+\lambda)(6-3\lambda)];\ \text{if: } 0\le\lambda\le 0.5\\ [(-2+4\lambda)(4-\lambda),(1+\lambda)(6-3\lambda)];\ \text{if: } 0.5<\lambda\le 1\end{cases}$$

$$[c_3(\lambda)] = [a^+(\lambda)]\times[b^-(\lambda)] = [(3-\lambda)(1+2\lambda),\ (5-3\lambda)(2+\lambda)]$$

$$[c_4(\lambda)] = [a^+(\lambda)]\times[b^+(\lambda)] = [(3-\lambda)(4-\lambda),\ (5-3\lambda)(6-3\lambda)]$$

The operation $min(\Phi_1(\lambda))$ is computed as follows:

$$min(\Phi_1(\lambda)) = min(min([c_1(\lambda)],\ [c_3(\lambda)]),\ min([c_2(\lambda)],\ [c_4(\lambda)]))$$

According to Fig. 13,

$$min([c_1(\lambda)],\ [c_3(\lambda)]) = [c_1(\lambda)];\ \text{and } min\ ([c_2(\lambda)],\ [c_4(\lambda)]) = [c_2(\lambda)].$$

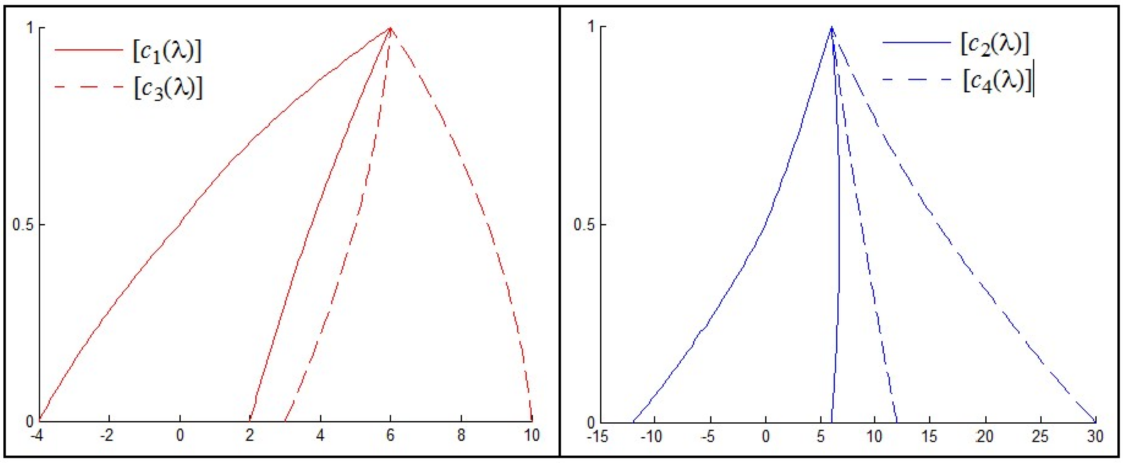


Fig. 13: GIs $[c_1(\lambda)]$, $[c_2(\lambda)]$, $[c_3(\lambda)]$ and $[c_4(\lambda)]$

In this scenario, the computational principle of $min(\Phi_1(\lambda))$ between GIs $[c_1(\lambda)]$ and $[c_2(\lambda)]$ is shown in Fig. 14 and leads to the following expression:

$$min(\Phi_1(\lambda)) = min([c_1(\lambda)], [c_2(\lambda)]) = \begin{cases} [c_2^-, c_1^+] = [(-2+4\lambda)(6-3\lambda), (1+\lambda)(2+\lambda)]; \text{ if: } 0 \le \lambda \le 0.5 \\ [c_1^-, c_1^+] = [(-2+4\lambda)(2+\lambda), (1+\lambda)(2+\lambda)]; \text{ if: } 0.5 < \lambda \le 1 \end{cases}$$

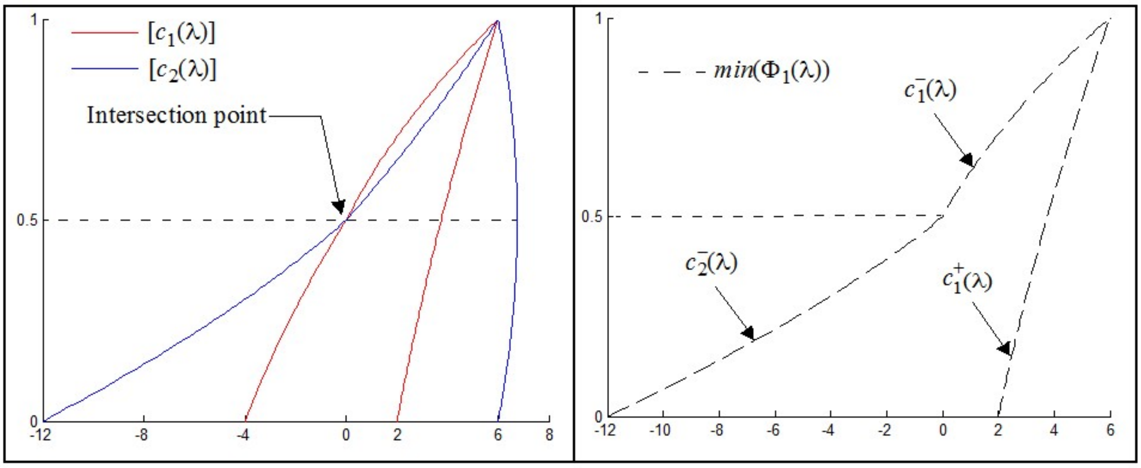


Fig. 14: Computational principle of $min(\Phi_1(\lambda))$

By adopting the same principle as for the *min* operator, $max(\Phi_1(\lambda))$ is expressed as follows (see Fig. 15.a.):

$$max(\Phi_1(\lambda)) = max(max([c_1(\lambda)], [c_3(\lambda)]), max([c_2(\lambda)], [c_4(\lambda)]) = max([c_3(\lambda)], [c_4(\lambda)] = [c_4(\lambda)]$$
$$= [c_4^-, c_4^+] = [(3-\lambda)(4-\lambda), (5-3\lambda)(6-3\lambda)]; \quad 0 \le \lambda \le 1$$

Finally, the TGI that results from the application of the multiplication operator is illustrated in Fig. 15.b.

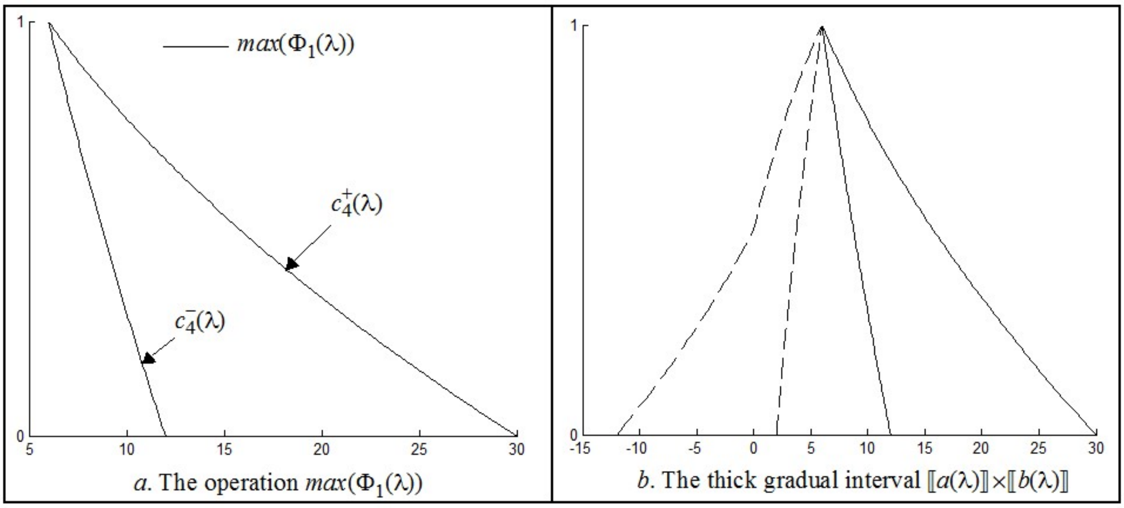


Fig. 15: $max(\Phi_1(\lambda))$ and the product $⟦a(\lambda)⟧\times⟦b(\lambda)⟧$

- Division: $⟦a(\lambda)⟧\div⟦b(\lambda)⟧ = ⟦a(\lambda)⟧\times(1\div⟦b(\lambda)⟧)$

$$= ⟦[a^-(\lambda)], [a^+(\lambda)]⟧\times(⟦1\div[b^+(\lambda)], 1\div[b^-(\lambda)]⟧)$$
$$= ⟦[-2+4\lambda, 1+\lambda], [3-\lambda, 5-3\lambda]⟧\times⟦1\div[4-\lambda, 6-3\lambda], 1\div[1+2\lambda, 2+\lambda]⟧$$
$$= ⟦[-2+4\lambda, 1+\lambda], [3-\lambda, 5-3\lambda]⟧\times⟦[1/(6-3\lambda), 1/(4-\lambda)], [1/(2+\lambda), 1/(1+2\lambda)]⟧$$
$$= ⟦min(\Phi_2(\lambda)), max(\Phi_2(\lambda))⟧; \Phi_2(\lambda) = \{[c_{11}(\lambda)], [c_{22}(\lambda)], [c_{33}(\lambda)], [c_{44}(\lambda)]\};$$

where:

$$[c_{11}(\lambda)] = [-2+4\lambda, 1+\lambda]\times[1/(6-3\lambda), 1/(4-\lambda)]$$
$$[c_{22}(\lambda)] = [-2+4\lambda, 1+\lambda]\times[1/(2+\lambda), 1/(1+2\lambda)]$$
$$[c_{33}(\lambda)] = [3-\lambda, 5-3\lambda]\times[1/(6-3\lambda), 1/(4-\lambda)]$$
$$[c_{44}(\lambda)] = [3-\lambda, 5-3\lambda]\times[1/(2+\lambda), 1/(1+2\lambda)]$$

As $[c_{11}(\lambda)] \preceq [c_{33}(\lambda)]$ and $[c_{22}(\lambda)] \preceq [c_{44}(\lambda)]$, *min* and *max* can be computed as follows:

$$min(\Phi_2(\lambda)) = min(min([c_{11}(\lambda)], [c_{33}(\lambda)]), min([c_{22}(\lambda)], [c_{44}(\lambda)])) = min([c_{11}(\lambda)], [c_{22}(\lambda)])$$

$$= \begin{cases} [c_{22}^-, c_{11}^+] = [(-2+4\lambda)/(1+2\lambda), (1+\lambda)/(4-\lambda)]; \text{ if: } 0 \le \lambda \le 0.5 \\ [c_{11}^-, c_{11}^+] = [(-2+4\lambda)/(6-3\lambda), (1+\lambda)/(4-\lambda)]; \text{ if: } 0.5 < \lambda \le 1 \end{cases}$$

$$max(\Phi_2(\lambda)) = max(max([c_{11}(\lambda)], [c_{33}(\lambda)]), max([c_{22}(\lambda)], [c_{44}(\lambda)])) = max([c_{33}(\lambda)], [c_{44}(\lambda)]) = [c_{44}(\lambda)]$$
$$= [c_{44}^-, c_{44}^+] = [(3-\lambda)/(2+\lambda), (5-3\lambda)/(1+2\lambda)]; \quad 0 \le \lambda \le 1$$

Finally, the TGI that results from the application of the division operation is illustrated in Fig. 16.

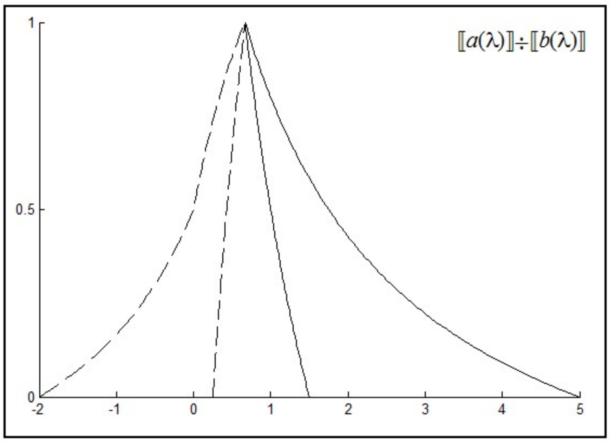


Fig. 16: Result of the division operator

VI.2. Arithmetic operations that use α-planes: comparison and discussion

The α-cut principle enables the representation of T1FIs $A^{\inf}$ and $A^{\sup}$ as follows (see Fig. 17):

$$A^{\inf} = \bigcup_{\alpha\in[0,1]} \alpha \cdot [\underline{a}_1^{\alpha}, \underline{a}_2^{\alpha}]; \text{ and: } A^{\sup} = \bigcup_{\alpha\in[0,1]} \alpha \cdot [\overline{a}_1^{\alpha}, \overline{a}_2^{\alpha}]$$

In this context, the arithmetic operations on T1FIs can be implemented using SIA over CIs. For example, when T1FIs $A^{\inf}$ and $B^{\inf}$ are considered, the arithmetic operations are expressed as follows:

$$C^{\inf} = A^{\inf} \odot B^{\inf} = \bigcup_{\alpha\in[0,1]} \alpha \cdot ([\underline{a}_1^{\alpha}, \underline{a}_2^{\alpha}] \odot [\underline{b}_1^{\alpha}, \underline{b}_2^{\alpha}]); \odot \in \{+,-,\times,\div\}$$

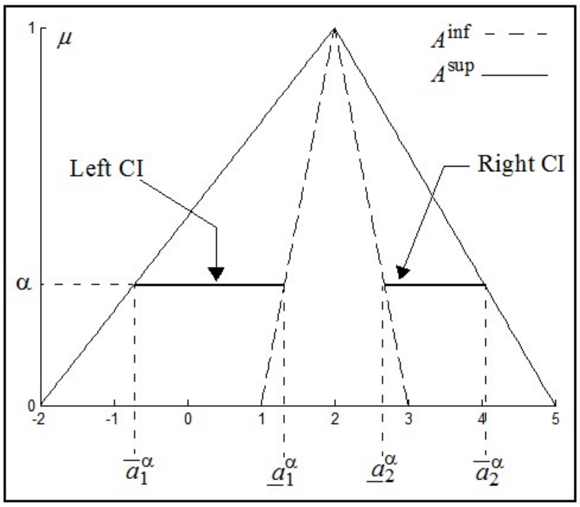


Fig. 17: α-plane principle on T2FI $\tilde{A}$

This representation approach has been extended to T2FIs through the concept of α-planes. Thus, a T2FI $\tilde{A}$ can be defined by the following α-plane representation [29] (see Fig. 17):

$$\tilde{A} = \bigcup_{\alpha\in[0,1]} \alpha \cdot ([\overline{a}_1^{\alpha}, \underline{a}_1^{\alpha}], [\underline{a}_2^{\alpha}, \overline{a}_2^{\alpha}])$$

Furthermore, T2FI $\tilde{A}$ is defined by the union of all its α-T2FIs cuts. This representation is interesting because, for each level α, the T2FI can be characterized by two CIs (left and right CIs). This suggests that the operations on T2FIs can be performed using available operations over CIs. For instance, the application of the α-plane principle on T2FIs $\tilde{A}$ and $\tilde{B}$ with a sampling step size of 0.01 is illustrated in Fig. 18. This small sampling step size is selected with the objective of obtaining accurate results.

