Contemporary Mathematics
https://ojs.wiserpub.com/index.php/CM/

UNIVERSAL WISER PUBLISHER

Research Article

# Advanced Fuzzy Polynomial Approximation with Triangular Linear Diophantine Least Squares

**Naveed Khan[1], Mudassir Shams[1,2], Nasreen Kausar[2,3,4*], Behzad Omidi Koma[5]**

[1] Department of Mathematics and Statistics, Riphah International University, Islamabad, 44000, Pakistan
[2] Department of Mathematics, Faculty of Arts and Science, Balikesir University, Balikesir, 10145, Turkey
[3] Department of Mathematics, Faculty of Engineering and Natural Sciences, Istinye University, Istanbul, 34396, Turkey
[4] Department of Mathematics, Faculty of Arts and Science, Yildiz Technical University, Esenler, Istanbul, 34220, Turkey
[5] College of Engineering and Technology, American University of the Middle East, Egaila, 54200, Kuwait
E-mail: nasreen.kausar@balikesir.edu.tr

**Received:** 24 September 2025; **Revised:** 24 October 2025; **Accepted:** 30 October 2025

**Abstract:** Modeling uncertainty and imprecision is a substantial challenge in a variety of scientific and engineering domains. Conventional techniques, such as intuitionistic, Pythagorean, and $q$-rung orthopair fuzzy sets, have a restricted ability to capture all degrees of membership and non-membership. This study introduces a triangular linear Diophantine fuzzy least squares polynomial, which provides an efficient and reliable framework for evaluating fuzzy problems. Numerical test problems from engineering are utilized to demonstrate the applicability and precision of the method compared to existing schemes. The numerical results reveal that the developed technique is more reliable and efficient than the existing methods in terms of memory usage, memory utilization, mean square error, root square error, standard deviations, and time to find the approximate solution to the fuzzy problem. The numerical findings show that the proposed triangular linear Diophantine fuzzy least-square framework is the best alternative to address scientific and technical problems.

***Keywords***: triangular intuitionistic fuzzy linear system, semi analytical method, iterative methods, computational Central Processing Unit (CPU)-time

**MSC:** 41A10, 65D15, 90C70

## 1. Introduction

Fuzzy logic, first introduced by Zadeh [1], is crucial for research and engineering as it simulates uncertainty and imprecision in real-world systems. It is now widely used in various scientific and academic domains. It enables more realistic depictions of complex processes without precise values. Fuzziness improves control, prediction, and decision-making in scientific and technological applications. In contrast to the conventional Cantorian set, fuzzy sets deal with ambiguity and uncertainty. In fuzzy set theory, an element's membership in a fuzzy set is represented by a single number between zero and one. However, because there may be some ambiguity, the degree of non-membership of an element in a fuzzy set is not always equal to one minus the membership degree. The hesitation margin, or degree of


DOI: https://doi.org/10.37256/cm.7220268708

hesitancy, is one of the intuitionistic fuzzy sets that Atanassov [2] analyzed. Using Left-Right (L-R) fuzzy numbers as its foundation, fuzzy arithmetic was first developed by Dubois and Prade [3]. Burillo et al. [4] introduced the concept of generalized intuitionistic fuzzy numbers and proposed new operations to further extend this subject. In addition, Mitchell used intuitionistic fuzzy numbers to create methods for ranking the best and worst options [5]. Mahapatra and Roy [6] proposed triangular intuitionistic fuzzy numbers and investigated their application in assessing unreliability. Additionally, the $(\alpha, \beta)$-cut technique was used to introduce trapezoidal intuitionistic fuzzy numbers and related arithmetic operations. In [7], Wang and Zhang examined several new operations and presented the idea of intuitionistic trapezoidal fuzzy numbers. Using the $(\alpha, \beta)$-cut, Parvathi presented and explored the characteristics of symmetric trapezoidal intuitionistic fuzzy numbers [8].

The ability of fuzzy sets to depict uncertainty in practical contexts makes them important. Fuzzy numbers can be employed in systems to represent ambiguous or incomplete data, resulting in more adaptable and realistic models. The concept of Linear Diophantine Fuzzy Sets (LDFS), first suggested by Riaz and Hashmi [9], enlarged the space of fuzzy and intuitive fuzzy sets by allowing reference parameters to correspond to membership and non-membership grades. This modification is particularly beneficial in multi-attribute decision-making problems, where decision-makers can select membership grades based on their subjective assessments [10].

Recent research has advanced the theoretical and practical aspects of LDFSs. Riaz et al. [11] developed hybrid models, including Linear Diophantine Fuzzy Soft Rough Sets (LDFSRSs) and Soft Rough Linear Diophanne Fuzzy Sets (SRLDFSs), to address multi-criteria decision-making problems. Kamaci [12] applied LDFSs to algebraic structures like groups, rings, and fields, examining key characteristics. Almagrabi [13] used LDFSs in q-linear Diophantine fuzzy emergency decision support systems to make COVID-19-related decisions. Khan et al. [14] developed Triangular Linear Diophantine Fuzzy Numbers (TLDFNs) to solve linear and quadratic equations. Despite these major advances, existing approaches for complicated fuzzy systems are still limited in terms of scalability, computing efficiency, and practical applicability.

Systems of fuzzy polynomial equations are widely used in fuzzy mathematics, engineering, and applied sciences, highlighting the need for efficient solution methods. Several studies [15–23] have investigated numerical solutions $x \in \mathbb{R}$ (if they exist) for equations of the form

$$A_1 x + A_2 x^2 + \cdots + A_n x^n = A_0,$$

where $A_1, A_2, \ldots, A_n$ are fuzzy numbers. Abbasbandy and Amirfakhrian [22] proposed methods for finding the best approximation of a fuzzy function on a set of points. Abbasbandy et al. [19] developed Fuzzy Neural Networks (FNN) using fuzzy input signals and real-number weights, similar to a fuzzy polynomial system, to estimate actual roots through learning. Abbasbandy and Amirfakhriana [24] applied optimization strategies to solve fuzzy function best approximation problems.

Fuzzy numbers are beneficial for large datasets that contain uncertainties due to measurement errors, experimental variability, or defective instruments. Fuzzy numerical techniques have gained popularity due to the need to approximate functions in uncertain settings. Fuzzy interpolation and approximation are important in statistics, data analysis, engineering, and control systems because they allow the modeling of relationships between variables in uncertain environments. Lowen's fuzzy Lagrange interpolation theorem ([24]) provides a foundation for interpolating fuzzy data. Kaleva [25] proposed fuzzy approximation functions to extend traditional interpolation concepts to fuzzy environments. Allahviranloo et al. [21] used fuzzy-lower-upper function approximation to present a fuzzy polynomial-based method for numerical estimation.

Despite these advancements, there are still limitations. Existing methods limit the usage of fuzzy numbers, lack frameworks for higher-order approximations, and fail to consider processing efficiency in large-scale applications. The fuzzy least squares approximation problem for high uncertainty data is discussed in this study, which employs triangular linear Diophantine fuzzy numbers. This framework is robust, adaptive, and computationally feasible.

This study extends fuzzy least squares estimation to TLDFNs, providing a practical approach for estimating uncertain data. The main contributions of this study are:

• A thorough review of fuzzy, intuitionistic, and linear Diophantine fuzzy numbers for approximation issues.
• Formulation of fuzzy least squares approximation problem using TLDFNs.
• Developed a polynomial-based solution approach and stepwise estimation procedure.
• Validation of the proposed method with numerical examples to ensure accuracy and robustness.
• Enhancements to previous methods include increased flexibility, generality, and computing efficiency.

This paper is structured as follows: Section 2 discusses the fundamentals of triangular linear Diophantine fuzzy numbers and associated arithmetic operations. In Section 3, the suggested model for the fuzzy least squares approximation problem is presented. To estimate a fuzzy function under uncertainty, a six-step method is described in Section 4. Section 5 offers numerical examples to validate the proposed methodology and show its application. Finally, Section 6 summarizes the findings and explores prospective future research directions, such as applications in engineering, statistics, and uncertainty decision-making.

## 2. Preliminaries

To provide a solid foundation for future improvements, we must first review a few fundamental concepts relating to fuzzy sets, convexity, cut-level sets, and LDFS. These definitions and characteristics will serve as the foundation for comprehending the study's theoretical framework.

**Definition 1** Let $X$ be a universal set and $\mu_{\widetilde{\alpha}}\colon X \to [0, 1]$ a membership function. The Fuzzy Set (FS) $\widetilde{\alpha}$ [1] is defined as

$$\widetilde{\alpha} = \{(\varsigma^{\star}, \mu_{\widetilde{\alpha}}(\varsigma^{\star})) \mid \varsigma^{\star} \in X,\ \mu_{\widetilde{\alpha}}(\varsigma^{\star}) \in [0, 1]\}.$$

**Definition 2** Let $\widetilde{\alpha}$ be a fuzzy subset of a universal set $X$. The fuzzy set $\widetilde{\alpha}$ is convex [26] if, for all $r, s \in X$ and $\lambda \in [0, 1]$,

$$\mu_{\widetilde{\alpha}}(\lambda r + (1-\lambda)s) \geq \min\{\mu_{\widetilde{\alpha}}(r), \mu_{\widetilde{\alpha}}(s)\}.$$

**Definition 3** A fuzzy set $\widetilde{\alpha}$ is normalized if $h(\widetilde{\alpha}) = 1$.

Fuzzy sets can be analyzed through their cut-levels, which provide crisp subsets corresponding to specific membership thresholds.

**Definition 4** For a fuzzy set $\widetilde{\alpha}$, the $\alpha$-cut (or cut-level set) is defined as [9]

$$\widetilde{\alpha}_{\alpha} = \{\varsigma^{\star} \in X \mid \mu_{\widetilde{\alpha}}(\varsigma^{\star}) \geq \alpha\}, \quad \forall \alpha \in (0, 1].$$

In many problems, it is useful to represent uncertainty and hesitation using fuzzy structures. A Triangular Linear Diophantine Fuzzy Sets (TLDFS) extends classical fuzzy sets by capturing dual membership degrees, which allows modeling hesitation more effectively [9].

**Definition 5** Let $X$ be a universal set. A TLDFS $L_R$ on $X$ is defined as

$$L_R = \left\{(\varsigma^{\star}, \langle \widetilde{\beta}_R^{\upsilon_1}(\varsigma^{\star}), \widetilde{\beta}_R^{\upsilon_2}(\varsigma^{\star})\rangle, \langle \widetilde{\beta}_R^{\upsilon_3}(\varsigma^{\star}), \widetilde{\beta}_R^{\upsilon_4}(\varsigma^{\star})\rangle) \colon \varsigma^{\star} \in X\right\},$$

where $\widetilde{\beta}_R^{\upsilon_i}(\varsigma^\star) \in [0, 1]$ for $i = 1, 2, 3, 4$, satisfying

$$0 \leq \widetilde{\beta}_R^{\upsilon_3}(\varsigma^\star)\widetilde{\beta}_R^{\upsilon_1}(\varsigma^\star) + \widetilde{\beta}_R^{\upsilon_4}(\varsigma^\star)\widetilde{\beta}_R^{\upsilon_2}(\varsigma^\star) \leq 1,$$

$$0 \leq \widetilde{\beta}_R^{\upsilon_3}(\varsigma^\star) + \widetilde{\beta}_R^{\upsilon_4}(\varsigma^\star) \leq 1, \quad \forall \varsigma^\star \in X.$$

The associated *hesitation degree* is given by

$$\dot{G}_R = 1 - \left(\widetilde{\beta}_R^{\upsilon_3}(\varsigma^\star)\widetilde{\beta}_R^{\upsilon_1}(\varsigma^\star) + \widetilde{\beta}_R^{\upsilon_4}(\varsigma^\star)\widetilde{\beta}_R^{\upsilon_2}(\varsigma^\star)\right),$$

where $\dot{G}_R$ is a reference parameter. For brevity, we write

$$L_R = (\langle \widetilde{\beta}_R^{\upsilon_1}, \widetilde{\beta}_R^{\upsilon_2} \rangle, \langle \widetilde{\beta}_R^{\upsilon_3}, \widetilde{\beta}_R^{\upsilon_4} \rangle).$$

In the context of TLDFS, we can define absolute and empty sets, as well as cuts sets, to handle extreme and intermediate membership cases [9].

**Definition 6** (Absolute and Empty LDFS) Let $X$ be a universal set. The *absolute LDFS* on $X$ is defined as

$${}^1L_R = \{\varsigma^\star, \langle 1, 0 \rangle, \langle 1, 0 \rangle : \varsigma^\star \in X\},$$

representing full membership with no hesitation. The *empty LDFS* is given by

$${}^0L_R = \{\varsigma^\star, \langle 0, 1 \rangle, \langle 0, 1 \rangle : \varsigma^\star \in X\},$$

representing zero membership with complete hesitation.

**Definition 7** (LDFS $\alpha$-cut) Let

$$L_R = \left\{(\varsigma^\star, \langle \widetilde{\beta}_R^{\upsilon_1}(\varsigma^\star), \widetilde{\beta}_R^{\upsilon_2}(\varsigma^\star) \rangle, \langle \widetilde{\beta}_R^{\upsilon_3}(\varsigma^\star), \widetilde{\beta}_R^{\upsilon_4}(\varsigma^\star) \rangle) : \varsigma^\star \in X\right\}$$

be an LDFS. For constants $\widetilde{\beta}_R^{\upsilon_i} \in [0, 1]$ $(i = 1, 2, 3, 4)$ satisfying

$$0 \leq \widetilde{\beta}_R^{\upsilon_1}\widetilde{\beta}_R^{\upsilon_3} + \widetilde{\beta}_R^{\upsilon_2}\widetilde{\beta}_R^{\upsilon_4} \leq 1, \quad 0 \leq \widetilde{\beta}_R^{\upsilon_3} + \widetilde{\beta}_R^{\upsilon_4} \leq 1,$$

the $\left(\langle \widetilde{\beta}_R^{\upsilon_1}, \widetilde{\beta}_R^{\upsilon_2} \rangle, \langle \widetilde{\beta}_R^{\upsilon_3}, \widetilde{\beta}_R^{\upsilon_4} \rangle\right)$-cut of $L_R$ is defined as

$$(L_R)^{\langle \widetilde{\beta}_R^{\upsilon_1}, \widetilde{\beta}_R^{\upsilon_2} \rangle}_{\langle \widetilde{\beta}_R^{\upsilon_3}, \widetilde{\beta}_R^{\upsilon_4} \rangle} = \left\{\varsigma^\star \in X : \widetilde{\beta}_R^{\upsilon_1}(\varsigma^\star) \geq \widetilde{\beta}_R^{\upsilon_1},\ \widetilde{\beta}_R^{\upsilon_2}(\varsigma^\star) \leq \widetilde{\beta}_R^{\upsilon_2},\ \widetilde{\beta}_R^{\upsilon_3}(\varsigma^\star) \geq \widetilde{\beta}_R^{\upsilon_3},\ \widetilde{\beta}_R^{\upsilon_4}(\varsigma^\star) \leq \widetilde{\beta}_R^{\upsilon_4}\right\}.$$

## 2.1 *TLDF numbers*

TLDF numbers (Figure 1) are widely used to model uncertainty with both membership and non-membership functions. They are characterized by a triangular shape, convexity of membership, and concavity of non-membership functions [14].