Via this approach, the arithmetic operations on T1FIs are extended to T2FIs. The elementary arithmetic operations of two T2FIs $\tilde{A}$ and $\tilde{B}$ are expressed as follows:

$$\tilde{C} = \tilde{A} \odot \tilde{B} = \bigcup_{\alpha\in[0,1]} \alpha \cdot ([\overline{a}_1^{\alpha}, \underline{a}_1^{\alpha}] \odot [\overline{b}_1^{\alpha}, \underline{b}_1^{\alpha}], [\underline{a}_2^{\alpha}, \overline{a}_2^{\alpha}] \odot [\underline{b}_2^{\alpha}, \overline{b}_2^{\alpha}]) \; ; \odot \in \{+,-,\times,\div\}$$

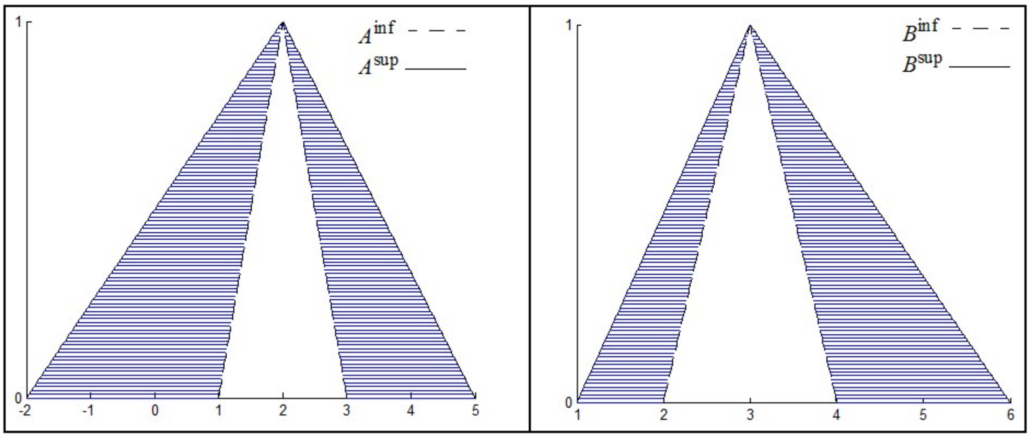
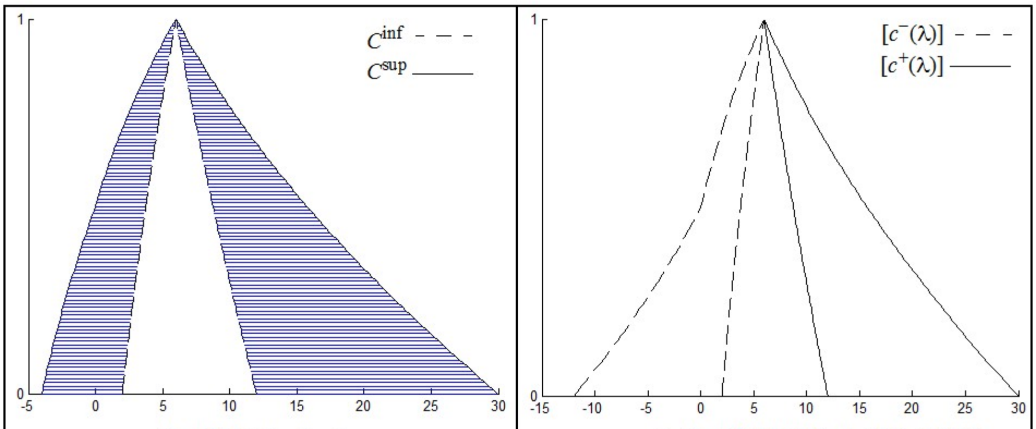


Fig. 18: α-plane representations of $\tilde{A}$ and $\tilde{B}$

Let us demonstrate this method and compare it with the proposed approach. For concision and without loss of generality, the illustrations are presented only for the multiplication operator. Thus, the result of the multiplication operator between $\tilde{A}$ and $\tilde{B}$ that uses the α-plane principle is illustrated in Fig. 19.a. For comparison, Fig. 19.b illustrates the multiplication result that is obtained via the TGI approach.



a. The T2FI $\tilde{C} = \tilde{A} \times \tilde{B}$ *b.* The TGI $[\![c(\lambda)]\!] = [\![a(\lambda)]\!] \times [\![b(\lambda)]\!]$

Fig. 19: Multiplication results of T2FIs that were obtained using the α-plane and the TGI approaches

The two approaches do not produce the same result. Indeed, in the α-plane approach, at each level α, a T2FI is not regarded as a unique interval with uncertain bounds but as two independent left and right CIs. In this context, the operations are performed between left (and right) CIs separately. This computing approach underestimates the results and does not permit rigorous enclosures for the ranges of operations. Let us present an example of this phenomenon. Consider two T1FIs A and B that are defined by their membership functions $\mu_A(x)$ and $\mu_B(x)$ such that $\mu_A^{\inf}(x) \le \mu_A(x) \le \mu_A^{\sup}(x)$ and $\mu_B^{\inf}(x) \le \mu_B(x) \le \mu_B^{\sup}(x)$ (see Fig. 20):

$$\mu_A(x) = \begin{cases} (4x+7)/15 \; ; \; -1.75 \le x \le 2 \\ (-x+4)/2 \; ; \; 2 \le x \le 4 \end{cases} \; ; \; \mu_B(x) = \begin{cases} (2x-3)/3 \; ; \; 1.5 \le x \le 3 \\ (-2x+11)/5 \; ; \; 3 \le x \le 5.5 \end{cases}$$

In this case, the T2FIs are regarded as uncertain representations of T1FIs. Indeed, A and B are regarded as possible realizations of T1FIs in their likelihood T2FIs. The computations of the multiplication operation between T1FIs A and B according to the α-cut and gradual arithmetic approaches are illustrated in Fig. 21.

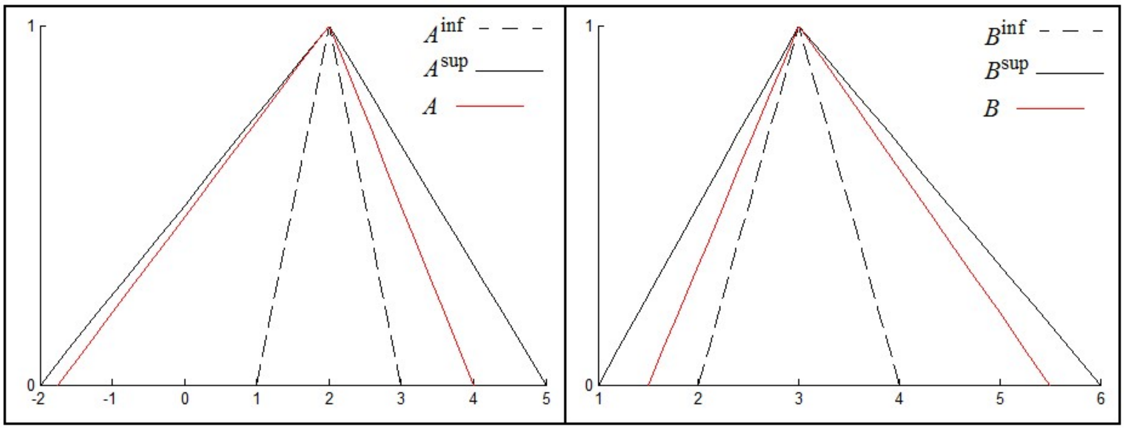


Fig. 20: Representations of T1FIs *A* and *B*

Although the discretization procedure requires a longer computing time, the two approaches yield strictly equivalent results. We recall here that in the GI arithmetic, T1FI *A* can be regarded as a monotone GI $[a(\lambda)] = [a^{-}(\lambda), a^{+}(\lambda)]$, where its GNs bounds $a^{-}(\lambda)$ and $a^{+}(\lambda)$ are computed such that

$$\begin{cases} \lambda = (4x+7)/15 \Rightarrow a^{-}(\lambda) = -1.75 + 3.75\lambda \\ \lambda = (4-x)/2 \Rightarrow a^{+}(\lambda) = 4 - 2\lambda \end{cases}$$

The same principle is applied to T1FI *B*, which leads to $[b(\lambda)] = [1.5+1.5\lambda, 5.5–2.5\lambda]$.

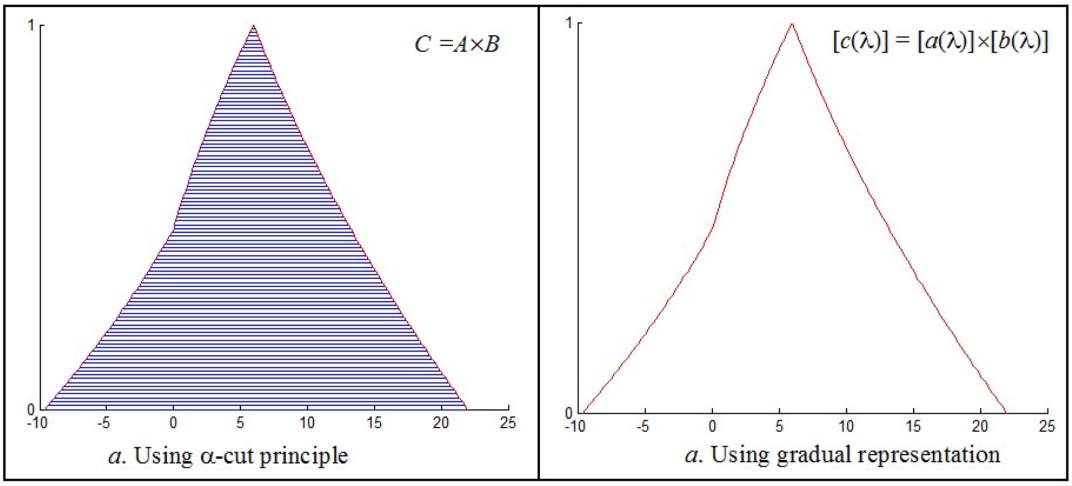


Fig. 21: Multiplication results between T1FIs using the α-cut and GI approaches

When the T1FIs of Fig. 21 are positioned in the T2FIs that result from the multiplication operations, the results that are presented in Fig. 22 are obtained.

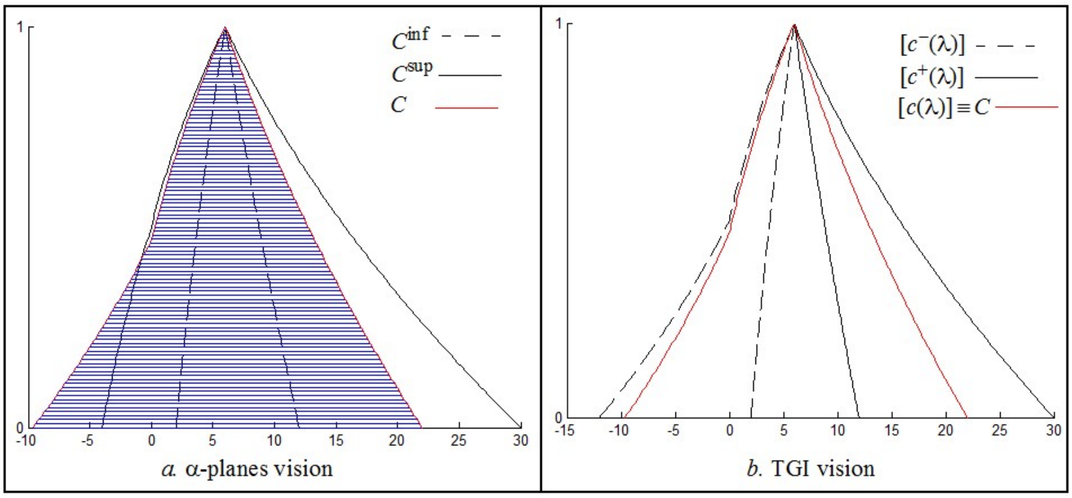


Fig. 22: Results of multiplication between T1FIs and T2FIs

According to the results of Fig. 22, although $A \subseteq \tilde{A}$ and $B \subseteq \tilde{B}$, $C = A \times B \not\subset \tilde{C} = \tilde{A} \times \tilde{B}$. Thus, the α-plane arithmetic does not guarantee the containment of the set of all possible results, namely, the T2FI $\tilde{C} = \tilde{A} \times \tilde{B}$ that was obtained via the α-plane method is not guaranteed to contain the multiplication result of the two T1FIs $A \subseteq \tilde{A}$ and $B \subseteq \tilde{B}$. Furthermore, this approach may lead to the exclusion of information in the realized operation (the underestimation problem). This phenomenon can degrade the relevance and the performance of this approach. In real applications, special attention must be paid to this phenomenon prior to the validation of the arithmetic operations. In contrast to the α-plane approach, the proposed methodology, which is based on SIA operations over TGIs, ensures rigorous enclosures for the ranges of operations since the results are TGIs in which the uncertain GI (T1FIs) results must lie.

VI.3. Remarks and discussion

- Based on its computational mechanism, the α-plane approach can lead to counterintuitive or incorrect results, especially if the T2FIs are negative or differ in sign. Indeed, as illustrated in Fig. 23, the product of two negative T2FIs, namely, $\tilde{C} = \tilde{A} \times \tilde{A}$, leads to a quantity where the $C^{\inf}$ and $C^{\sup}$ bounds are permuted. Thus, $\tilde{C}$ is not a T2FI.

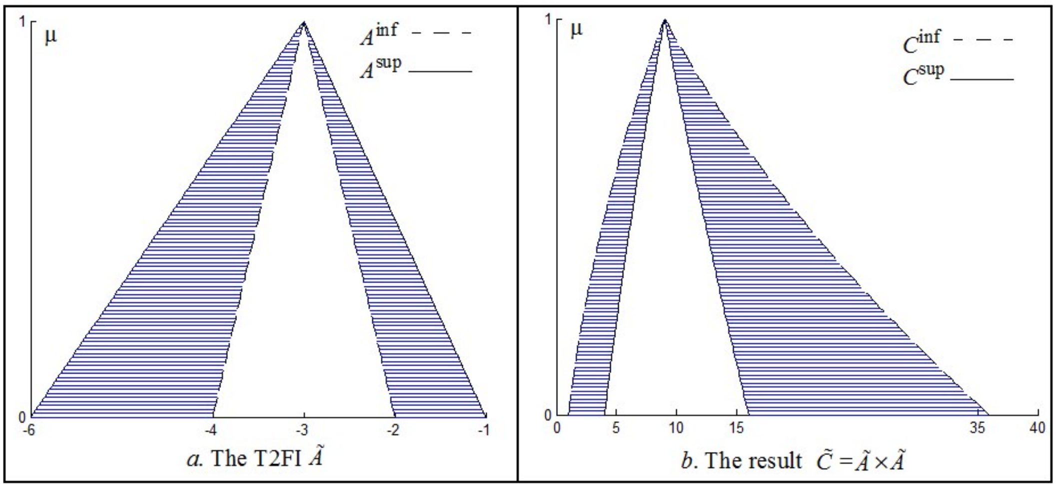


Fig. 23: Multiplication of two negative T2FIs

- More generally, a reflection on the relevance of the α-plane method can lead to the proposal of a new formulation of its representation to correct its anomalies. Furthermore, at a level α, the T2FI will not be represented by two separate and independent CIs, but by a single TI for which the left and right CIs become its uncertain bounds. In this case, an alternative to the conventional α-plane representation can be reformulated as follows:

$$\tilde{A} = \bigcup_{\alpha \in [0,1]} \alpha \cdot \left(\left[\!\left[[\overline{a}_1^{\alpha}, \underline{a}_1^{\alpha}], [\underline{a}_2^{\alpha}, \overline{a}_2^{\alpha}] \right]\!\right] \right) = \bigcup_{\alpha \in [0,1]} \alpha \cdot \left(\left[\!\left[[a_\alpha^-], [a_\alpha^+] \right]\!\right] \right)$$

Thus, at each level α, the operations between T2FIs can be implemented through the arithmetic over TIs. It is concluded that this alternative formulation of the α-plane principle can be regarded as a discrete version of the TGI approach.

- For T2FIs with nonlinear shapes, the sampling step must be small to preserve the form of the resultant T2FIs. This mechanism can substantially increase the computation time and, thus, weaken the applicability of the methods that are based on discretization procedures (e.g., α-planes, F-transform, and Zslices). The proposed method can overcome this constraint of discretization to obtain an analytical formulation of the results.

VII. Potential applications of TGI computations

VII.1. Potential use in computing T2FI aggregation operators

Crisp aggregation operators for real numbers were extended to aggregation operators for intervals and T1FIs (see, for example, [19][42][52]). Although the research on aggregation methods that are based on T1FIs has expanded substantially, the research on type-2 fuzzy methods is scant. Furthermore, interesting methodologies have been proposed in the literature [53][54][67][68][73][76]. An interesting analysis of aggregation operators in the type-2 fuzzy framework is presented in [68]. For instance, in [67], Zadeh's extension principle has been used to extend type-1 aggregation operators to T2FSs. In [73], the concept of the linguistic weighted average has been proposed. In [53][54], aggregation operators, such as weighted aggregation operators and geometric aggregation operators, are used in multiple-attribute group decision-making problems. A type-2 OWA operator is proposed in [76] for aggregating linguistic opinions in decision-making problems. All the above type-2 aggregation operators are based on the α-cut decomposition or Zadeh's extension principle. The main difference between the existing approaches and the proposed approach is that the aggregation operators are computed based on the TGI concept, where the obtained results are exact and purely analytical. Furthermore, the flexibility and the rigor of IA computations are preserved.

The TGI arithmetic and reasoning can be used to implement all aggregation operators (conjunctive and disjunctive operators, weighted average and ordered weighted average operators, the Choquet integral, etc.). Using an example, this section aims at demonstrating the potential application of our computational methodology to the 2-additive Choquet integral (2-ACI) [42].

Let us consider a set of crisp alternatives $\{a_1, \ldots, a_n\}$ to be aggregated and associated with a set of n criteria. The 2-ACI is expressed as follows:

$$CI = \sum_{I_{ij}>0} min(a_i, a_j)\cdot I_{ij} + \sum_{I_{ij}<0} max(a_i, a_j)\cdot |I_{ij}| + \sum_{i=1}^{n} a_i \cdot (v_i - \frac{1}{2}\cdot \sum_{j\neq i} |I_{ij}|) \qquad (19)$$

In (19), the coefficient I_{ij} represents the mutual interaction between criteria i and j and can be interpreted as follows:

• Positive I_{ij} corresponds to complementary criteria (positive synergy);
• Negative I_{ij} corresponds to redundant criteria (negative synergy);
• Null I_{ij} corresponds to no interaction between the criteria (the criteria are independent).

The coefficients v_i in (19) are the Shapley indices, which represent the relative importance of each elementary criterion relative to all the others, with $\sum_{i=1}^{n} v_i = 1$. The 2-ACI has been extended to the fuzzy context, where the alternatives are represented by T1FIs (GIs) [9][42]. This extension led to the following formulation:

$$[CI(\lambda)] = \sum_{I_{ij}>0} min([a_i(\lambda)],[a_j(\lambda)])\times I_{ij} + \sum_{I_{ij}<0} max([a_i(\lambda)],[a_j(\lambda)])\times |I_{ij}| + \sum_{i=1}^{n} [a_i(\lambda)]\times (v_i - \frac{1}{2}\sum_{j\neq i} |I_{ij}|) \quad (20)$$

In (20), $[a_1(\lambda)], \ldots, [a_n(\lambda)]$ are T1FI alternatives. Now, our objective is to extend (20) to the scenario in which the alternatives are uncertain and are represented by T2FIs. In this context, the 2-ACI that is specified by (20) becomes

$$[\![CI(\lambda)]\!] = \sum_{I_{ij}>0} min([\![a_i(\lambda)]\!],[\![a_j(\lambda)]\!])\times I_{ij} + \sum_{I_{ij}<0} min([\![a_i(\lambda)]\!],[\![a_j(\lambda)]\!])\times |I_{ij}| + \sum_{i=1}^{n} [\![a_i(\lambda)]\!]\times (v_i - \frac{1}{2}\sum_{j\neq i} |I_{ij}|) \quad (21)$$

In (21), $[\![a_1(\lambda)]\!], \ldots, [\![a_n(\lambda)]\!]$ are TGI (T2FI) alternatives. In the implementation of the 2-ACI that is expressed in (21), *min* and *max* between two TGIs $[\![a(\lambda)]\!]$ and $[\![b(\lambda)]\!]$ are computed as follows:

$$min([\![a(\lambda)]\!], [\![b(\lambda)]\!]) = [\![min([a^-(\lambda)], [b^-(\lambda)]), min([a^+(\lambda)], [b^+(\lambda)])]\!];$$
$$max([\![a(\lambda)]\!], [\![b(\lambda)]\!]) = [\![max([a^-(\lambda)], [b^-(\lambda)]), max([a^+(\lambda)], [b^+(\lambda)])]\!];$$

where the *min* and *max* operators between GIs are computed using the methodology of Section V.2.

Let us illustrate the computation of the TGI 2-ACI for aggregating the four alternatives $[\![a_1(\lambda)]\!], \ldots, [\![a_4(\lambda)]\!]$ that are illustrated in Fig. 24 and listed in Table 1, with $v_1 = 0.4$, $v_2 = 0.35$, $v_3 = 0.1$, $v_4 = 0.15$, $I_{14} = -0.35$, $I_{34} = -0.3$, $I_{13} = 0.15$, and $I_{23} = 0.6$. According to these data values, the TGI 2-ACI is expressed as follows:

$$[\![CI(\lambda)]\!] = 0.15\times min([\![a_1(\lambda)]\!], \{[\![a_3(\lambda)]\!]) + 0.6\times min([\![a_2(\lambda)]\!], \{[\![a_3(\lambda)]\!]) + 0.35\times max([\![a_1(\lambda)]\!], \{[\![a_4(\lambda)]\!])$$
$$+ 0.3\times max([\![a_3(\lambda)]\!], [\![a_4(\lambda)]\!]) + \sum_{i=1}^{n} [\![a_i(\lambda)]\!]\times (v_i - 0.5\sum_{j\neq i} |I_{ij}|)$$

$[\![a_1(\lambda)]\!]$	$[\![[2\lambda, 1+\lambda], [4-2\lambda, 6-4\lambda]]\!]$
$[\![a_2(\lambda)]\!]$	$[\![[2+3\lambda, 3+2\lambda], [8-3\lambda, 9-4\lambda]]\!]$
$[\![a_3(\lambda)]\!]$	$[\![[1+5\lambda, 3+3\lambda], [7-\lambda, 8-2\lambda]]\!]$
$[\![a_4(\lambda)]\!]$	$[\![[3\lambda, 1+2\lambda], [4-\lambda, 5-2\lambda]]\!]$

Table 1: Expressions of the four TGI alternatives

The *min* and *max* operators between the TGIs are computed as follows:

$$min([\![a_1(\lambda)]\!], [\![a_3(\lambda)]\!]) = [\![a_1(\lambda)]\!]; \text{ because } [\![a_1(\lambda)]\!] \preceq [\![a_3(\lambda)]\!]$$

In this case, the TGIs $[\![a_1(\lambda)]\!]$ and $[\![a_3(\lambda)]\!]$) are totally ordered. Indeed, the order relation $\preceq$ between two TGIs $[\![a(\lambda)]\!]$ and $[\![b(\lambda)]\!]$ is defined as follows:

$$[\![a(\lambda)]\!] \preceq [\![b(\lambda)]\!] \Leftrightarrow [a^-(\lambda)] \preceq [b^-(\lambda)] \text{ and } [a^+(\lambda)] \preceq [b^+(\lambda)]$$

Via the same approach, the order relation $\preceq$ between two GIs $[a(\lambda)]$ and $[b(\lambda)]$ is defined as follows:

$$[a(\lambda)] \preceq [b(\lambda)] \Leftrightarrow a^-(\lambda) \leq b^-(\lambda) \text{ and } b^-(\lambda) \leq b^+(\lambda)$$

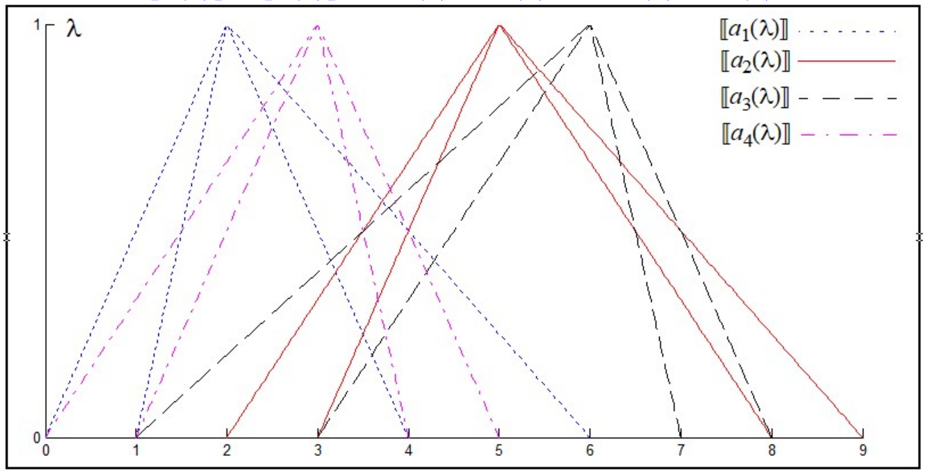


Fig. 24: Shapes of the four TGI alternatives

The order relation between $[\![a_2(\lambda)]\!]$ and $[\![a_3(\lambda)]\!]$ is not total and they cannot be well totally ordered. In this case, the *min* operator is computed as follows (see Fig. 25):

$$min([\![a_2(\lambda)]\!], [\![a_3(\lambda)]\!]) = [\![c(\lambda)]\!] = [\![[c^-(\lambda)], [c^+(\lambda)]]\!]; \text{ with:}$$

$$[c^-(\lambda)] = min([a_2^-(\lambda)], [a_3^-(\lambda)]) = \begin{cases} [1+5\lambda, 3+2\lambda]; \text{if: } 0 \leq \lambda \leq 0.5 \\ [2+3\lambda, 3+2\lambda]; \text{if: } 0.5 < \lambda \leq 1 \end{cases};$$

$$[c^+(\lambda)] = min([a_2^+(\lambda)], [a_3^+(\lambda)]) = \begin{cases} [7-\lambda, 8-2\lambda]; \text{if: } 0 \leq \lambda \leq 0.5 \\ [8-3\lambda, 9-4\lambda]; \text{if: } 0.5 < \lambda \leq 1 \end{cases}$$