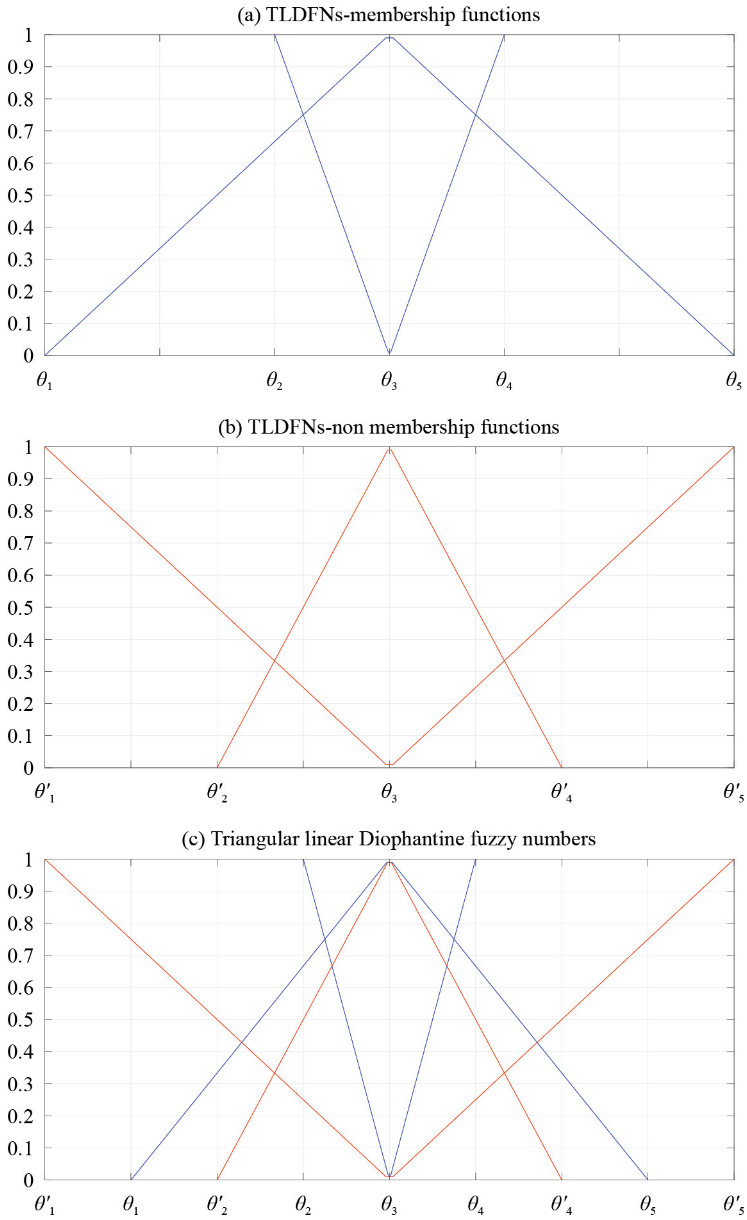


**Figure 1.** Shows Triangular linear Diophantine fuzzy numbers

**Definition 8** (TLDFN) A Linear Dual Fuzzy (LDF) number $L_R$ on $\mathbb{R}$ is defined as follows:

1. $L_R$ is an LDF fuzzy subset of $\mathbb{R}$.
2. **Normality:** There exists $\varsigma_0^\star \in \mathbb{R}$ such that

$$\widetilde{\beta}_R^{\upsilon_1}(\varsigma_0^\star) = 1, \;\; \widetilde{\beta}_R^{\upsilon_2}(\varsigma_0^\star) = 0, \;\; \widetilde{\beta}_R^{\upsilon_3}(\varsigma_0^\star) = 1, \;\; \widetilde{\beta}_R^{\upsilon_4}(\varsigma_0^\star) = 0.$$

3. **Membership functions are convex:** For all $\varsigma_1^\star$, $\varsigma_2^\star \in \mathbb{R}$ and $\lambda \in [0, 1]$,

$$\widetilde{\beta}_R^{\upsilon_1}(\lambda\varsigma_1^\star + (1-\lambda)\varsigma_2^\star) \geq \min\{\widetilde{\beta}_R^{\upsilon_1}(\varsigma_1^\star), \widetilde{\beta}_R^{\upsilon_1}(\varsigma_2^\star)\},$$

$$\widetilde{\beta}_R^{\upsilon_3}(\lambda\varsigma_1^\star + (1-\lambda)\varsigma_2^\star) \geq \min\{\widetilde{\beta}_R^{\upsilon_3}(\varsigma_1^\star), \widetilde{\beta}_R^{\upsilon_3}(\varsigma_2^\star)\}.$$

4. **Non-membership functions are concave:** For all $\varsigma_1^\star$, $\varsigma_2^\star \in \mathbb{R}$ and $\lambda \in [0, 1]$,

$$\widetilde{\beta}_R^{\upsilon_2}(\lambda\varsigma_1^\star + (1-\lambda)\varsigma_2^\star) \leq \max\{\widetilde{\beta}_R^{\upsilon_2}(\varsigma_1^\star), \widetilde{\beta}_R^{\upsilon_2}(\varsigma_2^\star)\},$$

$$\widetilde{\beta}_R^{\upsilon_4}(\lambda\varsigma_1^\star + (1-\lambda)\varsigma_2^\star) \leq \max\{\widetilde{\beta}_R^{\upsilon_4}(\varsigma_1^\star), \widetilde{\beta}_R^{\upsilon_4}(\varsigma_2^\star)\}.$$

Triangular LDF numbers are classified into four types based on the combinations of membership and non-membership functions.

**Definition 9** A TLDFN $\mathbb{R}$ is a fuzzy number in which the membership and non-membership functions are piecewise linear and form triangular shapes [14].

Let $L_R$ be a TLDFN characterized by parameters $\varsigma_1^\star < \varsigma_3^\star < \varsigma_5^\star$ for the membership function and corresponding parameters $\varsigma_2^{\star\prime} < \varsigma_3^{\star\prime} < \varsigma_4^{\star\prime}$ for the non-membership function. Then, the membership and non-membership functions are defined as follows:

$$\widetilde{\beta}_R^{\upsilon_1}(x) = \begin{cases} \dfrac{x-\varsigma_1^\star}{\varsigma_3^\star-\varsigma_1^\star}, & \varsigma_1^\star \leq x \leq \varsigma_3^\star, \\[2ex] \dfrac{\varsigma_5^\star - x}{\varsigma_5^\star-\varsigma_3^\star}, & \varsigma_3^\star \leq x \leq \varsigma_5^\star, \\[2ex] 0, & \text{otherwise}, \end{cases}$$

$$\widetilde{\beta}_R^{\upsilon_2}(x) = \begin{cases} \dfrac{\varsigma_3^\star - x}{\varsigma_3^\star-\varsigma_2^\star}, & \varsigma_2^\star \leq x \leq \varsigma_3^\star, \\[2ex] \dfrac{x-\varsigma_3^\star}{\varsigma_4^\star-\varsigma_3^\star}, & \varsigma_3^\star \leq x \leq \varsigma_4^\star, \\[2ex] 0, & \text{otherwise}, \end{cases}$$

and similarly, the dual pair functions are

$$\widetilde{\beta}_R^{\upsilon_3}(x) = \begin{cases} \dfrac{x - \varsigma_2^{\star\prime}}{\varsigma_3^{\star\prime} - \varsigma_2^{\star\prime}}, & \varsigma_2^{\star\prime} \leq x \leq \varsigma_3^{\star\prime}, \\ \\ \dfrac{\varsigma_4^{\star\prime} - x}{\varsigma_4^{\star\prime} - \varsigma_3^{\star\prime}}, & \varsigma_3^{\star\prime} \leq x \leq \varsigma_4^{\star\prime}, \\ \\ 0, & \text{otherwise,} \end{cases}$$

$$\widetilde{\beta}_R^{\upsilon_4}(x) = \begin{cases} \dfrac{\varsigma_3^{\star\prime} - x}{\varsigma_3^{\star\prime} - \varsigma_1^{\star\prime}}, & \varsigma_1^{\star\prime} \leq x \leq \varsigma_3^{\star\prime}, \\ \\ \dfrac{x - \varsigma_3^{\star\prime}}{\varsigma_5^{\star\prime} - \varsigma_3^{\star\prime}}, & \varsigma_3^{\star\prime} \leq x \leq \varsigma_5^{\star\prime}, \\ \\ 0, & \text{otherwise.} \end{cases}$$

**Remark.** The parameters $\varsigma_1^\star$, $\varsigma_3^\star$, and $\varsigma_5^\star$ define the left endpoint, peak point, and right endpoint of the triangular membership function $\widetilde{\beta}_R^{\upsilon_1}(x)$, respectively. In particular, $\varsigma_5^\star$ determines the upper support boundary of the TLDFN, ensuring that $\widetilde{\beta}_R^{\upsilon_1}(x)$ smoothly decreases to zero beyond $\varsigma_3^\star$ and thus completes the triangular structure.

The triangular LDF number $L_R$ is classified as:

1. LDFN-I: $\varsigma_3^\star = \varsigma_3^{\star\prime}$, $\varsigma_1^\star \leq \varsigma_2^\star \leq \varsigma_3^\star \leq \varsigma_4^\star \leq \varsigma_5^\star$;
2. LDFN-II: $\varsigma_3^\star \neq \varsigma_3^{\star\prime}$, $\varsigma_1^\star \leq \varsigma_2^\star \leq \varsigma_3^\star \leq \varsigma_4^\star \leq \varsigma_5^\star$;
3. LDFN-III: $\varsigma_3^\star = \varsigma_3^{\star\prime}$, $\varsigma_2^\star \leq \varsigma_1^\star \leq \varsigma_3^\star \leq \varsigma_5^\star \leq \varsigma_4^\star$;
4. LDFN-IV: $\varsigma_3^\star \neq \varsigma_3^{\star\prime}$, $\varsigma_2^\star \leq \varsigma_1^\star \leq \varsigma_3^\star \leq \varsigma_5^\star \leq \varsigma_4^\star$.

**Remark:** In this study, only TLDFN-I is considered, defined as

$$L_{R^\star} = \left\langle \begin{array}{c} (\varsigma_1^\star, \varsigma_2^\star, \varsigma_3^\star, \varsigma_4^\star, \varsigma_5^\star), \\ \\ (\varsigma_1^{\star\prime}, \varsigma_2^{\star\prime}, \varsigma_3^\star, \varsigma_4^{\star\prime}, \varsigma_5^{\star\prime}) \end{array} \right\rangle.$$

**Remark:** If we $\varsigma_1^{\star'} = \varsigma_2^{\star'} = \varsigma_1^\star = \varsigma_2^\star$ and $\varsigma_4^{\star'} = \varsigma_5^{\star'} = \varsigma_1^\star = \varsigma_2^\star$ then type-I and type-III are equivalent

**Definition 10** [14] Consider a TLDFN

$$L_{R^\star} = \left\langle \begin{array}{c} (\varsigma_1^\star, \varsigma_2^\star, \varsigma_3^\star, \varsigma_4^\star, \varsigma_5^\star), \\ \\ (\varsigma_1^{\star'}, \varsigma_2^{\star'}, \varsigma_3^\star, \varsigma_4^{\star'}, \varsigma_5^{\star'}) \end{array} \right\rangle.$$

Then, the $\widetilde{\beta}$-cut sets are defined as follows:

1. $\widetilde{\beta}_R^{\upsilon_1}$-cut set:

$$L_{R^*}^{\widetilde{\beta}_R^{\upsilon_1}} = \{x \in X; \widetilde{\alpha}_R^\tau(x) \geq \widetilde{\beta}_R^{\upsilon_1}\}$$

$$= \left[ {}^{\circ}\widetilde{\beta}_R^{\upsilon_1}(\widetilde{\beta}_R^{\upsilon_1}), {}_{\circ}\widetilde{\beta}_R^{\upsilon_1}(\widetilde{\beta}_R^{\upsilon_1}) \right]$$

$$= \left[ \varsigma_1^{\star} - \widetilde{\beta}_R^{\upsilon_1}(\varsigma_3^{\star} - \varsigma_1^{\star}), \varsigma_5^{\star} - \widetilde{\beta}_R^{\upsilon_1}(\varsigma_5^{\star} - \varsigma_3^{\star}) \right].$$

2. $\widetilde{\beta}_R^{\upsilon_2}$-cut set:

$$L_{R^*}^{\widetilde{\beta}_R^{\upsilon_2}} = \{x \in X; \widetilde{\beta}_R^{\upsilon_2}(x) \leq \widetilde{\beta}_R^{\upsilon_2}\}$$

$$= \left[ {}^{\circ}\widetilde{\beta}_R^{\upsilon_2}(\widetilde{\beta}_R^{\upsilon_2}), {}_{\circ}\widetilde{\beta}_R^{\upsilon_2}(\widetilde{\beta}_R^{\upsilon_2}) \right]$$

$$= \left[ \varsigma_3^{\star} - \widetilde{\beta}_R^{\upsilon_2}(\varsigma_3^{\star} - \varsigma_2^{\star}), \varsigma_3^{\star} + \widetilde{\beta}_R^{\upsilon_2}(\varsigma_4^{\star} - \varsigma_3^{\star}) \right].$$

3. $\widetilde{\beta}_R^{\upsilon_3}$-cut set:

$$L_{R^*}^{\widetilde{\beta}_R^{\upsilon_3}} = \{x \in X; \widetilde{\beta}_R^{\upsilon_3}(x) \geq \widetilde{\beta}_R^{\upsilon_3}\}$$

$$= \left[ {}^{\circ}\widetilde{\beta}_R^{\upsilon_3}(\widetilde{\beta}_R^{\upsilon_3}), {}_{\circ}\widetilde{\beta}_R^{\upsilon_3}(\widetilde{\beta}_R^{\upsilon_3}) \right]$$

$$= \left[ \varsigma_2^{\star'} + \widetilde{\beta}_R^{\upsilon_3}(\varsigma_3^{\star} - \varsigma_2^{\star'}), \varsigma_4^{\star'} - \widetilde{\beta}_R^{\upsilon_3}(\varsigma_4^{\star'} - \varsigma_3^{\star}) \right].$$

4. $\widetilde{\beta}_R^{\upsilon_4}$-cut set:

$$L_{R^*}^{\widetilde{\beta}_R^{\upsilon_4}} = \{x \in X; \widetilde{\beta}_R^{\upsilon_4}(x) \leq \widetilde{\beta}_R^{\upsilon_4}\}$$

$$= \left[ {}^{\circ}\widetilde{\beta}_R^{\upsilon_4}(\widetilde{\beta}_R^{\upsilon_3}), {}_{\circ}\widetilde{\beta}_R^{\upsilon_4}(\widetilde{\beta}_R^{\upsilon_3}) \right]$$

$$= \left[ \varsigma_3^{\star} - \varsigma^{\star}(\varsigma_3^{\star} - \varsigma_1^{\star'}), \varsigma_3^{\star} + \varsigma^{\star}(\varsigma_5^{\star'} - \varsigma_3^{\star}) \right].$$

The $(< \widetilde{\beta}_R^{\upsilon_1}, \widetilde{\beta}_R^{\upsilon_2} >, < \widetilde{\beta}_R^{\upsilon_3}, \widetilde{\beta}_R^{\upsilon_4} >)$-cut of $L_{R^\star}$ is denoted by:

$$(L_{R^\star})^{<\widetilde{\beta}_R^{\upsilon_1}, \widetilde{\beta}_R^{\upsilon_2}>}_{<\widetilde{\beta}_R^{\upsilon_3}, \widetilde{\beta}_R^{\upsilon_4}>} = \left\langle \begin{array}{c} \left([{}^{\circ}\widetilde{\beta}_R^{\upsilon_1}(\widetilde{\beta}_R^{\upsilon_1}), {}_{\circ}\widetilde{\beta}_R^{\upsilon_1}(\widetilde{\beta}_R^{\upsilon_1})], [{}^{\circ}\widetilde{\beta}_R^{\upsilon_2}(\widetilde{\beta}_R^{\upsilon_2}), {}_{\circ}\widetilde{\beta}_R^{\upsilon_2}(\widetilde{\beta}_R^{\upsilon_2})]\right), \\ \left([{}^{\circ}\widetilde{\beta}_R^{\upsilon_3}(\widetilde{\beta}_R^{\upsilon_3}), {}_{\circ}\widetilde{\beta}_R^{\upsilon_3}(\widetilde{\beta}_R^{\upsilon_3})], [{}^{\circ}\widetilde{\beta}_R^{\upsilon_4}(\widetilde{\beta}_R^{\upsilon_3}), {}_{\circ}\widetilde{\beta}_R^{\upsilon_4}(\widetilde{\beta}_R^{\upsilon_3})]\right) \end{array} \right\rangle.$$

The set of all TLDFN on $\mathbb{R}$ is denoted by $L_{R^\star}$.

**Definition 11** Let

$$L_R = \left\langle \left(\widetilde{\beta}_R^{\upsilon_1}, \widetilde{\beta}_R^{\upsilon_2}\right), \left(\widetilde{\beta}_R^{\upsilon_3}, \widetilde{\beta}_R^{\upsilon_4}\right) \right\rangle, \quad L_B = \left\langle \left(\widetilde{\beta}_B^{\upsilon_1}, \widetilde{\beta}_B^{\upsilon_2}\right), \left(\widetilde{\beta}_B^{\upsilon_3}, \widetilde{\beta}_B^{\upsilon_4}\right) \right\rangle$$

be two TLDFNs defined on $\mathbb{R}$, where $\widetilde{\beta}^{\upsilon_i}$ and $\widetilde{\beta}^{\upsilon_j}$ denote the membership and non-membership functions corresponding to the lower and upper linear dual components, respectively.

Using Zadeh's extension principle, the algebraic operations $(\oplus, \ominus, \otimes, \oslash)$ between $L_R$ and $L_B$ are defined for every real $z \in \mathbb{R}$ as follows:

**(i) Addition:**

$$\widetilde{\beta}^{\upsilon_1}_{(L_R \oplus L_B)}(z) = \sup_{x+y=z} \min\left\{\widetilde{\beta}_R^{\upsilon_1}(x), \widetilde{\beta}_B^{\upsilon_1}(y)\right\},$$

$$\widetilde{\beta}^{\upsilon_2}_{(L_R \oplus L_B)}(z) = \inf_{x+y=z} \max\left\{\widetilde{\beta}_R^{\upsilon_2}(x), \widetilde{\beta}_B^{\upsilon_2}(y)\right\},$$

$$\widetilde{\beta}^{\upsilon_3}_{(L_R \oplus L_B)}(z) = \sup_{x+y=z} \min\left\{\widetilde{\beta}_R^{\upsilon_3}(x), \widetilde{\beta}_B^{\upsilon_3}(y)\right\},$$

$$\widetilde{\beta}^{\upsilon_4}_{(L_R \oplus L_B)}(z) = \inf_{x+y=z} \max\left\{\widetilde{\beta}_R^{\upsilon_4}(x), \widetilde{\beta}_B^{\upsilon_4}(y)\right\}.$$

**(ii) Subtraction:**

$$\widetilde{\beta}^{\upsilon_i}_{(L_R \ominus L_B)}(z) = \sup_{x-y=z} \min\left\{\widetilde{\beta}_R^{\upsilon_i}(x), \widetilde{\beta}_B^{\upsilon_i}(y)\right\},$$

$$\widetilde{\beta}^{\upsilon_j}_{(L_R \ominus L_B)}(z) = \inf_{x-y=z} \max\left\{\widetilde{\beta}_R^{\upsilon_j}(x), \widetilde{\beta}_B^{\upsilon_j}(y)\right\},$$

where $i \in \{1, 3\}$ and $j \in \{2, 4\}$.

**(iii) Multiplication:**

$$\widetilde{\beta}^{\upsilon_i}_{(L_R \otimes L_B)}(z) = \sup_{x \cdot y=z} \min\left\{\widetilde{\beta}_R^{\upsilon_i}(x), \widetilde{\beta}_B^{\upsilon_i}(y)\right\},$$

$$\widetilde{\beta}^{\upsilon_j}_{(L_R \otimes L_B)}(z) = \inf_{x \cdot y=z} \max\left\{\widetilde{\beta}_R^{\upsilon_j}(x), \widetilde{\beta}_B^{\upsilon_j}(y)\right\}.$$

**(iv) Division:** for all $y \neq 0$,

$$\widetilde{\beta}^{\upsilon_i}_{(L_R \oslash L_B)}(z) = \sup_{x/y=z} \min\left\{\widetilde{\beta}^{\upsilon_i}_R(x),\, \widetilde{\beta}^{\upsilon_i}_B(y)\right\},$$

$$\widetilde{\beta}^{\upsilon_j}_{(L_R \oslash L_B)}(z) = \inf_{x/y=z} \max\left\{\widetilde{\beta}^{\upsilon_j}_R(x),\, \widetilde{\beta}^{\upsilon_j}_B(y)\right\}.$$

All suprema and infima are taken over real pairs $(x, y) \in \mathbb{R}^2$ satisfying the corresponding real constraint (e.g., $x+y=z$). These definitions properly describe the membership and non-membership values of the resulting TLDFN at each real $z$, avoiding any implication that $x+y$ or similar expressions are "equal to a fuzzy set."

**Definition 12** ([14]) A TLDFN

$$L_{R^\star} = \left\{ \begin{array}{c} (\varsigma_1^\star,\, \varsigma_2^\star,\, \varsigma_3^\star,\, \varsigma_4^\star,\, \varsigma_5^\star), \\ \\ (\varsigma_1^{\star'},\, \varsigma_2^{\star'},\, \varsigma_3^\star,\, \varsigma_4^{\star'},\, \varsigma_5^{\star'}) \end{array} \right\}$$

is said to be positive if and only if $\varsigma_1^\star \geq 0$ and $\varsigma_1^{\star'} \geq 0$.

**Definition 13** ([14]) Two TLDFNs

$$L_{R^\star} = \left\langle \begin{array}{c} (\varsigma_1^\star,\, \varsigma_2^\star,\, \varsigma_3^\star,\, \varsigma_4^\star,\, \varsigma_5^\star), \\ \\ (\varsigma_1^{\star'},\, \varsigma_2^{\star'},\, \varsigma_3^\star,\, \varsigma_4^{\star'},\, \varsigma_5^{\star'}) \end{array} \right\rangle, \quad \dot{G}_{R^*} = \left\langle \begin{array}{c} (\upsilon_1,\, \upsilon_2,\, \upsilon_3,\, \upsilon_4,\, \upsilon_5), \\ \\ (\upsilon_1',\, \upsilon_2',\, \upsilon_3,\, \upsilon_4',\, \upsilon_5') \end{array} \right\rangle$$

are said to be equal if all corresponding components are equal:

$$\varsigma_i^\star = \upsilon_i, \quad \varsigma_i^{\star'} = \upsilon_i', \quad i = 1, 2, 3, 4, 5.$$

**Definition 14** ([14]) Let $L_{R^\star} = (\varsigma_1^\star, \varsigma_2^\star, \varsigma_3^\star, \varsigma_4^\star, \varsigma_5^\star)$ and $\dot{G}_{R^\star} = (\upsilon_1, \upsilon_2, \upsilon_3, \upsilon_4, \upsilon_5)$ be two positive TLDFNs, where each $\upsilon_i$, $\upsilon_i'$, not equal to zero for $i = 1, 2, 3, 4, 5$. Then, the basic arithmetic operations between $L_{R^\star}$ and $\dot{G}_{R^\star}$ are defined component-wise as:

$$\text{(i)} \quad L_{R^\star} + \dot{G}_{R^\star} = \left\{ \begin{array}{l} (\varsigma_1^\star + \upsilon_1,\, \varsigma_2^\star + \upsilon_2,\, \varsigma_3^\star + \upsilon_3,\, \varsigma_4^\star + \upsilon_4,\, \varsigma_5^\star + \upsilon_5), \\ \\ (\varsigma_1^{\star'} + \upsilon_1',\, \varsigma_2^{\star'} + \upsilon_2',\, \varsigma_3^{\star'} + \upsilon_3',\, \varsigma_4^{\star'} + \upsilon_4',\, \varsigma_5^{\star'} + \upsilon_5'), \end{array} \right.$$

$$\text{(ii)} \quad L_{R^\star} - \dot{G}_{R^\star} = \left\{ \begin{array}{l} (\varsigma_1^\star - \upsilon_1,\, \varsigma_2^\star - \upsilon_2,\, \varsigma_3^\star - \upsilon_3,\, \varsigma_4^\star - \upsilon_4,\, \varsigma_5^\star - \upsilon_5), \\ \\ (\varsigma_1^{\star'} - \upsilon_1',\, \varsigma_2^{\star'} - \upsilon_2',\, \varsigma_3^{\star'} - \upsilon_3',\, \varsigma_4^{\star'} - \upsilon_4',\, \varsigma_5^{\star'} - \upsilon_5'), \end{array} \right.$$

$$\text{(iii)} \quad L_{R^\star} \times \dot{G}_{R^\star} = \begin{cases} (\varsigma_1^\star \upsilon_1,\, \varsigma_2^\star \upsilon_2,\, \varsigma_3^\star \upsilon_3,\, \varsigma_4^\star \upsilon_4,\, \varsigma_5^\star \upsilon_5), \\ \\ (\varsigma_1^{\star'} \upsilon_1',\, \varsigma_2^{\star'} \upsilon_2',\, \varsigma_3^{\star'} \upsilon_3',\, \varsigma_4^{\star'} \upsilon_4',\, \varsigma_5^{\star'} \upsilon_5'), \end{cases}$$

$$\text{(iv)} \quad L_{R^\star} / \dot{G}_{R^\star} = \begin{cases} \left( \dfrac{\varsigma_1^\star}{\upsilon_5}, \dfrac{\varsigma_2^\star}{\upsilon_4}, \dfrac{\varsigma_3^\star}{\upsilon_3}, \dfrac{\varsigma_4^\star}{\upsilon_2}, \dfrac{\varsigma_5^\star}{\upsilon_1} \right), \\ \\ \left( \dfrac{\varsigma_1^{\star'}}{\upsilon_5'}, \dfrac{\varsigma_2^{\star'}}{\upsilon_4'}, \dfrac{\varsigma_3^{\star'}}{\upsilon_3'}, \dfrac{\varsigma_4^{\star'}}{\upsilon_2'}, \dfrac{\varsigma_5^{\star'}}{\upsilon_1'} \right), \end{cases}$$

$$\text{(v)} \quad k \times L_{R^\star} = \begin{cases} (k\,\varsigma_1^\star,\, k\,\varsigma_2^\star,\, k\,\varsigma_3^\star,\, k\,\varsigma_4^\star,\, k\,\varsigma_5^\star), & k > 0, \\ \\ (k\,\varsigma_5^\star,\, k\,\varsigma_4^\star,\, k\,\varsigma_3^\star,\, k\,\varsigma_2^\star,\, k\,\varsigma_1^\star), & k < 0. \end{cases}$$

## 2.2 *Triangular Linear Diophantine Fuzzy (TLDF) functions*

We now define the concept of the TLDF value function, which extends classical fuzzy number representations by incorporating level-dependent membership functions. This formulation allows for more flexibility in modeling uncertainty.

**Definition 15** A fuzzy number value function

$$f^*(h_i) = \left( f(h_i),\, f(h_i)^{\widetilde{\beta}_R^{\upsilon_1}},\, f(h_i)^{\widetilde{\beta}_R^{\upsilon_2}},\, f(h_i)^{\widetilde{\beta}_R^{\upsilon_3}},\, f(h_i)^{\widetilde{\beta}_R^{\upsilon_4}} \right)$$

is called a TLDF value function if its membership functions ($\widetilde{\beta}_R^{\upsilon_1}(x)$, $\widetilde{\beta}_R^{\upsilon_2}(x)$) and non membership functions ($\widetilde{\beta}_R^{\upsilon_3}(x)$, $\widetilde{\beta}_R^{\upsilon_4}(x)$) are defined as

$$\widetilde{\beta}_R^{\upsilon_1}(x) = \begin{cases} \dfrac{f(h) - \varsigma_1^\star}{\varsigma_3^\star - \varsigma_1^\star}, & \varsigma_1^\star \le x \le \varsigma_3^\star, \\ \\ \dfrac{\varsigma_5^\star - f(h)}{\varsigma_5^\star - \varsigma_3^\star}, & \varsigma_3^\star \le x \le \varsigma_5^\star, \\ \\ 0, & \text{otherwise}, \end{cases}$$

$$\widetilde{\beta}_R^{\upsilon_2}(x) = \begin{cases} \dfrac{\varsigma_3^\star - f(h)}{\varsigma_3^\star - \varsigma_2^\star}, & \varsigma_1^\star \le x \le \varsigma_3^\star, \\ \\ \dfrac{f(h) - \varsigma_3^\star}{\varsigma_4^\star - \varsigma_3^\star}, & \varsigma_3^\star \le x \le \varsigma_4^\star, \\ \\ 0, & \text{otherwise}, \end{cases}$$

$$\widetilde{\beta}_R^{\upsilon_3}(x) = \begin{cases} \dfrac{f(h) - \varsigma_2^{\star\prime}}{\varsigma_3^{\star\prime} - \varsigma_2^{\star\prime}}, & \varsigma_2^{\star\prime} \le x \le \varsigma_3^{\star\prime}, \\ \\ \dfrac{\varsigma_4^{\star\prime} - f(h)}{\varsigma_4^{\star\prime} - \varsigma_3^{\star\prime}}, & \varsigma_3^{\star\prime} \le x \le \varsigma_4^{\star\prime}, \\ \\ 0, & \text{otherwise}, \end{cases}$$

$$\widetilde{\beta}_R^{\upsilon_4}(x) = \begin{cases} \dfrac{\varsigma_3^{\star\prime} - f(h)}{\varsigma_3^{\star\prime} - \varsigma_1^{\star\prime}}, & \varsigma_1^{\star\prime} \le x \le \varsigma_3^{\star\prime}, \\ \\ \dfrac{f(h) - \varsigma_3^{\star\prime}}{\varsigma_5^{\star\prime} - \varsigma_3^{\star\prime}}, & \varsigma_3^{\star\prime} \le x \le \varsigma_5^{\star\prime}, \\ \\ 0, & \text{otherwise}, \end{cases}$$

where $\varsigma_1^{\star\prime} \le \varsigma_2^{\star\prime} \le \varsigma_3^{\star\prime} \le \varsigma_4^{\star\prime} \le \varsigma_5^{\star\prime}$ for all $x \in \mathbb{R}$.