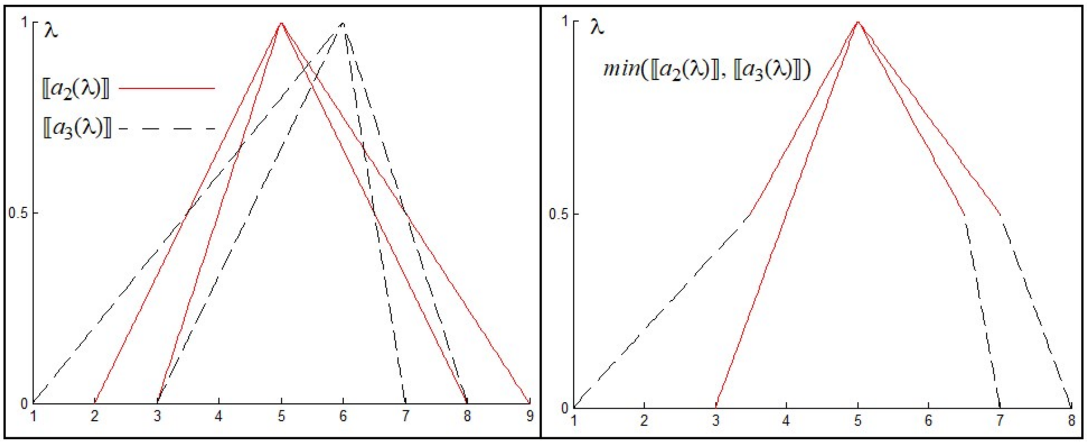


Fig. 25: *min* operator between $[\![a_2(\lambda)]\!]$ and $[\![a_3(\lambda)]\!]$

By applying the same methodology, the *max* operator is computed as follows:

$$max([\![a_3(\lambda)]\!], [\![a_4(\lambda)]\!]) = [\![a_3(\lambda)]\!]; \text{ since } [\![a_4(\lambda)]\!] \preceq [\![a_3(\lambda)]\!]; \text{ and:}$$

$$max(\llbracket a_1(\lambda)\rrbracket, \llbracket a_4(\lambda)\rrbracket) = \llbracket d(\lambda)\rrbracket; \text{ where:}$$

$$[d^-(\lambda)] = max([a_1^-(\lambda)], [a_4^-(\lambda)]) = [a_4^-(\lambda)] = [3\lambda,\ 1+2\lambda]; \text{ because: } [a_1^-(\lambda)] \preceq [a_4^-(\lambda)]$$

$$[d^+(\lambda)] = max([a_1^+(\lambda)], [a_4^+(\lambda)]) = \begin{cases} [4-\lambda, 6-4\lambda]; \text{if: } 0 \le \lambda \le 0.5 \\ [4-\lambda, 5-2\lambda]; \text{if: } 0.5 < \lambda \le 1 \end{cases}$$

The results of *max* between $\llbracket a_1(\lambda)\rrbracket$ and $\llbracket a_4(\lambda)\rrbracket$ are illustrated in Fig. 26.

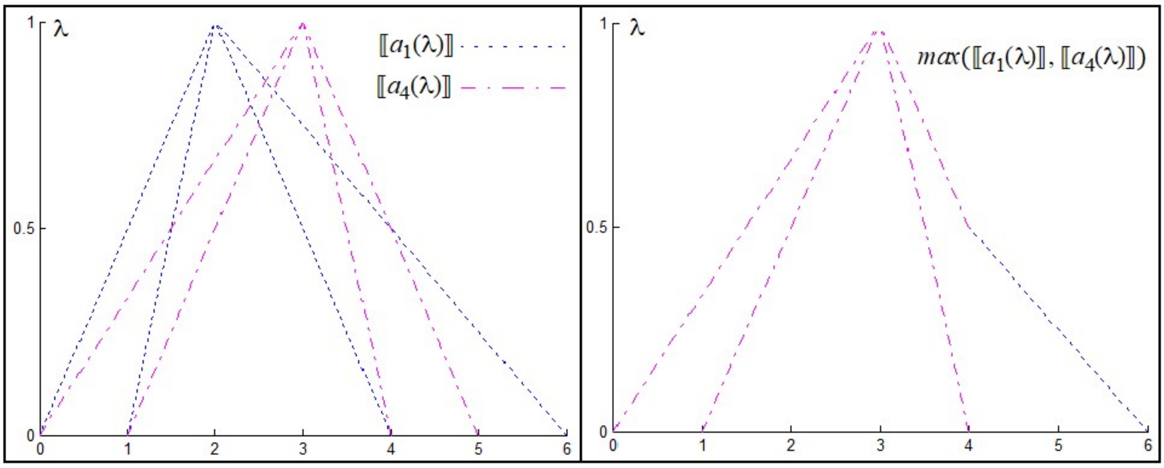


Fig. 26: *max* operator between $\llbracket a_1(\lambda)\rrbracket$ and $\llbracket a_4(\lambda)\rrbracket$

Using the TGI arithmetic, the final 2-ACI uncertain aggregation operator is defined by the following analytical expression (see Fig. 27):

$$\llbracket CI(\lambda)\rrbracket = \begin{cases} \llbracket [0.9+6.25\lambda, 3.35+3.2\lambda], [9.2-2.05\lambda, 11.5-4.7\lambda]\rrbracket; \text{if: } 0 \le \lambda \le 0.5 \\ \llbracket [1.5+5.05\lambda, 3.35+3.20\lambda], [9.8-3.25\lambda, 11.75-5.2\lambda]\rrbracket; \text{if: } 0.5 < \lambda \le 1 \end{cases}$$

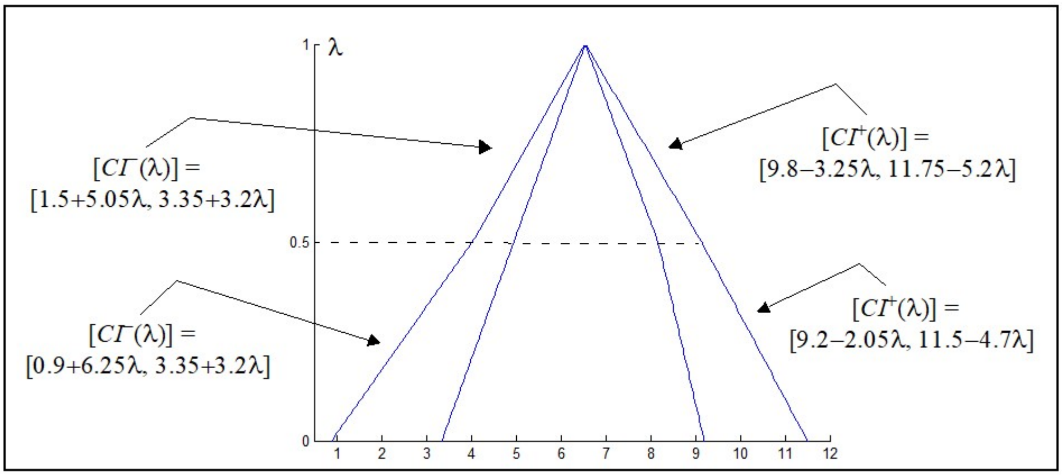


Fig. 27: TGI 2-ACI result

All remarks and advantages discussed in the illustrative example (refer to section VI) remain valid in this application example.

In the 2-ACI computation, the multiplication of a GI by a scalar is implemented by directly extending the multiplication of a CI by a scalar. At the same time, the multiplication of a TGI by a scalar is implemented by directly extending the multiplication of a GI by a scalar:

$$\omega\times[a(\lambda)] = \begin{cases} [\omega\cdot a^-(\lambda), \omega\cdot a^+(\lambda)];\ \text{if: } \omega \ge 0 \\ [\omega\cdot a^+(\lambda), \omega\cdot a^-(\lambda)];\ \text{if: } \omega < 0 \end{cases};\ \omega\times\llbracket a(\lambda)\rrbracket = \begin{cases} \llbracket \omega\times[a^-(\lambda)], \omega\times[a^+(\lambda)]\rrbracket;\ \text{if: } \omega \ge 0 \\ \llbracket \omega\times[a^+(\lambda)], \omega\times[a^-(\lambda)]\rrbracket;\ \text{if: } \omega < 0 \end{cases}$$

VII.2. Potential use in T2FI regression

The regression problem with type-1 fuzzy data has been previously addressed from various perspectives and has been successfully solved in several applications. Type-1 fuzzy regression has often been implemented as a regression by intervals (CIs *via* the concepts of α-cuts and/or GIs). Two main regression methodologies are considered: the possibilistic methods, which were introduced by Tanaka [66], and the least squares (LS) methods, which were proposed by Diamond [22]. Since the pioneering works of Tanaka and Diamond, research on T1FI regression has expanded substantially and various extensions have been proposed [3][5][12][16][25]. In this paper, the possibilistic approach is adopted. From a practical perspective, although T1FI regression has been of substantial interest, the T2FI remains poorly explored. In the literature, few researchers have investigated this problem [2][36][46][71]. For instance, in [71], a 0-cut possibilistic type-2 fuzzy qualitative regression model is proposed. In [46], a regression model that uses T2FIs and is based on the LS approach is developed. In [36], a weighted goal programming approach for T2FI linear regression is investigated. An *h*-cut piecewise possibilistic regression approach is proposed in [2]. These regression methods are often based on the α-cut principle, where the inclusion property between the observed and the predicted data is only ensured at an *h*-level. Moreover, although most of these methods could well model the T2FI regression, the models were reduced to a subset of the points of T2FIs. These methods involve many parameters and the associated computing remains expensive and difficult to generalize to any fuzzy interval shape and/or regression model. In this paper, we demonstrate that T2FI regression can be naturally extended from the CI and T1FI interpretations according to the TGI interpretation. All the T1FI regression methods (linear and nonlinear) can be extended to the T2FI framework, where the inputs, outputs and/or parameters can be represented by TGIs. Regarding the T2FI regression that is proposed in the literature, our extension does not depend on the model form, ensures the inclusion property and preserves the flexibility and the rigor of IA computations in the propagation of the T2FIs and their manipulation.

To demonstrate the development, the representation and the construction of a TGI model, the possibilistic regression approach is implemented on a univariate synthetic dataset with a heteroscedastic uncertainty structure [33]. In this dataset, the inputs are crisp and the outputs are symmetrical triangular T1FIs. The spread of T1FI outputs depends strongly on the input x. The data set $\{(x_i,[y_\lambda^{obs}]_i\}, i = 1,\dots, M$; where x_i is the i^{th} input and $[y_\lambda^{obs}]_i = [y_i - (1-\lambda)e_i, y_i + (1-\lambda)e_i]$ is the corresponding triangular T1FI output at x_i, is generated as follows:

$$\begin{cases} x_i = 0.02(i-1), i = 1,2,\dots,51; \;\; y_i = (2.7x_i - 0.2)^2 + 4.5 + err_i \\ e_i = 1.7\exp(-49(x_i - 0.5)^2) + 1.7x_i + 1.2 \end{cases} \tag{22}$$

The noise err_i in (22) has been drawn from a uniform distribution over the interval [−0.4, 0.4]. Due to the complexity of the data, a nonlinear model that is based on B-spline formalism is used [17][18][37] (see Appendix *C* for the B-spline mathematical formalism). This choice is also motivated by the applicability of the TGI approach regardless of the model form. Other model forms can be used in a similar manner. First, a T1FI regression approach is detailed. Next, a T2FI regression methodology is presented to justify its utility.

A. T1FI B-spline model construction

The regression objective is to find a T1FI (GI) B-spline model of the following form:

$$[y_\lambda(x)] = [y_\lambda^-(x), y_\lambda^+(x)] = \Sigma_{j=0}^{n-1} \; [c_j(\lambda)] \times B_{j,k}(x) \tag{23}$$

In (23), $[y_\lambda(x)]$ is the model output, $[c_j(\lambda)]$; $j = 0,\dots, n$-1 are the T1FI control coefficients and $B_{i,k}(x)$ are crisp basis functions. In this application, cubic splines with equidistant knots are used. Thus, assuming a set of knots, the B-spline regression problem is reduced to the estimation of the control coefficients. In this framework, the possibilistic regression aims at determining the coefficients $[c_j(\lambda)]$ such that the observed outputs $[y_\lambda^{obs}]_i$ are included in the outputs that are predicted by the model, namely, $[y_\lambda^{obs}]_i \subseteq [y_\lambda]_i$ ($[y_\lambda]_i$ denotes the model output at x_i). To ensure the satisfaction of the inclusion constraints, a B-spline model with trapezoidal control coefficients is employed (for a trapezoidal model, see [3]). The possibilistic T1FI B-spline regression corresponds to the following optimization problem under constraints (UC):

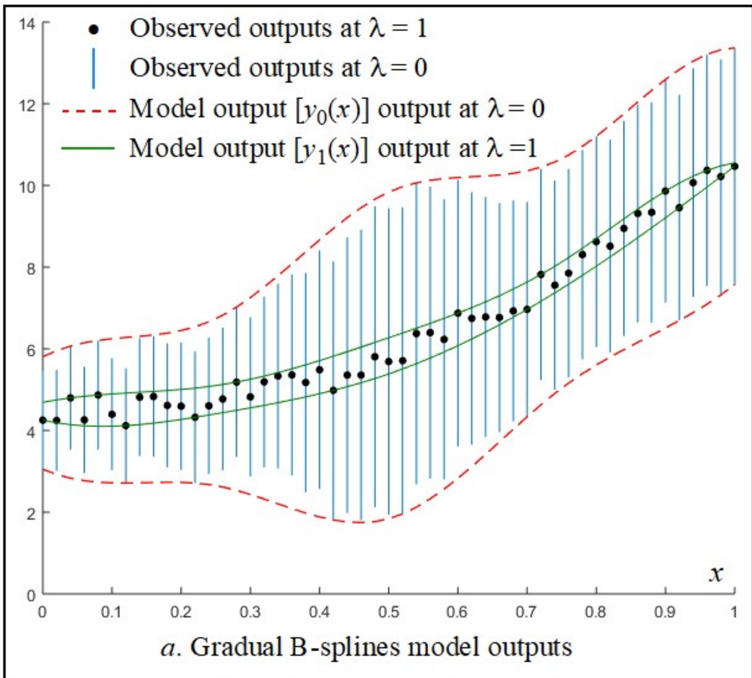
$$\underset{[c_j(\lambda)]}{\text{Min}}\, J = \sum_{i=1}^{M} R([y_\lambda(x_i)])$$
$$\text{UC.}: [y_\lambda^{obs}]_i \subseteq [y_\lambda]_i \;; i = 1, \ldots, M;\; R([c_j(\lambda)]) \geq 0 \;; j = 0, \ldots, n-1 \tag{24}$$

In (24), $R([y_\lambda(x_i)])$ represents the radius of $[y_\lambda(x_i)]$. The constraints are imposed to ensure that all observed data are included in the predicted outputs. Moreover, the identification of proper control coefficients requires the radius of each output to be positive. In the optimization problem (24), the vagueness of the model is represented by the sum of the radii of the outputs. Since the inputs are crisp, which implies that the basis splines are also crisp, the imprecision of the outputs corresponds to the imprecision of the control coefficients (the radius of $[c_j(\lambda)]$). Thus, the optimization objective is to estimate the control coefficients $[c_j(\lambda)]$; $j = 0, \ldots, n-1$ that minimize the criterion J.

As the inputs are crisp and the control coefficients have trapezoidal shapes, the model outputs are also trapezoidal T1FIs. Moreover, the trapezoidal shapes of the T1FIs enable us to express them in terms of only their levels λ=0 and λ=1. In this scenario, the optimization problem (24) is reduced to its implementation on levels λ = 0 and λ = 1. Moreover, additional inclusion constraints on level λ=1 in level λ=0 for obtaining well-defined T1FIs are necessary ($[y_1]_i \subseteq [y_0]_i, i = 1, \ldots, M$):

$$\underset{[c_j(0)],[c_j(1)]}{\text{Min}}\, J = \sum_{i=1}^{M} R([y_0(x_i)]) + R([y_1(x_i)])$$
$$\text{UC.}: [y_0^{obs}]_i \subseteq [y_0]_i \;;\; [y_1^{obs}]_i \subseteq [y_1]_i;\; [y_1]_i \subseteq [y_0]_i;\; i = 1, \ldots, M; \tag{25}$$
$$R([c_j(0)]) \geq 0 \;;\; R([c_j(1)] \;; j = 0, \ldots, n-1$$

In the implementation of (25), the number of control intervals is set to n = 7 based on the Akaike information criterion (AIC), which was proposed in [34]. The regression method led to a B-spline model of the form (23), in which the control coefficients $[c_j(\lambda)]$, j = 0, …, 6 are T1FIs. The model outputs are illustrated in Fig. 28.



a. Gradual B-splines model outputs

Method	Vagueness
Tanaka'sapproach (Poly $d = 3$)	8.2801
Hong's approach (RBF kernel)	6.4890
Hao's SVM appraoch (RBF kernel)	3.0642
Proposed approach (B-splines)	2.8015

b. Comparison results

Fig. 28: T1FI (GI) B-spline model and comparative results

For example, when considering the control coefficient $[c_3(\lambda)]$, the optimization procedure leads to the following CIs: $[c_3(0)]$ = [3.36, 5.60] (at λ = 0) and $[c_3(1)]$ = [4.39, 4.82] (at λ =1). Via linear interpolation between levels 0 and 1, the T1FI control coefficient $[c_3(\lambda)]$ is obtained (see Fig. 29.a):

$$[c_3(\lambda)] = [3.36 + 1.03\lambda, 5.60 - 0.78\lambda]$$

At each input x, the T1FI output $[y_\lambda(x)]$ can be expressed analytically via (23). For instance, the T1FI output at input x = 0.28 is illustrated in Fig. 29.b. and expressed as follows:

$$[y_\lambda(x=0.28)] = \Sigma_{j=0}^{6}\ [c_i(\lambda)] \times B_{j,k}(0.28) = [2.53+1.96\lambda, 7.02-1.84\lambda]$$

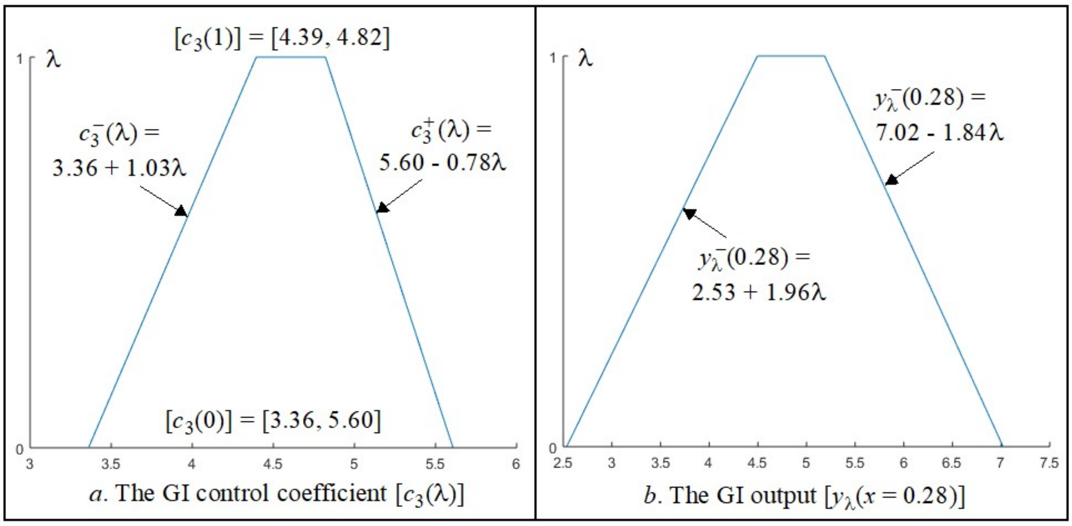


Fig. 29: T1FIs (GIs) for the control coefficient $[c_3(\lambda)]$ and the output $[y_\lambda(0.28)]$

The proposed regression approach has been compared with Tanaka's [65], Hong's [35] and Hao's [33] T1FI regression methods according to the vagueness criterion (on level $\lambda = 0$):

$$\text{Vagueness} = (1/M).\sum_{i=1}^{M} R([y_0(x_i)])$$

According to the results (see Fig. 28.b), the obtained B-spline model is less imprecise (vague) than the models that are obtained via the other methods. Moreover, the proposed method is better equipped to fit the data and the associated imprecision. The result of Tanaka's method is less efficient because only one polynomial model of degree 3 is used. The performance of the B-spline model is explained by its use of four polynomials of degree 3 (cubic-splines) instead of a single third-order polynomial model (see Appendix *C* for additional details). Moreover, if the proposed approach is similar in its design philosophy to regression that is based on SVM, its performance is better. The proposed methodology is effective for dealing with the vagueness of the data using T1FIs. However, the uncertainty phenomenon in the T1FI representation is not captured in the model. Indeed, if the process (data generation by (22)) is repeated *P* times, each time a different model with different T1FI control coefficients is obtained. This is due to the uncertain nature of the data. For instance, the output $[y_\lambda(0.28)]$ and the control coefficient $[c_3(\lambda)]$ that are obtained by repeating the process 10 times are presented in Fig. 30 (see Table 2 for the vaguenesses of the 10 models).

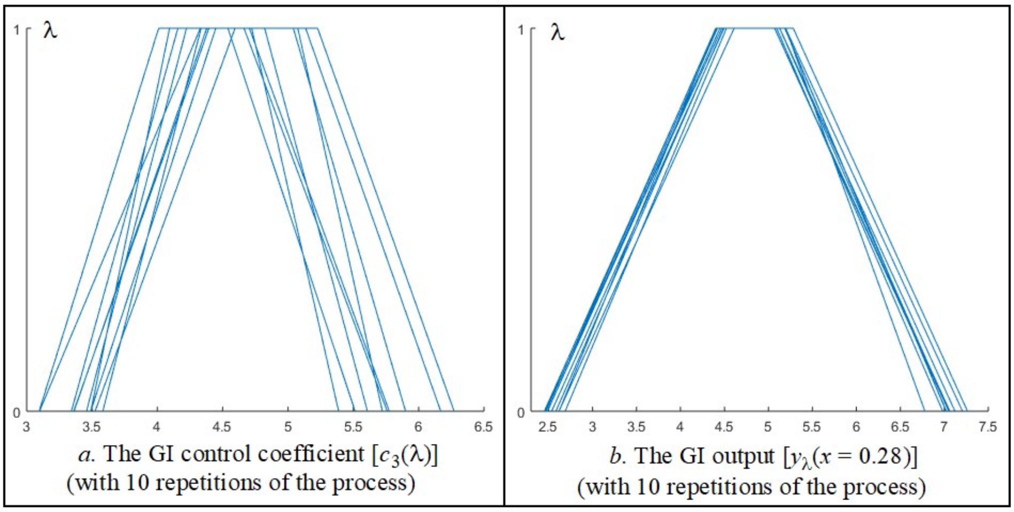


Fig. 30: T1FIs of $[c_3(\lambda)]$ and $[y_\lambda(0.28)]$ (10 repetitions)

Process	1	2	3	4	5	6	7	8	9	10
Model Vagueness	2.801	2.845	2.833	2.774	2.818	2.827	2.786	2.766	2.807	2.814

Table 2: Vagueness of the models obtained over 10 repetitions

Our current objective is to propose a possibilistic TGI (T2FI) regression approach that can integrate all the uncertain behaviors of the T1FIs into the TGI model. T2FI models provide an extensive knowledge representation compared to T1FI models.

B. TGI (T2FI) B-spline model construction

Let us consider a set of observed trapezoidal T1FIs that are obtained by model (23) through P process repetitions, namely, $[y_\lambda^{obs}]_{(i,p)}$, i = 1, …, M; p = 1, …, P (for each input x_i, P outputs are considered). By analogy with (23), the regression objective is to construct a TGI B-spline model of the following form:

$$[\![y_\lambda(x)]\!] = [\![[y_\lambda^-(x)],[y_\lambda^+(x)]]\!] = \Sigma_{j=0}^{n-1} \; [\![c_j(\lambda)]\!] \times B_{j,k}(x) \qquad (26)$$

In (26), $[\![y_\lambda(x)]\!]$ is the TGI output and $[\![c_j(\lambda)]\!]$ denotes the TGI control coefficients. Similar to the T21FI possibilistic regression, the TGI regression can be reduced to an estimation problem of the TGI control coefficients such that all the observed data will be encapsulated in the TGI model. To satisfy the inclusion constraints, a trapezoidal B-spline model with trapezoidal TGI control coefficients is employed. Similar to the T1FI regression, the optimization problem is considered only at levels λ=0 and λ=1. In this context, for a specified set of observed trapezoidal T1FIs, the TGI regression is expressed as the following optimization problem under constraints:

$$\underset{[\![c_j(0)]\!],\,[\![c_j(1)]\!]}{\text{Min}\, J} = \sum_{i=1}^{M} R([y_0^-(x_i)]) + R([y_0^+(x_i)]) + R([y_1^-(x_i)]) + R([y_1^+(x_i)])$$

$$\text{UC.}: [y_1^{obs}]_{(i,p)} \subseteq [\![y_1]\!]_i \;\; ; \; [y_0^{obs}]_{(i,p)} \subseteq [\![y_0]\!]_i \, ; \; i = 1,\dots,M; \, p = 1,\dots,P;$$

$$[c_j^-(0)] \leqq [c_j^+(0)]; \; [c_j^-(1)] \lneqq [c_j^+(1)]; \; j = 0,\dots,n-1; \qquad (27)$$

$$R([c_j^-(0)]) \geq 0 \; ; R([c_j^-(1)]) \geq; R([c_j^+(0)]) \geq 0; R([c_j^+(1)]) \geq 0; j = 0,\dots,n-1$$

For two intervals $[a] = [a^-, a^+]$ and $[b] = [b^-, b^+]$, the relations ≦ and ≨ are defined as follows:

$$[a] \leqq [b] \Leftrightarrow a^+ \leq b^- \text{ and } [a] \lneqq [b] \Leftrightarrow a^+ < b^-$$

and the relation ⊆ is interpreted as follows:

$$[a] = [a^-, a^+] \subseteq [\![a]\!] = [\![[a^-], [a^+]]\!] \Leftrightarrow a^- \in [a^-] \text{ and } a^+ \in [a^+]$$

In the optimization problem (27), the criterion considers the sum of the radii of the TGI left and right profiles at levels 0 and 1. The constraints $[y_1^{obs}]_{(i,p)} \subseteq [\![y_1]\!]_i$; $[y_0^{obs}]_{(i,p)} \subseteq [\![y_0]\!]_i$ ensure the inclusion of the observed data in the TGI model outputs. The constraints $[c_j^-(0)] \leqq [c_j^+(0)]$ and $[c_j^-(1)] \lneqq [c_j^+(1)]$ are used to ensure well-defined TGI control coefficients. Indeed, $[c_j^-(0)] \leqq [c_j^+(0)]$ guarantees that $[c_j^-(0)]$ is always before $[c_j^+(0)]$ with no intersection between them and $[c_j^-(1)] \lneqq [c_j^+(1)]$ ensures that $[c_j^-(1)]$ is always before $[c_j^+(1)]$ with a possible meeting intersection between them. The positivity of the radii of the left and right profiles of the TGI control coefficients at levels 0 and 1 is also required.