Based on the relative positions of the defining points, $L_R$ is classified as:

1. **LDFN-I:** $\varsigma_3^{\star} = \varsigma_3^{\star\prime}$, with $\varsigma_1^{\star} \le \varsigma_2^{\star} \le \varsigma_3^{\star} \le \varsigma_4^{\star} \le \varsigma_5^{\star}$;
2. **LDFN-II:** $\varsigma_3^{\star} \neq \varsigma_3^{\star\prime}$, with $\varsigma_1^{\star} \le \varsigma_2^{\star} \le \varsigma_3^{\star} \le \varsigma_4^{\star} \le \varsigma_5^{\star}$;
3. **LDFN-III:** $\varsigma_3^{\star} = \varsigma_3^{\star\prime}$, with $\varsigma_2^{\star} \le \varsigma_1^{\star} \le \varsigma_3^{\star} \le \varsigma_5^{\star} \le \varsigma_4^{\star}$;
4. **LDFN-IV:** $\varsigma_3^{\star} \neq \varsigma_3^{\star\prime}$, with $\varsigma_2^{\star} \le \varsigma_1^{\star} \le \varsigma_3^{\star} \le \varsigma_5^{\star} \le \varsigma_4^{\star}$.

Throughout this study, only the triangular LDFN of type I (LDFN-I) is considered.

**Lemma 1** ([24, 27, 28]) Least squares principle.

The least squares method approximates a set of data points $(h_i, g_i)$, $i = 1, 2, \ldots, m$, using a curve $g = \varphi(h)$. The principle requires that the variation between the measured data $g_i$ and the corresponding values $\varphi(h_i)$ is minimized. This is achieved by minimizing the sum of squared deviations:

$$\sum_{i=1}^{m} [\varphi(h_i) - g_i]^2 .$$

Two common measures of approximation error are:

**1-Weighted Least Squares Error**

To quantify the deviation between the approximation function $\varphi(h)$ and the observed data $g_i$, we define the weighted least squares error as

$$\omega_1 = \sqrt{\sum_{i=1}^{m} \Psi_i (\varphi(h_i) - g_i)^2}, \quad \Psi_i \ge 0 \text{ is the weight associated with the } i\text{-th data point.} \tag{1}$$

Here:

• $h_i$ denotes the independent variable at the $i$-th observation.
• $g_i$ is the corresponding observed value.
• $\varphi(h_i)$ represents the approximation or predicted value at $h_i$.
• $\Psi_i$ allows assigning different importance to different data points; for uniform weighting, one may set $\Psi_i = 1$.

This formulation provides a flexible framework for approximating fuzzy data with a continuous function. It naturally extends to TLDFNs by treating each component of the fuzzy number separately and applying the weighted least squares minimization to capture the overall deviation. Note that the expression computes a weighted root mean square error and does not involve a min operation.

**2-Maximum deviation:**

$$\omega_2 = \max_{1\leq i\leq m} |\varphi(h_i) - g_i|. \tag{2}$$

## 2.3 *Triangular Linear Diophantine Fuzzy Least Squares Approximation (TLDFLSA)*

We adopt the TLDFLSA since it generalizes conventional fuzzy sets such as triangular, trapezoidal, and intuitionistic forms. Its extended structure captures a wider spectrum of uncertainty and provides higher flexibility, making it efficient and robust for complex fuzzy approximation problems.

Consider the fuzzy data set $\{h_i, f^*(h_i)\}$, where $h_i \in \mathbb{R}$, $h_1 < h_2 < \cdots < h_m$, and $f^*(h_i)$ represents the Triangular Linear Diophantine Fuzzy Numbers (TLDFNs). Each fuzzy value comprises the membership $(f_{m*}(h_i))$ and non-membership components $(f_{n*}(h_i))$ expressed as five-tuples:

$$\widetilde{f}(h_i) = \begin{bmatrix} f_{m*}(h_i) \\ f_{n*}(h_i) \end{bmatrix} = \begin{bmatrix} \left(f(h_i),\, f(h_i)^{\widetilde{\beta}_R^{\upsilon_1}},\, f(h_i)^{\widetilde{\beta}_R^{\upsilon_2}},\, f(h_i)^{\widetilde{\beta}_R^{\upsilon_3}},\, f(h_i)^{\widetilde{\beta}_R^{\upsilon_4}}\right) \\ \left(f_*(h_i),\, f(h_i)^{\widetilde{\beta}_R^{\upsilon_1 *}},\, f(h_i)^{\widetilde{\beta}_R^{\upsilon_2 *}},\, f(h_i)^{\widetilde{\beta}_R^{\upsilon_3 *}},\, f(h_i)^{\widetilde{\beta}_R^{\upsilon_4 *}}\right) \end{bmatrix}, \tag{3}$$

where the superscripts $\widetilde{\beta}_R^{\upsilon_j}$ in (3) serve as symbolic identifiers of fuzzy association and are not used as exponents.

The corresponding fuzzy data sets can be arranged as follows:

$$\begin{bmatrix} h_i & h_1 & h_2 & \cdots & h_m \\ f_{m*}(h_i) & f_{m*}(h_1) & f_{m*}(h_2) & \cdots & f_{m*}(h_m) \end{bmatrix}, \quad \begin{bmatrix} h_i & h_1 & h_2 & \cdots & h_m \\ f_{n*}(h_i) & f_{n*}(h_1) & f_{n*}(h_2) & \cdots & f_{n*}(h_m) \end{bmatrix}. \tag{4}$$

For the approximation process, let us select a function class $S \subset C[a, b]$ defined as the span of linearly independent basis functions,

$$S = \text{Span}\{\varphi_0(h),\, \varphi_1(h),\, \ldots,\, \varphi_m(h)\}. \tag{5}$$

Furthermore, let $H = \{h_1, h_2, \ldots, h_m\}$ be a set of $m$ distinct points in $\mathbb{R}$ with $m > n$, ensuring that the system is overdetermined and thus suitable for least squares approximation. We now introduce the fuzzy least squares approximation polynomial $\widetilde{p}(h_i) = (p_{m*}(h_i),\, p_{n*}(h_i))$ corresponding to the membership function $(p_{m*}(h))$ of the TLDFN data, given by

$$p_{m*}(h) = \begin{pmatrix} p(h_i),\, p(h_i)^{\widetilde{\beta}_R^{\upsilon_1}},\, p(h_i)^{\widetilde{\beta}_R^{\upsilon_2}}, \\ p(h_i)^{\widetilde{\beta}_R^{\upsilon_3}},\, p(h_i)^{\widetilde{\beta}_R^{\upsilon_4}} \end{pmatrix} = \begin{pmatrix} \sum_{j=0}^{m} a_j\varphi_j(h),\, \sum_{j=0}^{m} b_j\varphi_j(h),\, \sum_{j=0}^{m} c_j\varphi_j(h), \\ \sum_{j=0}^{m} d_j\varphi_j(h),\, \sum_{j=0}^{m} e_j\varphi_j(h) \end{pmatrix}$$

and the polynomial approximation corresponding to the nonmembership function $(p_{n*}(h))$ is given as:

$$p_{n*}(h) = \begin{pmatrix} p_*(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h),\, p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h), \\ p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h),\, p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h) \end{pmatrix} = \begin{pmatrix} \sum_{j=0}^{m} a'_j \varphi_j(h),\, \sum_{j=0}^{m} b'_j \varphi_j(h),\, \sum_{j=0}^{m} c'_j \varphi_j(h), \\ \sum_{j=0}^{m} d'_j \varphi_j(h),\, \sum_{j=0}^{m} e'_j \varphi_j(h) \end{pmatrix}.$$

This formulation establishes the foundation of the TLDFLSA, where the approximation polynomial incorporates both the crisp and fuzzy components, ensuring that uncertainty is systematically preserved within the least squares framework.

The fuzzy least squares approximation seeks to minimize the weighted sum of squared deviations between the observed TLDFN data and the approximating polynomial. This objective can be expressed as

$$\sum_{i=0}^{m} \Psi_i \delta_i^2 = \sum_{i=0}^{m} \Psi_i \left[\widetilde{f}(h_i) - \widetilde{p}(h_i)\right]^2, \tag{6}$$

$$= \underbrace{\min_{p(h_i),\, p(h_i)^{\widetilde{\beta}_R^{\upsilon_1}},\, \cdots p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h),}} \sum_{i=0}^{m} \Psi_i \left[\widetilde{f}(h_i) - \widetilde{p}(h_i)\right]^2 \tag{7}$$

with respect to the components of the approximation polynomial

$$\widetilde{p}(h_i) = p(h_i),\; p(h_i)^{\widetilde{\beta}_R^{\upsilon_1}},\; p(h_i)^{\widetilde{\beta}_R^{\upsilon_2}},\; p(h_i)^{\widetilde{\beta}_R^{\upsilon_3}},\; p(h_i)^{\widetilde{\beta}_R^{\upsilon_4}},\, p_*(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h),\, p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h),\, p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h),\, p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h),$$

where $\Psi_i > 0$ denote the weight functions associated with each data point. By minimizing this functional, the fuzzy least squares approximation polynomial $p^*(h)$ provides the optimal estimation of the underlying TLDFN data within the least squares framework.

**Lemma 2** ([24]) The canonical system $Ga = o$ has a unique solution $(a_0, a_1, \ldots, a_n)$. The corresponding function $p(h) = \sum_{j=0}^{m} a_j \varphi_j(h)$ satisfies equation (9) and is thus the least squares solution for the data set $(h_i, f(h_i))$.

**Theorem 1** The five canonical systems corresponding to membership function of the TLDFN $Ga = o$, $Gb = p$, $Gc = q$, $Gd = r$, and $Ge = s$ admit unique solutions

$$\left[(a_0, a_1, \ldots, a_n), (b_0, b_1, \ldots, b_n), (c_0, c_1, \ldots, c_n), (d_0, d_1, \ldots, d_n), (e_0, e_1, \ldots, e_n)\right].$$

The corresponding approximation functions are given by

$$\begin{aligned} p(h) &= \sum_{j=0}^{n} a_j \varphi_j(h), \qquad & p^{\widetilde{\beta}_R^{\upsilon_1}}(h) &= \sum_{j=0}^{n} b_j \varphi_j(h), \\ p^{\widetilde{\beta}_R^{\upsilon_2}}(h) &= \sum_{j=0}^{n} c_j \varphi_j(h), & p^{\widetilde{\beta}_R^{\upsilon_3}}(h) &= \sum_{j=0}^{n} d_j \varphi_j(h), \\ p^{\widetilde{\beta}_R^{\upsilon_4}}(h) &= \sum_{j=0}^{n} e_j \varphi_j(h), \end{aligned} \tag{8}$$

which satisfy equations (9-13). Therefore, $p_{m*}(h)$ represents the least squares estimated solution for the data set $(h_i, f^*(h_i))$.

Suppose

$$p_{m*}(h) = \left(p(h_i),\, p(h)^{\widetilde{\beta}_R^{\upsilon_1}},\, p(h)^{\widetilde{\beta}_R^{\upsilon_2}},\, p(h)^{\widetilde{\beta}_R^{\upsilon_3}},\, p(h)^{\widetilde{\beta}_R^{\upsilon_4}}\right)$$

then

$$p(h) = (a_0\varphi_0(h) + a_1\varphi_1(h) + \cdots + a_n\varphi_n(h) \tag{9}$$

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = b_0\varphi_0(h) + b_1\varphi_1(h) + \cdots + b_n\varphi_n(h) \tag{10}$$

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = c_0\varphi_0(h) + c_1\varphi_1(h) + \cdots + c_n\varphi_n(h) \tag{11}$$

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = d_0\varphi_0(h) + d_1\varphi_1(h) + \cdots + d_n\varphi_n(h) \tag{12}$$

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h) = e_0\varphi_0(h) + e_1\varphi_1(h) + \cdots + e_n\varphi_n(h) \tag{13}$$

Based on least squares solution,

$$p(h) = (a_0\varphi_0(h) + a_1\varphi_1(h) + \cdots + a_n\varphi_n(h)),$$

is satisfied with (9). Based on the least squares formulation, the coefficients $a_0, a_1, \ldots, a_n$ (and similarly for $b_j, c_j, d_j, e_j$) are determined by minimizing the total squared deviation between the observed TLDFN data and the approximation functions. Consequently, $p(h)$ satisfies (9), providing the least squares solution for the membership function, while analogous reasoning applies for the non-membership function. Hence, $\widetilde{p}(h_i)$ represents the least squares estimated solution for the entire fuzzy data set.