Similar to before, a B-spline model with 7 control coefficients is employed. The proposed regression method led to a TGI B-spline model of the form (26), in which the coefficients $[\![c_j(\lambda)]\!]$, j= 0, …, 6 are TGIs. For example, the optimization problem yields the following control coefficient $[\![c_3(\lambda)]\!]$:

$$[\![[c_3(0)]]\!] = [\![[c_3^-(0)],[c_3^+(0)]]\!] = [\![[3.26, 3.43],[5.33, 6.01]]\!] : \text{at the level } \lambda = 0$$

$$[\![[c_3(1)]]\!] = [\![[c_3^-(1)],[c_3^+(1)]]\!] = [\![[4.16, 4.45],[4.88, 5.09]]\!] : \text{at the level } \lambda = 0$$

A linear interpolation between levels 0 and 1 yields the TGI control coefficient $[c_3(\lambda)]$, which is illustrated in Fig. 31 and expressed as follows:

$$[\![[c_3(\lambda)]]\!] = [\![[c_3^-(\lambda)],[c_3^+(\lambda)]]\!] = [\![[3.26 + 0.9\lambda, 3.43 + 1.02\lambda],[5.33 - 0.45\lambda, 6.01 - 0.92\lambda]]\!]$$

For each input, the TGI output can be expressed analytically by the model (26). For instance, the output at x = 0.28 corresponds to the following expression (see Fig. 32):

$$[\![y_\lambda(x=0.28)]\!] = [\![[y_\lambda^-(0.28)], [y_\lambda^+(0.28)]]\!] = \sum_{j=0}^{n-1} [\![c_j(\lambda)]\!] \times B_{j,k}(0.28)$$
$$= [\![[2.42+1.93\lambda, 2.69+1.92\lambda], [6.78-1.74\lambda, 7.28-1.96\lambda]]\!]$$

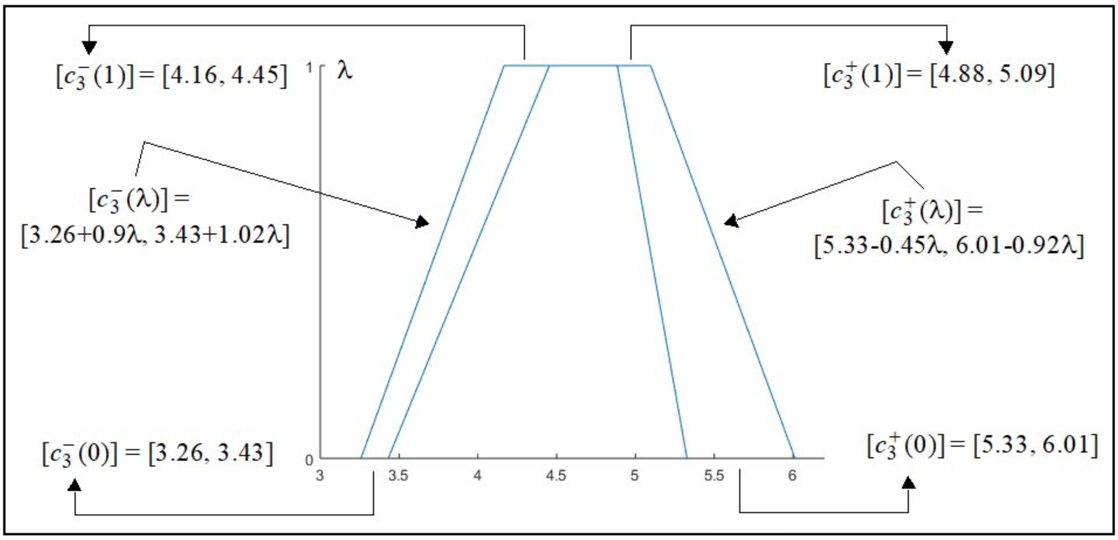


Fig. 31: TGI control coefficient $[\![c_3(\lambda)]\!]$

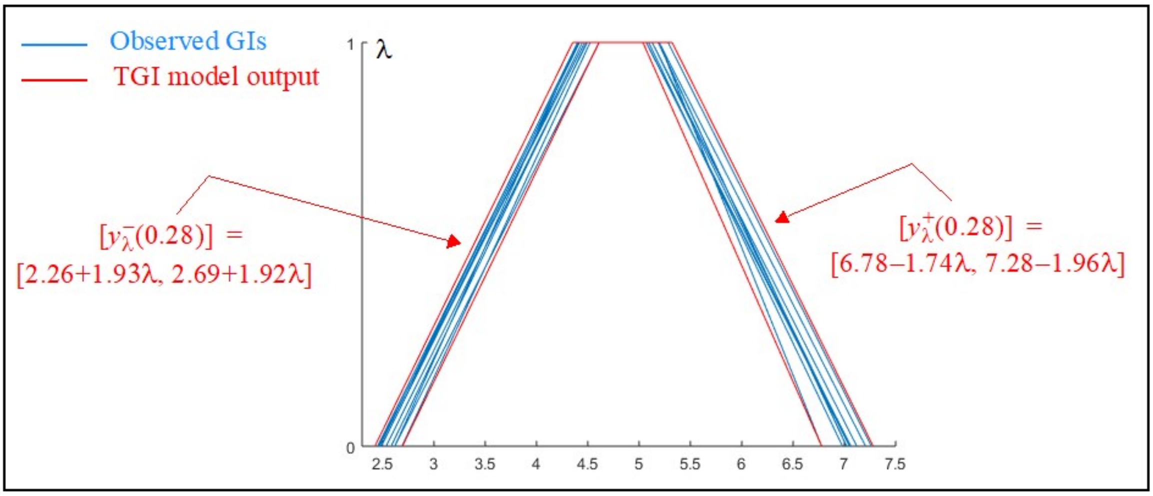


Fig. 32: TGI model output $[\![y_\lambda(0.28)]\!]$

According to Fig. 32, all the observed T1FIs are included in the TGI model.

C. Remarks and discussion

• In the proposed strategy, as the B-spline model is a linear combination of the basis functions and the control coefficients, the criterion expression J is linear with respect to the control intervals. In this context of B-spline regression, the originality of the proposed approach is that the optimization methodology is regarded as a conventional LP optimization problem, as is often the case in the possibilistic T1FI regression literature. In this paper, the interior-point method of MATLAB-R2017a from MathWorks is employed.

• The TGI regression has been implemented using trapezoidal T1FIs and TGIs, where the optimization computations are performed only on levels 0 and 1. Moreover, this methodology remains valid and can be adapted to any shapes of the considered T1FIs and TGIs. However, for a nonlinear TGI shape, the proposed method will be implemented as an interval or TI regression by discretizing the vertical dimension λ. In this scenario, the optimization must balance the computation time and the quality of the approximation.

• The proposed method has been developed in a possibilistic framework; however, it can also be naturally applied in LS-based regression. In this framework, the optimization criterion can be expressed in terms of distances between intervals and the inclusion constraints between the observed data and the predicted data must be relaxed. Furthermore, although the regression method has been demonstrated using crisp inputs, it can be adapted for dealing with fuzzy inputs.

- For illustrative purposes and to explain the interest in and the relevance of TGI regression, the approach was presented initially for T1FIs and subsequently for T2FIs. However, the method can be applied directly to determine the TGI (T2FI) model without going through a T1FI model.

VII. Conclusions

In this paper, a new interpretation of T2FIs is proposed. In this interpretation, a T2FI is regarded as a TGI. This interpretation enables the extension of SIA for computing with T2FIs. This extension, which preserves the flexibility of interval arithmetic and reasoning as major objectives, yields a revision and a new interpretation of the type-2 fuzzy arithmetic according to the concept of TGIs. The obtained results demonstrate the relevance and the applicability of the proposed strategy, via which analytic expressions are obtained. The proposed method has been demonstrated using linear T2FIs; however, it can be applied to any analytical form of the TGIs (see the example in Appendix *B* where the T2FIs have nonlinear shapes). Potential applications of our approach have been demonstrated in the frameworks of type-2 aggregation operators and type-2 fuzzy regression. Many other potential uses of the TGI approach can be envisioned in type-2 fuzzy modeling, type-2 fuzzy control applications and type-2 fuzzy decision-making strategies. For instance, the inverse model control strategy that is proposed in [7] for GIs can be naturally extended to TGIs. Furthermore, the proposed formalism is useful for solving type-2 fuzzy equations. For example, TGI equations $[\![a(\lambda)]\!]=[\![b(\lambda)]\!]+[\![x(\lambda)]\!]$ and $[\![a(\lambda)]\!]=[\![b(\lambda)]\!]\times[\![x(\lambda)]\!]$ can be regarded as new formulations of the T2FI equations. These new formulations enable us to solve these type-2 equations analytically, namely, $[\![x(\lambda)]\!]=[\![a(\lambda)]\!]-[\![x(\lambda)]\!]$ and $[\![x(\lambda)]\!]=[\![b(\lambda)]\!]\div[\![x(\lambda)]\!]$. In this paper, all the fuzzy computations are performed on normal T2FIs. This approach can be generalized through the concept of thick gradual sets [20]. Future papers will be devoted to these interesting research directions.

VII. References

[1] Ashraf Z., Muhuri P-K., and Roy M.L., Type-2 fuzzy reliability–redundancy allocation problem and its solution using particle-swarm optimization algorithm, Granular Comput. Vol. 4 (2) , pp 145–166, 2019.

[2] Bajestani N-S., Kamyad A-V. and Zare A., A Piecewise Type-2 Fuzzy Regression Model, Int. J. of Comput. Intelligence Systems, Vol. 10, pp. 734–744, 2017.

[3] Bisserier A., Boukezzoula R., Galichet S., A Revisited Approach for Linear Fuzzy Regression Using Trapezoidal Fuzzy Intervals, Information Sciences, Vol. 180 (19), pp. 3653-3673, 2010.

[4] Bodjanova S., Alpha-bounds of fuzzy numbers, Inform. Sci., Vol. 152, pp. 237-266, 2003.

[5] Boukezzoula R., Galichet S. and Coquin D., From fuzzy regression to gradual regression: Interval-based analysis and extensions, Inform. Sci., Vol. 441, pp. 18-40, 2018.

[6] Boukezzoula R., Galichet S., Foulloy L., Min and Max Operators for Gradual intervals, IEEE Trans. on Fuzzy Systems, Vol. 26 (6), pp. 3569-3578, 2018.

[7] Boukezzoula R., Foulloy L., Galichet S., Model Inversion Using Extended Gradual Intervals Arithmetic, IEEE Trans. on Fuzzy Syst. 2012, Vol. 1, pp. 82-95, 2012.

[8] Boukezzoula R., Galichet S., Foulloy L., Elmasry M., Extended gradual interval (EGI) arithmetic and its application to gradual weighted averages, Fuzzy Sets and Systems, Vol. 257, pp. 67-84, 2014.

[9] Boukezzoula R., Galichet S., Foulloy L., MIN and MAX Operators for Fuzzy Intervals and their Potential Use in Aggregation Operators, IEEE Trans. on Fuzzy Systems, Vol. 15 (6), pp. 1135-1144, 2007.

[10] Castillo O., Cervantes L., Soria J., Sanchez M. and Castro J-R., Generalized type-2 fuzzy granular approach with applications to aerospace, Information Sciences, Vol. 354, pp. 165–177, 2016.

[11] Castillo O., Amador-Angulo L., Castro J-R. and Garcia-Valdez M., A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems, Information Sciences, Vol. 354, pp. 257-274, 2016.

[12] Cerny M. and Hladik M., Possibilistic linear regression with fuzzy data: Tolerance approach with prior information, Fuzzy Sets & Syst., Vol. 304, pp. 127-144, 2018.

[13] Cervantes L., Oscar Castillo O., Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control, Information Sciences, Vol. 324, pp. 247–256, 2015.
[14] Chen, T. Y, An ELECTRE-based outranking method for multiple criteria group decision making using interval type-2 fuzzy sets, Information Sciences, Vol. 263 , pp. 1–21, 2014.
[15] Chabert, G., and Jaulin, L., A Priori Error Analysis with Intervals. SIAM Journal on Scientific Computing, Vol. 31(3), pp. 2214–2230, 2009.
[16] Coppi R., D'Urso P., Giordani P., Santoro A., Least squares estimation of a linear regression model with LR fuzzy response, Computational Stat. & Data Analysis, Vol. 51 (1), pp. 267-286, 2006.
[17] De Boor C., On calculating with B-splines, J. Approx. Theory, Vol. 6, pp. 50–62, 1972.
[18] De Boor C., A Practical Guide to Splines, Springer-Verlag, N-Y, 1978.
[19] Deschrijver G., Kerre E., Aggregation operation in interval-valued fuzzy and Atanassov's intuitionistic fuzzy set theory, in: H. Bustince, et al. (Eds.), Fuzzy Sets and Their Extensions: Representation, Aggregation and Models, Springer, Berlin, pp.183–203, 2008.
[20] Desrochers B., and Jaulin L., Thick set inversion, Artificial Intelligence, Vol. 249, pp. 1-18, 2017.
[21] Desrochers B., Simultaneous Localization and Mapping in Unstructured Environments, A Set-Membership Approach, PhD thesis, ENSTA Bretagne, France, 2018.
[22] Diamond P., Fuzzy Least Squares, Inform. Sci., Vol. 46, pp. 141-157, 1988.
[23] Dubois D. and Prade H., Gradual elements in a fuzzy set, Soft Computing, No. 12, pp. 165-175, 2008.
[24] Dubois D. and Prade H., Operations on fuzzy numbers, Int. J. Syst. Sci. Vol. 9 (6), pp. 613–626, 1978.
[25] D'Urso P. and Gastaldi T., A least-squares approach to fuzzy linear regression analysis, Com. Stat. & D. Anal., Vol. 34, pp. 427-440, 2000.
[26] Fortin J., Dubois D. and Fargier H., Gradual numbers and their application to fuzzy interval analysis, IEEE Trans. on Fuzzy Systems, Vol. 16 (2), pp. 388-402, 2008.
[27] Golsefid, S. M. M., Zarandi, M. H. F., and Turksen, I. B., Multi-central general type-2 fuzzy clustering approach for pattern recognition, Information Sciences, Vol. 328, pp. 172–188, 2016.
[28] Guerra M.L., and Stefanini L., Approximate fuzzy arithmetic operations using monotonic interpolations, Fuzzy Sets and Systems, Vol. 150, pp. 5-33, 2005.
[29] Hamrawi H., and Coupland S., Type-2 Fuzzy Arithmetic using Alpha-planes, Proceeding of the IFSA-EUSFLAT Conference, pp. 606-611, 2009.
[30] Hamrawi H., Coupland S., and John R., A Novel Alpha-cut Representation for Type-2 Fuzzy Sets, Proc. of the WCCI 2010 IEEE World Congress on Computational Intelligence, pp. 351-358, 2010.
[31] Hamrawi H., Coupland S., and John R., Parallel computation of type-2 fuzzy sets using alpha-cuts, In the Proc. of the WCCI-FUZZ-IEEE Conf., 2012.
[32] Hamrawi, S. Coupland, and R. John, Type-2 Fuzzy Alpha-Cuts, IEEE Transactions on Fuzzy systems, Vol. 25 (3), pp. 682-692, 2017.
[33] Hao P-Y., Dual possibilistic regression analysis using support vector networks, Fuzzy Sets and Systems, doi.org/10.1016/j.fss.2019.03.012, Accepted in March 2019.
[34] Harmening C., and Neuner H., Choosing the Optimal Number of B-spline Control Points (Part 1: Methodology and Approximation of Curves), Journal of Appl. Geodesy, Vol. 10 (3), pp.139-157, 2016.
[35] Hong D.H., Hwang C., Interval regression analysis using quadratic loss support vector machine, IEEE Trans. Fuzzy Systems Vol. 13 (2), pp. 229–237, 2005.
[36] Hosseinzadeh E., Hassanpour H., Arefi M., A weighted goal programming approach to fuzzy linear regression with crisp inputs and type-2 fuzzy outputs, Soft Comput. Vol. 19 (5), pp. 1143-1151, 2014.
[37] Lin H., Chen W., Wang G., Curve reconstruction based on an interval B-spline curve, Visual computer, Vol. 21, pp. 418-427, 2005.
[38] Melin, P., Gonzalez, C. I., Castro, J. R., Mendoza, O., and Castillo, O., Edge-detection method for image processing based on generalized type-2 fuzzy logic, IEEE Trans. on Fuzzy Systems, Vol. 22 (6), pp. 1515–1525, 2014.
[39] Mendel J.M., Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice Hall, Upper Saddle River, NJ, 2001.
[40] Mendel J.M., John R.I. and Liu F., Interval type-2 fuzzy logic systems made simple, IEEE Trans. Fuzzy Syst. Vol. 14 (6), pp. 808-821, 2006.
[41] Mendel J.M., Liu F., and Zhai D., Alpha-Plane Representation for Type-2 Fuzzy Sets: Theory and Applications, IEEE Transactions on Fuzzy Systems, Vol. 17 (5), pp. 1189-1207, 2009.

[42] Meyer P. and Roubens M., On the use of the Choquet integral with fuzzy numbers in multiple criteria decision support," Fuzzy Sets & Systems, Vol. 157 (7), pp. 927–938, 2006.
[43] Moore R. E., Interval Arithmetic and Automatic Error Analysis in Digital Computing, PhD Thesis, Department of Computer Science, Stanford University, 1962.
[44] Moore R. E., Interval Analysis, Prentice-Hall, Englewood Cliffs, N.J., 1966.
[45] Piegat A. and Landowski M., Is an Interval The Right Result of Arithmetic Operations on Intervals?, Int. J. Appl. Math. Comput. Sci., Vol. 27 (3), pp. 575–590, 2017.
[46] Poleshchuk O., E.Komarov E., A Fuzzy Nonlinear Regression Model for Interval Type-2 Fuzzy Sets, Int. J. of Mathematical and Comput. Sciences, Vol. 8 (6), pp. 840-844, 2014.
[47] Kaucher E., Interval Analysis in the Extended Interval Space IR, Comp. Suppl., Vol. 2 pp. 33–49, 1980.
[48] Kaufmann A. and Gupta M.M., Introduction to fuzzy arithmetic: Theory and Applications, Van Nostrand Reinhold Company Inc. New York, 1991.
[49] Kulpa Z., Diagrammatic representation for interval arithmetic, Linear Algebra and its Applications, Vol. 324, (1-3), pp. 55-80, 2001.
[50] Kulpa Z., A diagrammatic approach to investigate interval relations, J. Vis. Lang. Comput., Vol. 17 (5), pp. 466-502, 2006.
[51] Liu F., An efficient centroid type reduction strategy for general type-2 fuzzy logic system, Inf. Sci., Vol. 178, pp. 2224-2236, 2008.
[52] Liu X., Mendel J. M., and Wu D., Analytical solution methods for the fuzzy weighted average, Information Sciences, Vol. 187, pp. 151–170, 2012.
[53] Liu P., A weighted aggregation operators multi-attribute group decision-making method based on interval-valued trapezoidal fuzzy numbers, Expert Sys. with Appl., Vol. 38, (1), pp. 1053–1060, 2011.
[54] Liu P., and Jin F., A multi-attribute group decision-making method based on weighted geometric aggregation operators of interval-valued trapezoidal fuzzy numbers, Applied Mathematical Modelling, vol. 36 (6), pp. 2498–2509, 2012.
[55] Ontiveros-Robles E., Melin P. and Castillo O., Comparative analysis of Noise Robustness of Type 2 Fuzzy Logic Controllers, Kybernetika, Vol. 54 (1), pp. 175–201, 2018.
[56] Sanchez M-A., Castillo O. and Castro J-R., a Generalized Type-2 Fuzzy Systems for controlling a mobile robot and a performance comparison with Interval Type-2 and Type-1 Fuzzy Systems, Expert Systems with Applications, Vol. 42, pp. 5904–5914, 2015.
[57] Shukla A.K., and Muhuri P.K., Big-data clustering with interval type-2 fuzzy uncertainty modeling in gene expression datasets, Engineering Applications of Artificial Intelligence, Vol. 77, pp. 268-282, 2019.
[58] Shukla A.K., Seth T., Muhuri P. K., Interval type-2 fuzzy sets for enhanced learning in deep belief networks, IEEE Int. Conf. on Fuzzy Systems, FUZZ-IEEE, 2017.
[59] Stefanini L., A generalization of Hukuhara difference and division for interval and fuzzy arithmetic, Fuzzy Sets and Systems, Vol. 161, pp. 1564-1584, 2010.
[60] Stefanini L., and Sorini L., Approximation of fuzzy numbers by F-transform, Proc. IPMU Conf., 2012.
[61] Stefanini L., and Sorini L., Type-2 fuzzy numbers and operations by F-transform, Proc. of Joint IFSA World Congress IFSA/NAFIPS, 2013.
[62] Stefanini L., A generalization of Hukuhara difference and division for interval and fuzzy arithmetic, Fuzzy Sets and Systems, Vol. 161, pp. 1564-1584, 2010.
[63] Stephen Dinagar D., and Anbalagan A., A new Type-2 Fuzzy number arithmetic using extension principle, IEEE Int. Conf. On Adv. In Eng., Science and Mana. (ICAESM -2012), pp. 113-118, 2012.
[64] Sunaga T., Theory of an Interval Algebra and its application to Numerical Analysis, RAAG Memories 2, pp. 547-564, 1958.
[65] Tanaka H. and Lee H., Interval regression analysis by quadratic programming approach, IEEE Trans. Fuzzy Syst. Vol 6 (4), pp. 473–481, 1998.
[66] Tanaka H., Uejima S. and Asai K., Linear regression analysis with fuzzy model, IEEE Trans. SMC-2, pp. 903-907, 1982.
[67] Takác Z., Aggregation of fuzzy truth values, Information Sciences, Vol. 271, pp. 1–13, 2014.
[68] Torres-Blanc C., Cubillo S. and Hernández P., Aggregation operators on type-2 fuzzy sets, Fuzzy Sets & Systems, Vol. 324, pp. 74–90, 2017.
[69] Wagner C., and Hagras H., zSlices–Towards bridging the gap between Interval and General Type-2 Fuzzy Logic, in Proc. Of Fuzz-IEEE Conf., pp. 489-497, 2008.
[70] Warmus M., Calculus of Appoximations, Bull. Acad. Polon. Science, C1. III, Vol. IV, pp. 253-259, 1956.

[71] Wei Y., Watada J., Building a Type II Fuzzy Qualitative Regression, Intell. Decision Technol. Vol. 1 (15), pp. 145-154, 2012.
[72] Wu D., and Mendel J.M., Corrections to Aggregation Using the Linguistic Weighted Average and Interval Type-2 Fuzzy Sets, IEEE Transactions on Fuzzy Systems, Vol. 16 (6), pp. 1664–1666, 2008.
[73] Wu D. and Mendel J. M., Aggregation Using the Linguistic Weighted Average and Interval Type-2 Fuzzy Sets, IEEE Transactions on Fuzzy Systems, Vol. 15 (6), pp. 1145-1161, 2007.
[74] Zadeh L. A., Fuzzy sets, Infor. and Control, Vol. 8, pp. 338-353, 1965.
[75] Zadeh L.A., The concept of a linguistic variable and its application to approximate reasoning, Inf. Sci., Vol. 8, pp. 199–250, pp. 301–357, pp. 9:42–80, 1975.
[76] Zhou S.M., John R. I., Chiclana F., and Garibaldi J. M., On aggregating uncertain information by type-2 OWA operators for soft decision making," Inter. J. of Intelligent Syst., Vol. 25 (6), pp. 540–558, 2010.

APPENDIX *A*: Computational example over TIs

Let us consider the two TIs $[\![a]\!]$ and $[\![b]\!]$, which are defined as follows:

$$[\![a]\!]=[\![[a^-], [a^+]]\!] = [\![[-2, 1], [3, 5]]\!] \text{ and } [\![b]\!]=[\![[b^-], [b^+]]\!] = [\![[1, 2], [4, 6]]\!]$$

The thick operations (7)-(10) are expressed as follows:

- Addition: $[\![a]\!]+[\![b]\!] = [\![[-2, 1]+[1, 2], [3, 5]+[4, 6]]\!] = [\![[-1, 3], [7, 11]]\!]$
- Subtraction: $[\![a]\!]-[\![b]\!] = [\![[-2, 1]-[4, 6], [3, 5]-[1, 2]]\!] = [\![[-8, -3], [1, 4]]\!]$
- Multiplication: $[\![a]\!]\times[\![b]\!] = [\![min(\Phi_1), max(\Phi_1)]\!]$; where:

  $\Phi_1 = \{[-2, 1]\times[1, 2], [-2, 1]\times[4, 6], [3, 5]\times[1, 2], [3, 5]\times[4, 6]\} = \{[-4, 2], [-12, 6], [3, 10], [12, 30]\}$

  Let us compute the result of $min(\Phi_1)$:

  $min(\Phi_1) = min\{[-4, 2], [-12, 6], [3, 10], [12, 30]\}= min(min([-4, 2], [-12, 6]), min([3, 10], [12, 30]))$

  In this scenario,

  $$min([3, 10], [12, 30]) = [3, 10] \text{ and } min([-4, 2], [-12, 6]) = [-12, -4]$$

  Finally,

  $$min(\Phi_1) = min([3, 10], [-12, -4]) = [-12, -4]$$

  The same principle is applied to compute $max(\Phi_1)$, thereby leading to the final multiplication result:

  $$[\![a]\!]\times[\![b]\!] = [\![min(\Phi_1), max(\Phi_1)]\!] = [\![[-12, -4], [12, 30]]\!]$$

- Division: $[\![a]\!]\div[\![b]\!] = [\![[-2, 1], [3, 5]]\!]\times[\![1\div[4, 6], 1\div[1, 2]]\!]$

  $= [\![[-2, 1], [3, 5]]\!]\times[\![[1/6, 1/4], [1/2, 1]]\!] = [\![min(\Phi_2), max(\Phi_2)]\!]$; where:

  $$\Phi_2 = \{[-2, 1]\times[1/6, 1/4], [-2, 1]\times[1/2, 1], [3, 5]\times[1/6, 1/4], [3, 5]\times[1/2, 1]\}$$
  $$= \{[-1/2, 1/4], [-2, 1], [1/2, 5/4], [3/2, 5]\}$$

  The computation of $min(\Phi_2)$ and $max(\Phi_2)$ leads to the following:

  $$min(\Phi_2) = min(min([-1/2, 1/4], [-2, 1]), min([1/2, 5/4], [3/2, 5])\})$$
  $$= min([-2, -1/3], [1/2, 5/4]) = [-2, -1/3]$$
  $$max(\Phi_2) = max(max([-1/2, 1/4], [-2, 1]), max([1/2, 5/4], [3/2, 5]))$$
  $$= max([1/4, 1], [3/2, 5]) = [3/2, 5]$$

Finally, the division operation is expressed as follows:

$$[\![a]\!]\div[\![b]\!] = [\![[-2, -1/3], [3/2, 5]]\!]$$

APPENDIX *B*: Computational example using that uses TGIs with nonlinear shapes

Let us consider two TGIs $[\![a(\lambda)]\!]$ and $[\![b(\lambda)]\!]$, which are illustrated in Fig. 33 and expressed as follows:

$$[\![a(\lambda)]\!] = [\![[a^-(\lambda)], [a^+(\lambda)]]\!] = [\![[10-5.e^{-3\lambda}, 10-2.\ e^{-4\lambda}], [10+2.e^{-4\lambda}, 10+5.e^{-3\lambda}]]\!]; \text{ and}$$

$$[\![b(\lambda)]\!] = [\![[b^-(\lambda)], [b^+(\lambda)]]\!] = [\![[6-2.e^{-3\lambda}, 6-e^{-4\lambda}], [6+e^{-4\lambda}, 6+2.e^{-3\lambda}]]\!]$$

The SIA operations over the TGIs are defined by the following expressions and illustrated in Fig. 34:

- Addition: $[\![a(\lambda)]\!]+[\![b(\lambda)]\!] = [\![[a^-(\lambda)]+[b^-(\lambda)], [a^+(\lambda)]+[b^+(\lambda)]]\!]$

$$= [\![[16-7.e^{-3\lambda}, 16-3.e^{-4\lambda}], [16+3.e^{-4\lambda}, 16+7.e^{-3\lambda}]]\!]$$

- Subtraction: $[\![a(\lambda)]\!]-[\![b(\lambda)]\!] = [\![[a^{-}(\lambda)]-[b^{+}(\lambda)], [a^{+}(\lambda)]-[b^{-}(\lambda)]]\!]$

$$= [\![[4-7.e^{-3\lambda}, 4-3.e^{-4\lambda}]+[4+3.e^{-4\lambda}, 4+7.e^{-3\lambda}]]\!]$$