In fact, we resolve the coefficient $a_0, a_1, \ldots, a_n$, and it can transform into solve the minimum points $(a_0, a_1, \ldots, a_n)$ of the multivariate functions $G(a_0, a_1, \ldots, a_n)$;

$$G(a_0, a_1, \ldots, a_n) = \sum_{i=0}^{m} \Psi_i \left[f(h) - \sum_{j=0}^{n} a_j\varphi_j(h_i)\right]^2, \tag{14}$$

and we are able to get $\frac{\partial G}{\partial a_j} = 0,\, k = 0, 1, \ldots, n$:

$$\frac{\partial G}{\partial a_j} = 2\sum_{i=0}^{n} \Psi_i \left[f(h) - \sum_{j=0}^{m} a_j\varphi_j(h_i)\right] \varphi_k(h_i) = 0, \tag{15}$$

$$\sum_{j=0}^{n} a_j \sum_{i=1}^{m} \Psi_i \varphi_j(h_j) \varphi_k(h_i) = \sum_{i=0}^{m} \Psi_i f(h_i) \varphi_k(h_i), \tag{16}$$

The inner product defined on the discrete point set implies that equation (16) can be expressed as

$$\sum_{j=0}^{n} (\varphi_j, \varphi_k) a_j = (f, \varphi_k), \qquad (k = 0, 1, \ldots, n). \tag{17}$$

In matrix form $Ga = o$ is written as:

$$\begin{pmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \cdots & (\varphi_0, \varphi_n) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) & \cdots & (\varphi_1, \varphi_n) \\ \vdots & \vdots & \vdots & \vdots \\ (\varphi_m, \varphi_0) & (\varphi_m, \varphi_1) & \cdots & (\varphi_n, \varphi_n) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} (f, \varphi_0) \\ (f, \varphi_1) \\ \vdots \\ (f, \varphi_n) \end{pmatrix}, \tag{18}$$

When the basis functions $(\varphi_0(h), \varphi_1(h), \ldots, \varphi_n(h))$ are linearly independent, the corresponding Gram matrix satisfies $\det(G_{n+1}) \neq 0$. These base functions $\varphi_0(h)$, $\varphi_1(h)$, $\ldots$, $\varphi_n(h)$ are not fuzzy functions but rather deterministic basis functions (such as polynomial, exponential, or orthogonal functions) employed to construct the approximation $p_*(h)$. Therefore, the linear system has a unique solution for the coefficients $a_0$, $a_1$, $\ldots$, $a_n$, ensuring that the least squares estimated solution exists. Consequently, the unique approximation function is given by

$$p(h) = a_0 \varphi_0(h) + a_1 \varphi_1(h) + \cdots + a_n \varphi_n(h),$$

which satisfies equation (18).

For any $p^*(h) \in S$, we can expand

$$f(h_i) - p^*(h_i) = \big(f(h_i) - p(h_i)\big) + \big(p(h_i) - p^*(h_i)\big).$$

Squaring, multiplying by $\Psi_i$, and summing gives

$$\sum_{i=0}^{m} \Psi_i [f(h_i) - p^*(h_i)]^2 = \sum_{i=0}^{m} \Psi_i [f(h_i) - p(h_i)]^2 + 2 \sum_{i=0}^{m} \Psi_i (f(h_i) - p(h_i))(p(h_i) - p^*(h_i))$$
$$+ \sum_{i=0}^{m} \Psi_i [p(h_i) - p^*(h_i)]^2. \tag{19}$$

Since $p(h)$ is the minimizer of the functional (14), it satisfies the orthogonality condition

$$\sum_{i=0}^{m} \Psi_i [f(h_i) - p(h_i)] \varphi_j(h_i) = 0, \qquad j = 0, 1, \ldots, n, \tag{20}$$

which implies that

$$\sum_{i=0}^{m} \Psi_i [f(h_i) - p(h_i)][p(h_i) - p^*(h_i)] = 0. \tag{21}$$

Substituting (21) into (19) gives

$$\sum_{i=0}^{m} \Psi_i [f(h_i) - p^*(h_i)]^2 = \sum_{i=0}^{m} \Psi_i [f(h_i) - p(h_i)]^2 + \sum_{i=0}^{m} \Psi_i [p(h_i) - p^*(h_i)]^2. \tag{22}$$

Because the last term is nonnegative, we have

$$\sum_{i=0}^{m} \Psi_i [f(h_i) - p^*(h_i)]^2 \geq \sum_{i=0}^{m} \Psi_i [f(h_i) - p(h_i)]^2.$$

Since (22) holds for every $p \in S$, it immediately follows that $p^*$ gives the minimum of the weighted residual norm, i.e.,

$$\sum_{i=0}^{m} \Psi_i [f(h_i) - p^*(h_i)]^2 = \min_{p(x) \in S} \sum_{i=0}^{m} \Psi_i [f(h_i) - p(h_i)]^2. \tag{23}$$

Therefore, expression (23) directly follows from the orthogonality condition and the inequality established in (22).
By expanding $[f - p^*]^2 = [f - p + (p - p^*)]^2$, applying the discrete inner-product orthogonality of the least-squares solution, and eliminating the cross term, we obtain the decomposition in (22). This establishes that the residual norm at $p^*$ is minimal, thus confirming that equation (23) follows naturally from (22).

Consequently, $P(x)$ represents the least squares solution of the data set $(h_i,\ f(h_i))$, $i = 1, 2, \ldots, n$, satisfying (19). Similarly, the coefficient vectors $Gb = o$, $Gc = p$, $Gd = q$, and $Ge = s$ yield the unique parameter sets

$$a_j, b_j, c_j, d_j, e_j, \quad j = 1, 2, \ldots, n, \tag{24}$$

which define the membership function components as

$$
\begin{bmatrix}
p(h) = \sum_{j=0}^{n} a_j \varphi_j(h) \\
p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = \sum_{j=0}^{n} b_j \varphi_j(h) \\
p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = \sum_{j=0}^{n} c_j \varphi_j(h) \\
p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = \sum_{j=0}^{n} d_j \varphi_j(h) \\
p^{\widetilde{\beta}_R^{\upsilon_4}}(h) = \sum_{j=0}^{n} e_j \varphi_j(h)
\end{bmatrix}. \tag{25}
$$

These expressions provide least squares approximations for the fuzzy data sets $\{(h_i,\, f^{\widetilde{\beta}_R^{\upsilon_j}}(h_i))\}_{i=1}^{n}$, and analogous forms hold for the nonmembership components satisfying (23).

The resulting fuzzy least squares polynomial is thus

$$
\widetilde{p}(h_i) = \left( p(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_1}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_2}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_3}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_4}}(h_i),\, p_*(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h_i) \right). \tag{26}
$$

The quality of the approximation is assessed using the minimum square error and maximum deviation metrics for each fuzzy component:

$$
\omega_1^{(\cdot)} = \sqrt{\sum_{i=1}^{n} \Psi_i \big( f(h_i) - p^{(\cdot)}(h_i) \big)^2}, \tag{27}
$$

$$
\omega_2^{(\cdot)} = \max_{1 \le i \le n} \left| f(h_i) - p^{(\cdot)}(h_i) \right|, \tag{28}
$$

where $(\cdot)$ denotes any of the fuzzy components $\{1,\, \widetilde{\beta}_R^{\upsilon_1},\, \widetilde{\beta}_R^{\upsilon_2},\, \widetilde{\beta}_R^{\upsilon_3},\, \widetilde{\beta}_R^{\upsilon_4}\}$ for both membership and nonmembership functions.

These measures collectively evaluate the overall deviation and pointwise accuracy of the fuzzy least squares approximation.

## 3. Solving the Fuzzy Least Squares Approximation (FLSA) using TFLD polynomial

In order to address the problem of fuzzy least squares approximation using polynomials, we employ the least squares principle. This principle provides a systematic approach for minimizing the sum of squared deviations between the observed fuzzy data points and the values predicted by the approximating polynomial. By applying this methodology, the constructed fuzzy polynomial not only ensures the best possible fit in the least squares sense but also preserves the inherent uncertainty and imprecision present in the fuzzy data $h_i$ and $f^*(h_i)$, where $h_i \in \mathbb{R}$, $h_1 < h_2 \cdots < h_n$ and

$$\widetilde{f}(h_i) = \begin{bmatrix} f_{m*}(h_i) \\ \\ f_{n*}(h_i) \end{bmatrix} = \begin{bmatrix} \left(f(h_i),\ f(h_i)^{\widetilde{\beta}_R^{\upsilon_1}},\ f(h_i)^{\widetilde{\beta}_R^{\upsilon_2}},\ f(h_i)^{\widetilde{\beta}_R^{\upsilon_3}},\ f(h_i)^{\widetilde{\beta}_R^{\upsilon_4}}\right) \\ \\ \left(f_*(h_i),\ f(h_i)^{\widetilde{\beta}_R^{\upsilon_1 *}},\ f(h_i)^{\widetilde{\beta}_R^{\upsilon_2 *}},\ f(h_i)^{\widetilde{\beta}_R^{\upsilon_3 *}},\ f(h_i)^{\widetilde{\beta}_R^{\upsilon_4 *}}\right) \end{bmatrix}, \tag{29}$$

such that

$$\begin{bmatrix} h_i & h_1 & h_2 & \cdots & h_m \\ \\ f_{m*}(h_i) & f_{m*}(h_1) & f_{m*}(h_2) & \cdots & f_{m*}(h_m) \end{bmatrix} \tag{30}$$

and

$$\begin{bmatrix} h_i & h_1 & h_2 & \cdots & h_m \\ \\ f_{n*}(h_i) & f_{n*}(h_1) & f_{n*}(h_2) & \cdots & f_{n*}(h_m) \end{bmatrix} \tag{31}$$

More specifically, we typically select S as the basis $\{1,\ h,\ h^2,\ \ldots,\ h^m\}$ and the weight coefficient is $\Psi_i = 1$ $(i = 1, 2, \ldots, m)$.

The fuzzy least squares approximation polynomial of membership function $p_{m*}(h)$ satisfies

$$p_{m*}(h) = f_{m*}(h_i), \quad i = 0,\ 1,\ \ldots,\ m. \tag{32}$$

Thus, $p_{m*}(h)$ can be approximated by applying the least squares principle to five polynomials, namely

$$\left(f(h),\ f(h_i)^{\widetilde{\beta}_R^{\upsilon_1}},\ f(h_i)^{\widetilde{\beta}_R^{\upsilon_2}},\ f(h_i)^{\widetilde{\beta}_R^{\upsilon_3}},\ f(h_i)^{\widetilde{\beta}_R^{\upsilon_4}}\right),$$

which can be carried out in five steps.

To determine the approximation polynomial $p(h)$, we start with the given data set

$$\begin{bmatrix} h_i & h_1 & h_2 & \cdots & h_m \\ \\ f(h_i) & f(h_1) & f(h_2) & \cdots & f(h_m) \end{bmatrix} \tag{33}$$

and assume a polynomial of the form

$$p(h) = a_0 + a_1 h + a_2 h^2 + \cdots + a_m h^m. \tag{34a}$$

The next step involves computing the necessary summations for the least squares formulation, namely

$$\sum_{i=1}^{m}\Psi_i,\ \sum_{i=1}^{m}h_i,\ \sum_{i=1}^{m}h_i^2,\ \ldots,\ \sum_{i=1}^{m}h_i^n,\ \sum_{i=1}^{m}h_i^{n+1},\ \ldots,\ \sum_{i=1}^{m}h_i^{2n},\ \sum_{i=1}^{m}f(h_i),\ \sum_{i=1}^{m}h_if(h_i),\ \sum_{i=1}^{m}h_i^2f(h_i),\ \ldots,\ \sum_{i=1}^{m}h_i^nf(h_i). \tag{35}$$

Finally, by solving the normal equations

$$\begin{pmatrix} \sum_{i=1}^{m}1 & \sum_{i=1}^{m}h_i & \ldots & \sum_{i=1}^{m}h_i^n \\ \sum_{i=1}^{m}h_i & \sum_{i=1}^{m}h_i^2 & \ldots & \sum_{i=1}^{m}h_i^{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{m}h_i^n & \sum_{i=1}^{m}h_i^{n+1} & \ldots & \sum_{i=1}^{m}h_i^{2n} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m}f(h_i) \\ \sum_{i=1}^{m}h_if(h_i) \\ \vdots \\ \sum_{i=1}^{m}h_i^nf(h_i) \end{pmatrix}, \tag{36}$$

we obtain the coefficients $a_0$, $a_1$, ..., $a_n$, yielding the approximation polynomial

$$p(h) = a_0 + a_1h + a_2h^2 + \cdots + a_nh^n.$$

Similarly, to compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_1}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & h_1 & h_2 & \cdots & h_m \\ f^{\widetilde{\beta}_R^{\upsilon_1}}(h_i) & f^{\widetilde{\beta}_R^{\upsilon_1}}(h_1) & f^{\widetilde{\beta}_R^{\upsilon_1}}(h_2) & \cdots & f^{\widetilde{\beta}_R^{\upsilon_1}}(h_m) \end{bmatrix}. \tag{37}$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = b_0 + b_1h + b_2h^2 + \cdots + b_mh^m. \tag{38}$$

Then, calculate

$$\left[\sum_{i=1}^{m}f(h_i),\ \sum_{i=1}^{m}h_if(h_i),\ \sum_{i=1}^{m}h_i^2f^{\widetilde{\beta}_R^{\upsilon_1}}(h_i)\cdots\sum_{i=1}^{m}h_i^nf^{\widetilde{\beta}_R^{\upsilon_1}}(h_i)\right].$$

Solving the equation,

$$\begin{pmatrix} \sum\limits_{i=1}^{m} 1 & \sum\limits_{i=1}^{m} h_i & \cdots & \sum\limits_{i=1}^{m} h_i^n \\ \sum\limits_{i=1}^{m} h_i & \sum\limits_{i=1}^{m} h_i^2 & \cdots & \sum\limits_{i=1}^{m} h_i^{n+1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum\limits_{i=1}^{m} h^n & \sum\limits_{i=1}^{m} h_i^{n+1} & \cdots & \sum\limits_{i=1}^{n} h_i^{2n} \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} \sum\limits_{i=1}^{m} f^{\widetilde{\beta}_R^{\upsilon_1}}(h_i) \\ \sum\limits_{i=1}^{m} h_i f^{\widetilde{\beta}_R^{\upsilon_1}}(h_i) \\ \vdots \\ \sum\limits_{i=1}^{m} h_i^n f(h_i) \end{pmatrix}, \tag{40}$$

we obtain the coefficients $b_0, b_1, \ldots, b_n$, yielding the approximation polynomial

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = b_0 + b_1 h + b_2 h^2 + \cdots + b_n h^n.$$

Similarly, to compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_2}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & h_1 & h_2 & \cdots & h_m \\ f^{\widetilde{\beta}_R^{\upsilon_2}}(h_i) & f^{\widetilde{\beta}_R^{\upsilon_2}}(h_1) & f^{\widetilde{\beta}_R^{\upsilon_2}}(h_2) & \cdots & f^{\widetilde{\beta}_R^{\upsilon_2}}(h_m) \end{bmatrix} \tag{41}$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = c_0 + c_1 h + c_2 h^2 + \cdots + c_m h^m. \tag{42}$$

Then, calculate

$$\left[ \sum_{i=1}^{m} f(h_i), \sum_{i=1}^{m} h_i f(h_i), \sum_{i=1}^{m} h_i^2 f^{\widetilde{\beta}_R^{\upsilon_2}}(h_i) \cdots \sum_{i=1}^{m} h_i^n f^{\widetilde{\beta}_R^{\upsilon_2}}(h_i) \right]. \tag{43}$$

Resolving the equation,

$$\begin{pmatrix} \sum\limits_{i=1}^{m} 1 & \sum\limits_{i=1}^{m} h_i & \cdots & \sum\limits_{i=1}^{m} h_i^n \\ \sum\limits_{i=1}^{m} h_i & \sum\limits_{i=1}^{m} h_i^2 & \cdots & \sum\limits_{i=1}^{m} h_i^{n+1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum\limits_{i=1}^{m} h^n & \sum\limits_{i=1}^{m} h_i^{n+1} & \cdots & \sum\limits_{i=1}^{m} h_i^{2n} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_m \end{pmatrix} = \begin{pmatrix} \sum\limits_{i=1}^{m} f^{\widetilde{\beta}_R^{\upsilon_2}}(h_i) \\ \sum\limits_{i=1}^{m} h_i f^{\widetilde{\beta}_R^{\upsilon_2}}(h_i) \\ \vdots \\ \sum\limits_{i=1}^{m} h_i^n f^{\widetilde{\beta}_R^{\upsilon_2}}(h_i) \end{pmatrix} \tag{44}$$

we obtain the coefficients $c_0$, $c_1$, …, $c_n$, yielding the approximation polynomial

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = c_0 + c_1 h + c_2 h^2 + \cdots + c_m h^m.$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_3}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & h_1 & h_2 & \cdots & h_m \\ f^{\widetilde{\beta}_R^{\upsilon_3}}(h_i) & f^{\widetilde{\beta}_R^{\upsilon_3}}(h_1) & f^{\widetilde{\beta}_R^{\upsilon_3}}(h_2) & \cdots & f^{\widetilde{\beta}_R^{\upsilon_3}}(h_m) \end{bmatrix}. \tag{45}$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = d_0 + d_1 h + d_2 h^2 + \cdots + d_m h^m. \tag{46}$$

Then, calculate

$$\left[\sum_{i=1}^{m} f(h_i), \sum_{i=1}^{m} h_i f(h_i), \sum_{i=1}^{m} h_i^2 f^{\widetilde{\beta}_R^{\upsilon_3}}(h_i) \cdots \sum_{i=1}^{m} h_i^n f^{\widetilde{\beta}_R^{\upsilon_3}}(h_i)\right]. \tag{47}$$

Resolving the system,

$$\begin{pmatrix} \sum_{i=1}^{m} 1 & \sum_{i=1}^{m} h_i & \cdots & \sum_{i=1}^{n} h_i^n \\ \sum_{i=1}^{m} h_i & \sum_{i=1}^{m} h_i^2 & \cdots & \sum_{i=1}^{m} h_i^{n+1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{m} h^n & \sum_{i=1}^{m} h_i^{n+1} & \cdots & \sum_{i=1}^{m} h_i^{2n} \end{pmatrix} \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ d_m \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m} f^{\widetilde{\beta}_R^{\upsilon_3}}(h_i) \\ \sum_{i=1}^{m} h_i f^{\widetilde{\beta}_R^{\upsilon_3}}(h_i) \\ \vdots \\ \sum_{i=1}^{m} h_i^n f^{\widetilde{\beta}_R^{\upsilon_3}}(h_i) \end{pmatrix} \tag{48}$$

we obtain the coefficients $d_0$, $d_1$, …, $d_n$, yielding the approximation polynomial

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = d_0 + d_1 h + d_2 h^2 + \cdots + d_n h^n.$$

Finally, to compute the last fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_4}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & h_1 & h_2 & \cdots & h_m \\ f^{\widetilde{\beta}_R^{\upsilon_4}}(h_i) & f^{\widetilde{\beta}_R^{\upsilon_4}}(h_1) & f^{\widetilde{\beta}_R^{\upsilon_4}} h_2) & \cdots & f^{\widetilde{\beta}_R^{\upsilon_4}}(h_m) \end{bmatrix} \tag{49}$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h) = e_0 + e_1 h + e_2 h^2 + \cdots + e_m h^m. \tag{50}$$

Then, calculate

$$\left[\sum_{i=1}^{m} f(h_i),\, \sum_{i=1}^{m} h_i f(h_i),\, \sum_{i=1}^{m} h_i^2 f^{\widetilde{\beta}_R^{\upsilon_4}}(h_i) \cdots \sum_{i=1}^{m} h_i^{\widetilde{\beta}_R^{\upsilon_4}} f(h_i)\right]. \tag{51}$$

Solving the resulting system of equations

$$\begin{pmatrix} \sum_{i=1}^{m} 1 & \sum_{i=1}^{m} h_i & \cdots & \sum_{i=1}^{m} h_i^n \\ \sum_{i=1}^{m} h_i & \sum_{i=1}^{m} h_i^2 & \cdots & \sum_{i=1}^{m} h_i^{n+1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{m} h^n & \sum_{i=1}^{m} h_i^{n+1} & \cdots & \sum_{i=1}^{m} h_i^{2n} \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m} f^{\widetilde{\beta}_R^{\upsilon_4}}(h_i) \\ \sum_{i=1}^{m} h_i f^{\widetilde{\beta}_R^{\upsilon_4}}(h_i) \\ \vdots \\ \sum_{i=1}^{m} h_i^n f^{\widetilde{\beta}_R^{\upsilon_4}}(h_i) \end{pmatrix}, \tag{52}$$

yields the coefficients $e_0, e_1, \ldots, e_m$, giving the polynomial

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h) = e_0 + e_1 h + e_2 h^2 + \cdots + e_n h^n.$$

The final fuzzy least squares approximation polynomial is then expressed as

$$p_{m*}(h) \approx \left(f(h),\, f^{\widetilde{\beta}_R^{\upsilon_1}}(h),\, f^{\widetilde{\beta}_R^{\upsilon_2}}(h),\, f^{\widetilde{\beta}_R^{\upsilon_3}}(h),\, f^{\widetilde{\beta}_R^{\upsilon_4}}(h)\right), \tag{53}$$

and similarly for the non-membership function. Thus, the Two-Layer Discrete Fuzzy Network (TLDFN) polynomial can be written compactly as:

$$\widetilde{p}(h_i) = \begin{bmatrix} p(h_i),\, p(h_i)^{\widetilde{\beta}_R^{\upsilon_1}},\, p(h_i)^{\widetilde{\beta}_R^{\upsilon_2}},\, p(h_i)^{\widetilde{\beta}_R^{\upsilon_3}},\, p(h_i)^{\widetilde{\beta}_R^{\upsilon_4}}, \\ p_*(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h_i) \end{bmatrix}. \tag{54}$$

This formulation provides a compact and consistent least squares-based construction of the TLDFN approximation polynomial, ensuring both optimality and preservation of fuzzy uncertainty.

## 4. Numerical outcomes: real-world applications

In this section, we demonstrate the effectiveness of the Triangular Linear Diophantine (TLD) Least Squares approximation polynomial in addressing several real-world problems. The suggested fuzzy framework is especially well-suited for triangular fuzzy data, trapezoidal fuzzy data, or intuitionistic fuzzy data sets due to its generalized fuzzy set structure, which captures higher degrees of uncertainty and ambiguity in the data . This enables the suggested system to better describe complicated and imprecise data while preserving processing throughput, computational economy, and reliability. Three widely used methods that exist are—Weighted Least Squares approximation method (WLS) [29], the Total Least Squares approximation technique (TLS) [30], and Ridge Regression (RRM) [31]—are analyzed and contrasted with the proposed Triangular Linear Diophantine Fuzzy Least Squares. These contemporary numerical algorithms are commonly employed to effectively deal with uncertainty and fuzzy data using the flowchart shown in Figure 2. The comparison with various methods shows that the proposed TLDFLS formulation improves accuracy, durability, and computing supremacy, making it the best option for handling fuzzy data more efficiently.

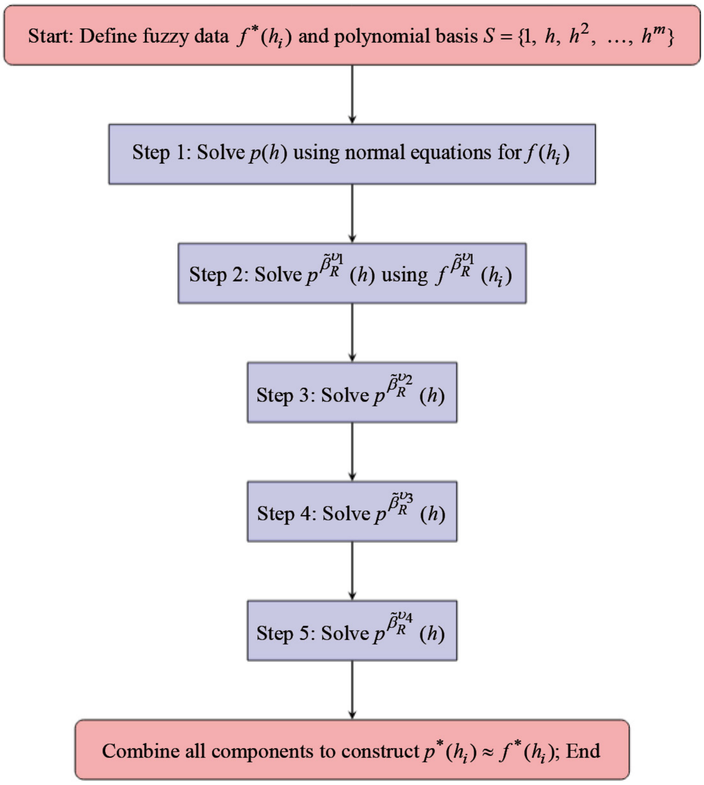


**Figure 2.** Flowchart for solving triangular linear Diophantine fuzzy least square approximation

To construct the approximation, we follow the least squares principle, generating the fuzzy polynomial

$$\widetilde{p}(h_i) \approx \widetilde{f}(h_i)) = \begin{pmatrix} f(h),\, f^{\widetilde{\beta}_R^{\upsilon_1}}(h),\, f(h)^{\widetilde{\beta}_R^{\upsilon_2}},\, f(h)^{\widetilde{\beta}_R^{\upsilon_3}},\, f(h)^{\widetilde{\beta}_R^{\upsilon_4}} \\ \\ f_*(h),\, f^{\widetilde{\beta}_R^{\upsilon_1*}}(h),\, f(h)^{\widetilde{\beta}_R^{\upsilon_2*}},\, f(h)^{\widetilde{\beta}_R^{\upsilon_3*}},\, f(h)^{\widetilde{\beta}_R^{\upsilon_4*}} \end{pmatrix}, \quad i = 0,\, 1,\, \ldots,\, m. \tag{55}$$

Some fitting functions in practical scenarios may exhibit hyperbolic or exponential behavior, represented by

$$p(x) = ae^{b/t}, \quad \text{or} \quad p(x) = \frac{t}{at+b}, \tag{56}$$

which, through the substitution

$$Y = \frac{1}{y}, \quad X = \frac{1}{t}, \tag{57}$$

can be transformed into a linear fitting problem. To simplify computations while maintaining accuracy, a low-order polynomial fitting strategy is applied:

$$S(X) = A + BX. \tag{58}$$

To quantitatively assess the performance of the TFLD Least Squares approximation, we compute several standard statistical metrics:

• **Mean Squared Error (MSE):**

$$\text{MSE} = \frac{1}{m+1} \sum_{i=0}^{m} \left[\widetilde{f}(h_i) - \widetilde{p}(h_i)\right]^2. \tag{59}$$

• **Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\text{MSE}}. \tag{60}$$

• **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{m+1} \sum_{i=0}^{m} \left|\widetilde{f}(h_i) - \widetilde{p}(h_i)\right|. \tag{61}$$

• **Standard Deviation (SD) of errors:**

$$\text{SD} = \sqrt{\frac{1}{m} \sum_{i=0}^{m} \left(\left[\widetilde{f}(h_i) - \widetilde{p}(h_i)\right] - \overline{e}\right)^2}, \quad \overline{e} = \frac{1}{m+1} \sum_{i=0}^{m} \left[\widetilde{f}(h_i) - \widetilde{p}(h_i)\right]. \tag{62}$$

These metrics provide a comprehensive evaluation of the approximation accuracy and robustness. In the following subsections, we present numerical outcomes for selected real-world problems, illustrating how the TFLD Least Squares approximation effectively captures uncertainty and delivers reliable predictions even for complex and imprecise datasets.

**Example 1:** The dataset presented to the Senate Antitrust Subcommittee examines the crash-survivability characteristics of automobiles across various categories [32]. Understanding the safety performance of different car classes is critical for policy-making and automotive safety analysis. The dataset summarizes the proportion of vehicles involved in collisions in which the most severe injuries were either fatal or serious. To analyze and interpret this data effectively, it is important to identify a trend that accurately captures the underlying relationship between the variables. The least squares method is employed for this purpose, as it provides an unbiased numerical approximation and is well-suited for fitting such data.