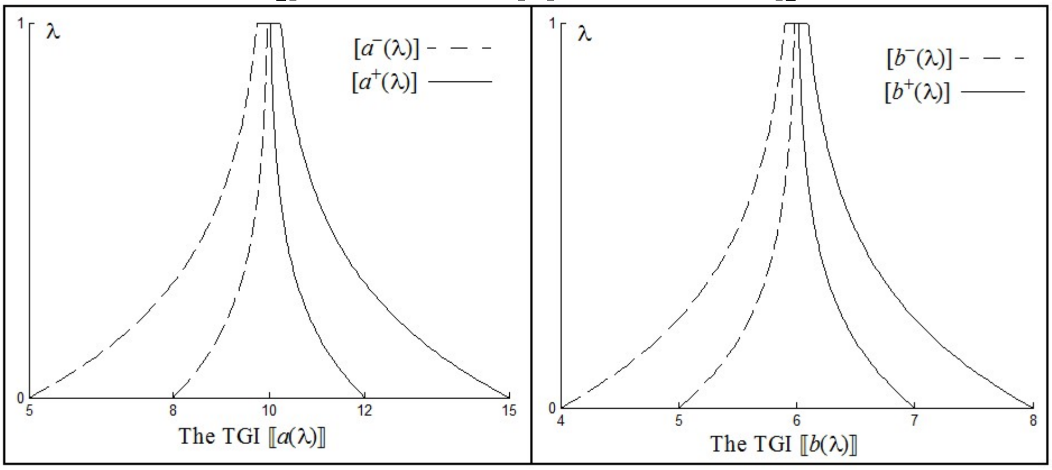


Fig. 33: Two TGIs $[\![a(\lambda)]\!]$ and $[\![b(\lambda)]\!]$

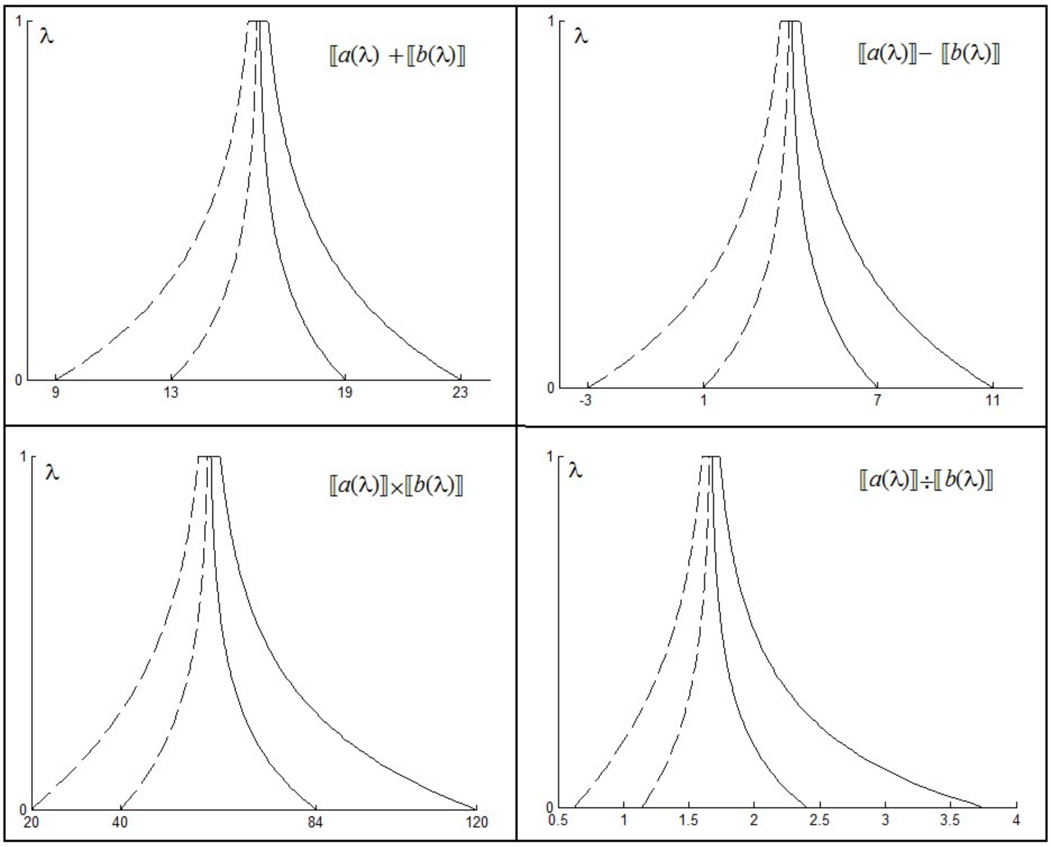


Fig. 34: Four arithmetic operations over TGIs $[\![a(\lambda)]\!]$ and $[\![b(\lambda)]\!]$

- Multiplication: $[\![a(\lambda)]\!]\times[\![b(\lambda)]\!] = [\![min(\Phi_1(\lambda)), max(\Phi_1(\lambda))]\!]$; where:

$$\Phi_1(\lambda) = \{[c_1(\lambda)], [c_2(\lambda)], [c_3(\lambda)], [c_4(\lambda)]\};$$

$$[c_1(\lambda)] = [a^{-}(\lambda)]\times[b^{-}(\lambda)] = [(10-5.e^{-3\lambda})(6-2.e^{-3\lambda}), (10-2.\,e^{-4\lambda})(6-e^{-4\lambda})]$$

$$[c_2(\lambda)] = [a^{-}(\lambda)]\times[b^{+}(\lambda)] = [(10-5.e^{-3\lambda})(6+e^{-4\lambda}), (10-2.\,e^{-4\lambda})(6+2.e^{-3\lambda})]$$

$$[c_3(\lambda)] = [a^{+}(\lambda)]\times[b^{-}(\lambda)] = [(10+2.e^{-4\lambda})(6-2.e^{-3\lambda}), (10+5.e^{-3\lambda})(6-e^{-4\lambda})]$$

$$[c_4(\lambda)] = [a^{+}(\lambda)]\times[b^{+}(\lambda)] = [(10+2.e^{-4\lambda})(6+e^{-4\lambda}), (10+5.e^{-3\lambda})(6+2.e^{-3\lambda})]$$

In this case,

$$min(\Phi_1(\lambda)) = [c_1(\lambda)];\ \text{and: } max(\Phi_1(\lambda)) = [c_4(\lambda)].$$

Finally,

$$⟦a(\lambda)⟧\times⟦b(\lambda)⟧ = ⟦[(10-5.e^{-3\lambda})(6-2.e^{-3\lambda}), (10-2.e^{-4\lambda})(6-e^{-4\lambda})], [(10+2.e^{-4\lambda})(6+e^{-4\lambda}), (10+5.e^{-3\lambda})(6+2.e^{-3\lambda})]⟧$$

• Division: $⟦a(\lambda)⟧\div⟦b(\lambda)⟧ = ⟦a(\lambda)⟧\times(1\div⟦b(\lambda)⟧)$

Since $1\div⟦b(\lambda)⟧ = ⟦1\div[6+e^{-4\lambda}, 6+2.e^{-3\lambda}], 1\div[6-2.e^{-3\lambda}, 6-e^{-4\lambda}]⟧$

$$= ⟦[1/(6+2.e^{-3\lambda}), 1/(6+e^{-4\lambda})], [1/(6-e^{-4\lambda}), 1/(6-2.e^{-3\lambda})]⟧$$

applying the same approach as for the multiplication operator yields

$$⟦a(\lambda)⟧\div⟦b(\lambda)⟧ = ⟦[(10-5.e^{-3\lambda})/(6+2.e^{-3\lambda}), (10-2.e^{-4\lambda})/(6+e^{-4\lambda})], [(10+2.e^{-4\lambda})/(6-e^{-4\lambda}), (10+5.e^{-3\lambda})/(6-2.e^{-3\lambda})]⟧$$

The four operations over the TGIs are illustrated in Fig. 25.

APPENDIX *C*: Brief review of the B-spline mathematical formalism

For simplicity, the mathematical formalism is presented for a univariate regression model. Let us consider a set of crisp input-output data of an unknown function. The regression objective is to identify a model of the following form:

$$y = f(x)$$

where x is the input, y is the output, and $f(x)$ is represented by a B-spline function. The B-spline formalism attempts to represent the input-output behavior with a piecewise polynomial function that interpolates the given data. Let us assume that x is contained in a finite interval domain $[x_0, x_f]$. The values $x = t_j$, $j = 0, \ldots, n+k$-1 are known as the knots, where n represents the number of control coefficients and k denotes the B-spline order. The B-spline $f(x)$ of order k is a piecewise polynomial function of degree k-1. The following non-decreasing sequence is called the knot sequence:

$$T = \{x_0 = t_0, \ldots, t_k, t_{k+1}, \ldots, t_n, \ldots, t_{n+k-1} = x_f\}$$

A knot has multiplicity r if it appears r times in the knot sequence. For obtaining clamped B-splines, the first and last knots are of multiplicity k and the sequence T becomes

$$T = \{x_0 = t_0 = \ldots = t_{k-1}, t_k, \ldots, t_{n-1}, t_n = \ldots = t_{n+k-1} = x_f\}$$

The B-spline function is defined as follows:

$$y(x) = f(x) = \Sigma_{j=0}^{n-1}\ c_j \cdot B_{j,k}(x);\ \ x \in [t_0, t_{n+k-1}]$$

The $B_{j,k}(x)$s are defined by the following recursive expression [17][18]:

$$B_{j,0}(x) = \begin{cases} 1, & \text{if } t_j \le x \le t_{j+1}, \forall j = 0, \cdots, n-1 \\ 0 & \text{otherwise} \end{cases}$$

and: $\forall j = 0, \ldots, n$-k-1: $B_{j,k}(x) = \dfrac{x - t_j}{t_{j+k} - t_j} B_{j,k-1}(x) + \dfrac{t_{j+k+1} - x}{t_{j+k+1} - t_{j+1}} B_{j+1,k-1}(x)$

The function $f(x)$ is a linear combination of the basis functions $B_{j,k}(x)$ that are defined on the knot sequence T with n control coefficients c_j, $j = 0,\ldots, n$-1. In the B-spline construction, two main problems are considered: knot specification and the estimation of the control coefficients. The first problem consists of selecting suitable knots for a data set. This problem can be approached *a priori*. In the literature, various automatic numerical methods are available for optimizing the positions of the knots. For simplicity, B-splines with equidistant knots are used in this paper. In this case, when assuming a set of specified knots, the B-spline approximation problem is reduced to the estimation of the control coefficients. This mathematical formalism can be extended to interval and fuzzy interval frameworks. For example, in an imprecise environment, the crisp B-spline model can be transformed into an interval B-spline as follows:

$$[y(x)] = [y^-(x), y^+(x)] = [f^-(x), f^+(x)] = \Sigma_{j=0}^{n-1}\ [c_i] \cdot B_{j,k}(x)$$

where $[c_j]$ denote the control intervals and $[f(x)]$ is the interval B-spline.

Let us illustrate the B-spline principle using crisp input-output data. Fig. 35.a. illustrates a crisp model in which the objective is to identify a function $f(x)$ that fits the input-output data. In this illustration, the number of control coefficients is 5 and the knot sequence is $T = \{1, 1, 1, 1, 15.5, 30, 30, 30, 30\}$. The control coefficients are crisp values (see Fig. 35.a). The first and last knots are of multiplicity 4 ($k = 4$ for cubic B-splines). Thus, 2 interval regions on the abscissa axis are considered: [1, 15.5] and [15.5, 30]. On each region, a polynomial of degree 3 (cubic spline) is used and the polynomials are constrained such that they join smoothly at the knots (at the region boundaries).
Fig. 35.b. illustrates a possibilistic interval B-spline function $[f(x)]$, where the objective is to envelop the data, namely, the measured outputs are included in the model outputs. In contrast to the scenario in Fig. 35.a, the control coefficients and the model outputs are CIs.

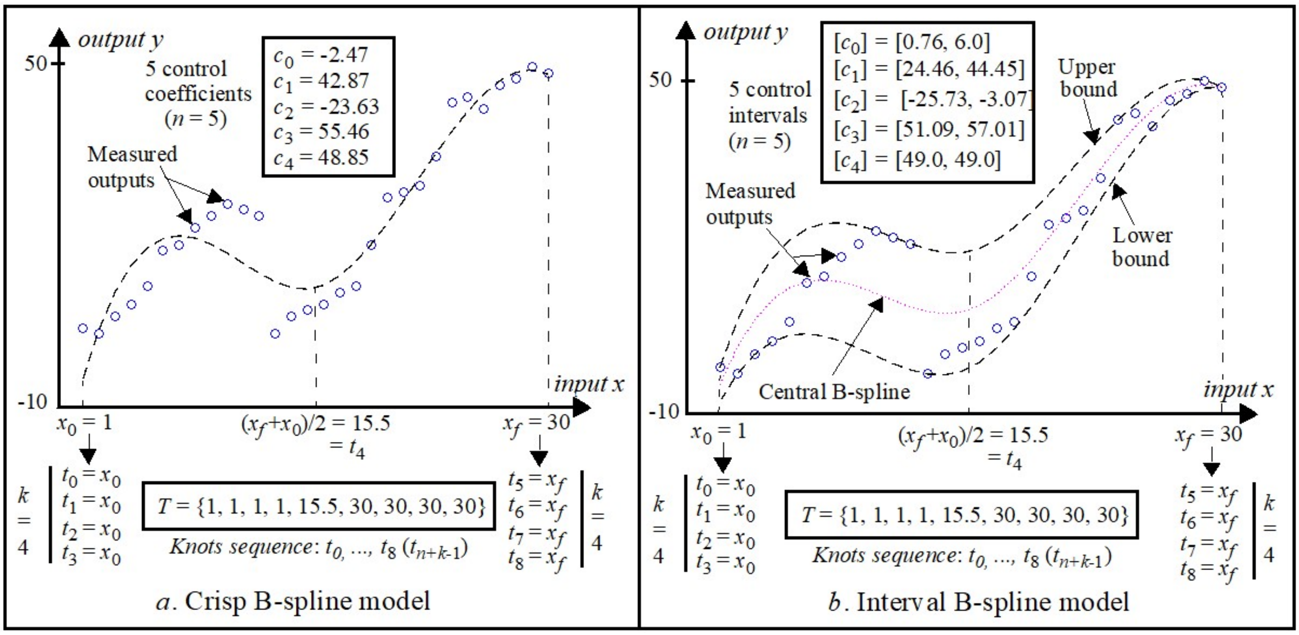


Fig. 35: B-spline model illustration (5 control coefficients)

If the control coefficients are T1FIs and are represented by $[c_j(\lambda)]$, the B-spline model can be expressed by the following T1FI expression:

$$[y_\lambda(x)] = [y_\lambda^-(x), y_\lambda^+(x)] = [f_\lambda^-(x), f_\lambda^+(x)] = \Sigma_{j=0}^{n-1}\, [c_j(\lambda)] \times B_{j,k}(x)$$

where $[y_\lambda(x)]$ is the T1FI (GI) model output.
In the application that is proposed in Section VII.2., the number of control coefficients is selected as 7. In this case, as the domain of inputs x is [0, 1], the equidistant sequence of knots is {0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1}. Therefore, the B-spline uses four polynomials of degree 3 (cubic splines). Indeed, according to the knot sequence, we have 4 interval regions on the abscissa axis: [0, 0.25], [0.25, 0.5], [0.5, 0.75] and [0.75, 1]. On each interval region, a polynomial of degree 3 is used.