The dataset is organized as follows:

$$\begin{bmatrix} \text{Classes} & \text{Type} & \text{Percent Occurrence} \\ 0 & \text{D1:} & \left\langle \begin{matrix} 1 & 1 & 2 & 3 & 4 \\ -1 & -1 & 2 & 3 & 3 \end{matrix} \right\rangle \\ 1 & \text{D2} & \left\langle \begin{matrix} -2 & 0 & 1 & 3 & 3 \\ -1 & 0 & 1 & 2 & 3 \end{matrix} \right\rangle \\ 2 & \text{D3} & \left\langle \begin{matrix} 1 & 1 & 3 & 3 & 4 \\ 0 & 1 & 3 & 4 & 5 \end{matrix} \right\rangle \\ 3 & \text{D4} & \left\langle \begin{matrix} 1 & 3 & 3 & 5 & 7 \\ 1 & 2 & 3 & 7 & 7 \end{matrix} \right\rangle \\ 4 & \text{D5} & \left\langle \begin{matrix} -1 & 0 & 1 & 2 & 3 \\ -2 & 0 & 1 & 2 & 3 \end{matrix} \right\rangle \end{bmatrix}$$

Here, D1 represents Domestic Luxury Regular, D2 Domestic Intermediate Regular, D3 Domestic Economy Regular, D4 Domestic Compact, and D5 Foreign Compact. The corresponding fuzzy least squares data points (Figure 3) are as follows:

$$\begin{bmatrix} h_i & f^*(h_i) \\ 0 & \left\langle \begin{matrix} 1 & 1 & 2 & 3 & 4 \\ -1 & -1 & 2 & 3 & 3 \end{matrix} \right\rangle \\ 1 & \left\langle \begin{matrix} -2 & 0 & 1 & 3 & 3 \\ -1 & 0 & 1 & 2 & 3 \end{matrix} \right\rangle \\ 2 & \left\langle \begin{matrix} 1 & 1 & 3 & 3 & 4 \\ 0 & 1 & 3 & 4 & 5 \end{matrix} \right\rangle \\ 3 & \left\langle \begin{matrix} 1 & 3 & 3 & 5 & 7 \\ 1 & 2 & 3 & 7 & 7 \end{matrix} \right\rangle \\ 4 & \left\langle \begin{matrix} -1 & 0 & 1 & 2 & 3 \\ -2 & 0 & 1 & 2 & 3 \end{matrix} \right\rangle \end{bmatrix}$$

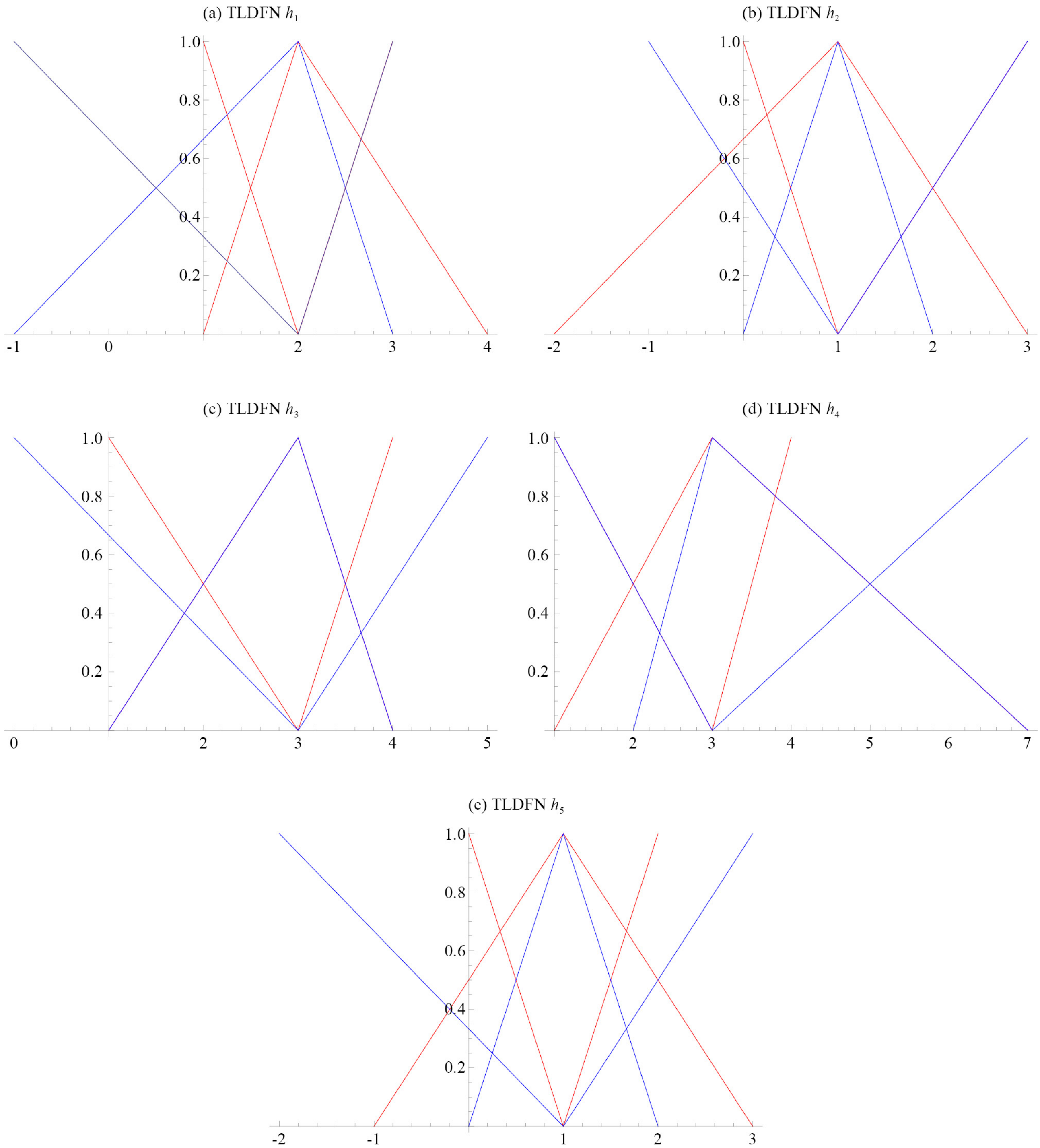


**Figure 3.** (a-e): Example 1's approximate solution using the TLDF least-squares approximation method

For convenience, the central values of the fuzzy numbers can also be summarized in a simplified tabular form:

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 & 4 \\ f(h_i) & 1 & -2 & 1 & 1 & -1 \end{bmatrix}$$

This representation facilitates the application of the fuzzy least squares approximation method to determine a trend that best describes the relationship between vehicle class and crash-survivability characteristics, providing a reliable basis for further analysis and decision-making.

To determine the approximation polynomial $p(h)$,

$$p(h) = a_0 + a_1 h + a_2 h^2 + a_3 h^3 + a_4 h^4. \tag{63}$$

The next step involves computing the necessary summations for the least squares formulation, namely

$$\sum_{i=1}^{5} 1 = 5,\ \sum_{i=1}^{5} h_i = 10,\ \sum_{i=1}^{5} h_i^2 = 30,\ \sum_{i=1}^{5} h_i^3 = 100,\ \sum_{i=1}^{5} h_i^4 = 354,\ \sum_{i=1}^{5} h_i^5 = 1300,\ \sum_{i=1}^{5} h_i^6 = 4890,$$

$$\sum_{i=1}^{5} h_i^7 = 18700,\ \sum_{i=1}^{5} h_i^8 = 72,354,\ \sum_{i=1}^{5} f(h_i) = 0,\ \sum_{i=1}^{5} h_i f(h_i) = -1,$$

$$\sum_{i=1}^{5} h_i^2 f(h_i) = -5,\ \sum_{i=1}^{5} h_i^3 f(h_i) = -31,\ \sum_{i=1}^{5} h_i^4 f(h_i) = -161.$$

Solving the system of equations,

$$\begin{pmatrix} 5 & 10 & 30 & 100 & 354 \\ 10 & 30 & 100 & 354 & 1300 \\ 30 & 100 & 354 & 1300 & 4890 \\ 100 & 354 & 1300 & 4890 & 18700 \\ 354 & 1300 & 4890 & 18700 & 72354 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ -5 \\ -31 \\ -161 \end{pmatrix}.$$

we obtain the coefficients

$$[a_0 = 1.0000,\ a_1 = -11.5000,\ a_2 = 12.0833,\ a_3 = -4.0000,\ a_4 = 0.4167],$$

which, when substituted into Equation (63), yield the following approximation polynomial:

$$p(h) = 1.0000 + 11.5000h + 12.0833h^2 - 4.000h^3 + 0.4167h^4. \tag{64}$$

Similarly, to compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_1}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_1}}(h) & 1 & 0 & 1 & 3 & 0 \end{bmatrix},$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = b_0 + b_1 h + b_2 h^2 + b_3 h^3 + b_4 h^4. \tag{65}$$

Then, calculate

$$\sum_{i=1}^{5} f(h_i) = 5, \sum_{i=1}^{5} h_i f(h_i) = 11, \sum_{i=1}^{5} h_i^2 f(h_i) = 31, \sum_{i=1}^{5} h_i^3 f(h_i) = 89, \sum_{i=1}^{5} h_i^4 f(h_i) = 259.$$

Resolving the system of equations,

$$\begin{pmatrix} 5 & 10 & 30 & 100 & 354 \\ 10 & 30 & 100 & 354 & 1300 \\ 30 & 100 & 354 & 1300 & 4890 \\ 100 & 354 & 1300 & 4890 & 18700 \\ 354 & 1300 & 4890 & 18700 & 72354 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} 5 \\ 11 \\ 31 \\ 89 \\ 259 \end{pmatrix}.$$

we obtain the following coefficients

$$[b_0 = 1.000,\, b_1 = -1.0833,\, b_2 = -0.7917,\, b_3 = 1.0833,\, b_4 = -0.2083],$$

which, when substituted into Equation (65), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = 1 - 1.0833h - 0.7917h^2 + 1.0833h^3 - 0.2083h^4. \tag{66}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_2}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_2}}(h) & 2 & 1 & 3 & 3 & 1 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = c_0 + c_1 h + c_2 h^2 + c_3 h^3 + c_4 h^4. \tag{67}$$

Then, calculate

$$\sum_{i=1}^{5} f(h_i) = 5,\ \sum_{i=1}^{5} h_i f(h_i) = 11,\ \sum_{i=1}^{5} h_i^2 f(h_i) = 31,\ \sum_{i=1}^{5} h_i^3 f(h_i) = 89,\ \sum_{i=1}^{5} h_i^4 f(h_i) = 259.$$

Solving the system of equations,

$$\begin{pmatrix} 5 & 10 & 30 & 100 & 354 \\ 10 & 30 & 100 & 354 & 1300 \\ 30 & 100 & 354 & 1300 & 4890 \\ 100 & 354 & 1300 & 4890 & 18700 \\ 354 & 1300 & 4890 & 18700 & 72354 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} = \begin{pmatrix} 10 \\ 20 \\ 56 \\ 170 \\ 548 \end{pmatrix}.$$

we obtain the following coefficients

$$[c_0 = 2.000,\ c_1 = -5.4167,\ c_2 = 6.2917,\ c_3 = -2.0833,\ c_4 = 0.2083],$$

which, when substituted into Equation (67), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = 2 - 5.4167h + 6.2917h^2 - 2.0833h^3 + 0.2083h^4. \tag{68}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_3}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_3}}(h) & 3 & 3 & 3 & 5 & 2 \end{bmatrix}$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = d_0 + d_1 h + d_2 h^2 + d_3 h^3 + d_4 h^4. \tag{69}$$

Then, compute the following

$$\sum_{i=1}^{5} f(h_i) = 16,\ \sum_{i=1}^{5} h_i f(h_i) = 32,\ \sum_{i=1}^{5} h_i^2 f(h_i) = 92,\ \sum_{i=1}^{5} h_i^3 f(h_i) = 290,\ \sum_{i=1}^{5} h_i^4 f(h_i) = 968.$$

Solving the system of equations,

$$\begin{pmatrix} 5 & 10 & 30 & 100 & 354 \\ 10 & 30 & 100 & 354 & 1300 \\ 30 & 100 & 354 & 1300 & 4890 \\ 100 & 354 & 1300 & 4890 & 18700 \\ 354 & 1300 & 4890 & 18700 & 72354 \end{pmatrix} \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix} = \begin{pmatrix} 16 \\ 32 \\ 92 \\ 290 \\ 968 \end{pmatrix}.$$

we obtain the following

$$[c_0 = 3.000,\ c_1 = 2.9167,\ c_2 = -5.1250,\ c_3 = 2.5833,\ c_4 = -0.375]$$

which, when substituted into Equation (69), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = 3 + 2.9167h - 5.125h^2 + 2.5833h^3 - 0.3750h^4. \tag{70}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_4}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_4}}(h) & 4 & 3 & 4 & 7 & 3 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h) = e_0 + e_1 h + e_2 h^2 + e_3 h^3 + e_4 h^4. \tag{71}$$

Then, compute

$$\sum_{i=1}^{5} f(h_i) = 21, \sum_{i=1}^{5} h_i f(h_i) = 44, \sum_{i=1}^{5} h_i^2 f(h_i) = 130, \sum_{i=1}^{5} h_i^3 f(h_i) = 416, \sum_{i=1}^{5} h_i^4 f(h_i) = 1402.$$

Solving the system of equations,

$$\begin{pmatrix} 5 & 10 & 30 & 100 & 354 \\ 10 & 30 & 100 & 354 & 1300 \\ 30 & 100 & 354 & 1300 & 4890 \\ 100 & 354 & 1300 & 4890 & 18700 \\ 354 & 1300 & 4890 & 18700 & 72354 \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} = \begin{pmatrix} 21 \\ 44 \\ 130 \\ 416 \\ 1402 \end{pmatrix}.$$

we obtain the following coefficient

$$[e_0 = 4.000, e_1 = 0.2500, e_2 = -3.1250, e_3 = 2.2500, e_4 = -0.3750],$$

which, when substituted into Equation (71), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h) = 4 + 0.2500h - 3.1250h^2 + 2.2500h^3 - 0.3750h^4. \tag{72}$$

Next, to compute the fuzzy component $p^*(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 & 4 \\ f_*(h_i) & -1 & -1 & 0 & 1 & -2 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p_*(h) = a_0 + a_1 h + a_2 h^2 + a_3 h^3 + a_4 h^4. \tag{73}$$

We comput the necessary summations for the least squares formulation as

$$\sum_{i=1}^{5} 1 = 5, \sum_{i=1}^{5} h_i = 10, \sum_{i=1}^{5} h_i^2 = 30, \sum_{i=1}^{5} h_i^3 = 100, \sum_{i=1}^{5} h_i^4 = 354, \sum_{i=1}^{5} h_i^5 = 1300, \sum_{i=1}^{5} h_i^6 = 4890,$$

$$\sum_{i=1}^{5} h_i^7 = 18700, \sum_{i=1}^{5} h_i^8 = 72,354, \sum_{i=1}^{5} f(h_i) = -3, \sum_{i=1}^{5} h_i f(h_i) = -6,$$

$$\sum_{i=1}^{5} h_i^2 f(h_i) = -24, \sum_{i=1}^{5} h_i^3 f(h_i) = -102, \sum_{i=1}^{5} h_i^4 f(h_i) = -432.$$

Solving the system of equations

$$\begin{pmatrix} 5 & 10 & 30 & 100 & 354 \\ 10 & 30 & 100 & 354 & 1300 \\ 30 & 100 & 354 & 1300 & 4890 \\ 100 & 354 & 1300 & 4890 & 18700 \\ 354 & 1300 & 4890 & 18700 & 72354 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} -3 \\ -6 \\ -24 \\ -102 \\ -432 \end{pmatrix},$$

we obtain the following coefficient

$$[a_0 = -1.000, a_1 = -0.0833, a_2 = -0.3750, a_3 = 0.5833, a_4 = -0.1250],$$

which, when substituted into Equation (73), yield the following approximation polynomial:

$$p_*(h) = -1 - 0.0833h - 0.3750h^2 + 0.5833h^3 - 0.1250h^4. \quad (74)$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_1 *}}(h) & -1 & 0 & 1 & 2 & 0 \end{bmatrix}$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h) = b_0 + b_1 h + b_2 h^2 + b_3 h^3 + b_4 h^4. \quad (75)$$

By computing the following necessary summations gives

$$\sum_{i=1}^{5} f(h_i) = 2, \sum_{i=1}^{5} h_i f(h_i) = 8, \sum_{i=1}^{5} h_i^2 f(h_i) = 22, \sum_{i=1}^{5} h_i^3 f(h_i) = 62, \sum_{i=1}^{5} h_i^4 f(h_i) = 178.$$

Solving the system of equations,

$$\begin{pmatrix} 5 & 10 & 30 & 100 & 354 \\ 10 & 30 & 100 & 354 & 1300 \\ 30 & 100 & 354 & 1300 & 4890 \\ 100 & 354 & 1300 & 4890 & 18700 \\ 354 & 1300 & 4890 & 18700 & 72354 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 8 \\ 22 \\ 62 \\ 178 \end{pmatrix},$$

we obtain the following coefficient

$$[b_0 = -1.000, \, b_1 = 1.7500, \, b_2 = -1.3750, \, b_3 = 0.7500, \, b_4 = -0.1250, \, ]$$

which, when substituted into Equation (75), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h) = -1 + 1.7500h - 1.3750h^2 + 0.7500h^3 - 0.125h^4. \tag{76}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_2 *}}(h) & 2 & 1 & 3 & 3 & 1 \end{bmatrix}$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h) = c_0 + c_1 h + c_2 h^2 + c_3 h^3 + c_4 h^4. \tag{77}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{5} f(h_i) = 10, \, \sum_{i=1}^{5} h_i f(h_i) = 20, \, \sum_{i=1}^{5} h_i^2 f(h_i) = 56, \, \sum_{i=1}^{5} h_i^3 f(h_i) = 170, \, \sum_{i=1}^{5} h_i^4 f(h_i) = 548.$$

Solving the system of equations,

$$\begin{pmatrix} 5 & 10 & 30 & 100 & 354 \\ 10 & 30 & 100 & 354 & 1300 \\ 30 & 100 & 354 & 1300 & 4890 \\ 100 & 354 & 1300 & 4890 & 18700 \\ 354 & 1300 & 4890 & 18700 & 72354 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} = \begin{pmatrix} 10 \\ 20 \\ 56 \\ 170 \\ 548 \end{pmatrix},$$

we obtain the following coefficient

$$[c_0 = 2.000,\ c_1 = -5.4167,\ c_2 = 6.2917,\ c_3 = -2.0833,\ c_4 = 0.2083],$$

which, when substituted into Equation (65), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h) = 2.0000 - 5.4167h + 6.2917h^2 - 2.0833h^3 + 0.2083h^4. \tag{78}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_3 *}}(h) & 3 & 2 & 4 & 7 & 2 \end{bmatrix}$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h) = d_0 + d_1 h + d_2 h^2 + d_3 h^3 + d_4 h^4. \tag{79}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{5} f(h_i) = 18,\ \sum_{i=1}^{5} h_i f(h_i) = 39,\ \sum_{i=1}^{5} h_i^2 f(h_i) = 113,\ \sum_{i=1}^{5} h_i^3 f(h_i) = 351,\ \sum_{i=1}^{5} h_i^4 f(h_i) = 1145.$$

Solving the system of equations,

$$\begin{pmatrix} 5 & 10 & 30 & 100 & 354 \\ 10 & 30 & 100 & 354 & 1300 \\ 30 & 100 & 354 & 1300 & 4890 \\ 100 & 354 & 1300 & 4890 & 18700 \\ 354 & 1300 & 4890 & 18700 & 72354 \end{pmatrix} \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix} = \begin{pmatrix} 18 \\ 39 \\ 113 \\ 351 \\ 1145 \end{pmatrix}.$$

we obtain the following coefficient

$$[c_0 = 3.000,\ c_1 = -1.4167,\ c_2 = -0.7083,\ c_3 = 1.4167,\ c_4 = -0.2917]$$

which, when substituted into Equation (79), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h) = 3.000 - 1.4167h - 0.7083h^2 + 1.4167h^3 - 0.2917h^4. \tag{80}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_4 *}}(h) & 3 & 3 & 5 & 7 & 3 \end{bmatrix}$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h) = e_0 + e_1 h + e_2 h^2 + e_3 h^3 + e_4 h^4. \tag{81}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{5} f(h_i) = 21,\ \sum_{i=1}^{5} h_i f(h_i) = 46,\ \sum_{i=1}^{5} h_i^2 f(h_i) = 134,\ \sum_{i=1}^{5} h_i^3 f(h_i) = 424,\ \sum_{i=1}^{5} h_i^4 f(h_i) = 1418.$$

Solving the system of equations,

$$\begin{pmatrix} 5 & 10 & 30 & 100 & 354 \\ 10 & 30 & 100 & 354 & 1300 \\ 30 & 100 & 354 & 1300 & 4890 \\ 100 & 354 & 1300 & 4890 & 18700 \\ 354 & 1300 & 4890 & 18700 & 72354 \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} = \begin{pmatrix} 21 \\ 46 \\ 134 \\ 424 \\ 1418 \end{pmatrix}.$$

we obtain the following coefficient

$$[e_0 = 3.000,\, e_1 = -0.6667,\, e_2 = 0.1667,\, e_3 = 0.6667,\, e_4 = -0.1667],$$

which, when substituted into Equation (65), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h) = 3.000 + -0.6667h + 0.1667h^2 + 0.6667h^3 - 0.1667h^4. \tag{82}$$

Consequently, the polynomial used to approximate fuzzy least squares is

$$f_{m*}(h_i) = p_{m*}(h_i) = (p(h),\, p^{\widetilde{\beta}_R^{\upsilon_1}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_2}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_3}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_4}}(h_i), \tag{83}$$

where

$$p(h) = \left(1.000 - 11.5000h + 12.0833h^2 - 4.0000h^3 + 0.4167h^4\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h_i) = \left(1.000 - 1.0833h - 0.7917h^2 - 1.0833h^3 + 0.2083h^4\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h_i) = \left(2.000 - 5.4167h + 6.2917h^2 - 2.0833h^3 + 0.2083h^4\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h_i) = \left(3.000 - 2.9167h - 5.1250h^2 + 2.5833h^3 - 0.3750h^4\right)$$

and

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h_i) = \left(4.000 - 0.2500h - 3.0125h^2 + 2.2500h^3 - 0.3750h^4\right)$$

and corresponding polynomial approximation function is

$$f_{n*}(h_i) = \left[p_*(h),\, p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_4} *}(h_i)\right],$$

where

$$p_*(h) = \left[(-1.000 - 0.0833h - 0.3750h^2 - 0.5833h^3 - 0.125h^4)\right],$$

$$p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h_i) = \left[(-1.000 + 1.7500h - 1.3750h^2 + 0.7500h^3 - 0.1250h^4)\right],$$

$$p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h_i) = \left[(2.000 - 5.4167h + 6.2917h^2 - 2.0833h^3 + 0.2083h^4)\right],$$

$$p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h_i) = \left[(3.000 - 1.4167h - 0.7083h^2 + 1.4167h^3 - 0.2917h^4)\right],$$

$$p^{\widetilde{\beta}_R^{\upsilon_4} *}(h_i) = \left[(3.000 - 0.6667h - 0.1667h^2 + 0.6667h^3 - 0.1667h^4)\right].$$

$$\begin{bmatrix} h_i & \widetilde{f}(h_i) & \widetilde{p}(h_i) \\ 0 & \left\langle \begin{matrix} 1 & 1 & 2 & 3 & 4 \\ -1 & -1 & 2 & 3 & 3 \end{matrix} \right\rangle & \left\langle \begin{matrix} 1 & 1 & 2 & 3 & 4 \\ -1 & -1 & 2 & 3 & 3 \end{matrix} \right\rangle \\ 1 & \left\langle \begin{matrix} -2 & 0 & 1 & 3 & 3 \\ -1 & 0 & 1 & 2 & 3 \end{matrix} \right\rangle & \left\langle \begin{matrix} -2 & 0 & 1 & 3 & 3 \\ -1 & 0 & 1 & 2 & 3 \end{matrix} \right\rangle \\ 2 & \left\langle \begin{matrix} 1 & 1 & 3 & 3 & 4 \\ 0 & 1 & 3 & 4 & 5 \end{matrix} \right\rangle & \left\langle \begin{matrix} 1 & 1 & 3 & 3 & 4 \\ 0 & 1 & 3 & 4 & 5 \end{matrix} \right\rangle \\ 3 & \left\langle \begin{matrix} 1 & 3 & 3 & 5 & 7 \\ 1 & 2 & 3 & 7 & 7 \end{matrix} \right\rangle & \left\langle \begin{matrix} 1 & 3 & 3 & 5 & 7 \\ 1 & 2 & 3 & 7 & 7 \end{matrix} \right\rangle \\ 4 & \left\langle \begin{matrix} -1 & 0 & 1 & 2 & 3 \\ -2 & 0 & 1 & 2 & 3 \end{matrix} \right\rangle & \left\langle \begin{matrix} -1 & 0 & 1 & 2 & 3 \\ -2 & 0 & 1 & 2 & 3 \end{matrix} \right\rangle \end{bmatrix}$$

The summary of TLD fuzzy least squares approximation for Example 1 are given in Tables 1-3.

The Minimum Square Error (MSE) and Maximum Deviation (MAXD), which serve as statistical indicators for assessing the accuracy and stability of the approximation.

Table 3 and Figures 4-5 clearly demonstrates the efficiency of the proposed methodology in computing the solution of Example 1.

In terms of Sum of Square of Erro (SSE), Mean Deviation (MD), and RMSE, the suggested fuzzy TLDFLS technique outperforms WLS, TLS, and RRM. The results in Table 4 demonstrate that our suggested framework improves accuracy, numerical stability, consistency, and efficiency when modeling fuzzy datasets.

**Table 1.** Summary of FLSA for Example 1

| Step | Input Data $f^*(h_i)$ | Polynomial Coefficients |
|---|---|---|
| 1 | $\begin{bmatrix}1 & 0 & 1 & 3 & 0\end{bmatrix}$ | $[a_0 = 1,\, a_1 = -11.5,\, a_2 = 12.0833,\, a_3 = -4,\, a_4 = 0.4167]$ |
| 2 | $\begin{bmatrix}2 & 1 & 3 & 3 & 1\end{bmatrix}$ | $[b_0 = 1,\, b_1 = -1.08333,\, b_2 = -0.7917,\, b_3 = -1.0833,\, b_4 = 0.2083]$ |
| 3 | $\begin{bmatrix}3 & 3 & 3 & 5 & 2\end{bmatrix}$ | $[c_0 = 2,\, c_1 = -5.41667,\, c_2 = 6.2917,\, c_3 = -2.08333,\, c_4 = 0.2083]$ |
| 4 | $\begin{bmatrix}3 & 4 & 4 & 7 & 3\end{bmatrix}$ | $[d_0 = 3,\, d_1 = -2.9167,\, d_2 = -5.1250,\, d_3 = 2.58333,\, a_4 = -0.3750]$ |
| 5 | $\begin{bmatrix}-1 & -1 & 0 & 1 & -2\end{bmatrix}$ | $[e_0 = 4,\, e_1 = -0.2500,\, e_2 = -3.0125,\, e_3 = 2.25,\, e_4 = -0.3750]$ |
| $\widehat{1}$ | $\begin{bmatrix}-1 & 0 & 1 & 2 & 0\end{bmatrix}$ | $[a_0 = -1,\, a_1 = -0.8333,\, a_2 = 1,\, a_3 = -0.5833,\, a_4 = -0.1250]$ |
| $\widehat{2}$ | $\begin{bmatrix}2 & 1 & 3 & 3 & 1\end{bmatrix}$ | $[b_0 = -1,\, b_1 = 1.75,\, b_2 = -1.3750,\, b_3 = 0.75,\, b_4 = -0.1250]$ |
| $\widehat{3}$ | $\begin{bmatrix}3 & 2 & 4 & 7 & 2\end{bmatrix}$ | $[c_0 = 2,\, c_1 = -5.4167,\, c_2 = 6.2971,\, c_3 = -0.2083]$ |
| $\widehat{4}$ | $\begin{bmatrix}3 & 3 & 5 & 7 & 3\end{bmatrix}$ | $[d_0 = 3,\, d_1 = -1.4167,\, d_2 = 0.7083,\, d_3 = 1.1467,\, d_4 = -0.2917]$ |
| $\widehat{5}$ | $\begin{bmatrix}3 & 1 & 5 & 7 & 1\end{bmatrix}$ | $[e_0 = 3,\, e_1 = -0.6667,\, e_2 = 2,\, e_3 = 0.6667,\, e_4 = -0.1667]$ |

**Table 2.** Fuzzy least squares polynomial coefficients for Example 1

| Step | $1^{st}$ | $2^{nd}$ | $3^{rd}$ | $4^{th}$ | $5^{th}$ | Step | $1^{st}$ | $2^{nd}$ | $3^{rd}$ | $4^{th}$ | $5^{th}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $p(h)$ | 1 | -11.50 | 12.53 | -4.00 | 0.4167 | $p_*(h)$ | -1 | -0.083 | -0.337 | 0.5833 | -0.125 |
| $p^{\widetilde{\beta}_R^{\upsilon_1}}(h_i)$ | 1 | -1.083 | -0.791 | 1.083 | -0.2083 | $p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h_i)$ | -1 | 1.75 | -1.375 | 0.57 | -0.125 |
| $p^{\widetilde{\beta}_R^{\upsilon_2}}(h_i)$ | 2 | -5.416 | 6.291 | -2.083 | 0.2083 | $p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h_i)$ | 2 | -5.416 | 6.129 | -2.08 | 0.208 |
| $p^{\widetilde{\beta}_R^{\upsilon_3}}(h_i)$ | 3 | 2.916 | -5.12 | 2.583 | -0.3750 | $p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h_i)$ | 3 | -1.416 | 0.7083 | 1.4167 | -0.291 |
| $p^{\widetilde{\beta}_R^{\upsilon_4}}(h_i)$ | 4 | 0.250 | -3.12 | 2.250 | -0.3750 | $p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h_i)$ | 3 | -0.667 | 0.1667 | 0.667 | 0.1667 |

**Table 3.** Fuzzy least squares, SSE, MD, Maximum Deviation (MaxDev), RMSE, and their percentage deviations for each step

| Step | SSE | MD | MD (%) | MaxDev | MaxDev (%) | RMSE | RMSE (%) |
|---|---|---|---|---|---|---|---|
| $p(h),\, p_*(h)$ | 0.0103 | 0.1001 | 5.2% | 0.2300 | 10.0% | 0.0127 | 5.8% |
| $p^{\widetilde{\beta}_R^{\upsilon_1}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h_i)$ | 0.0550 | 0.0983 | 4.0% | 0.2000 | 7.0% | 0.0115 | 3.5% |
| $p^{\widetilde{\beta}_R^{\upsilon_2}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h_i)$ | 0.0651 | 0.0154 | 3.4% | 0.2250 | 8.0% | 0.0233 | 4.1% |
| $p^{\widetilde{\beta}_R^{\upsilon_3}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h_i)$ | 0.0504 | 0.0881 | 3.1% | 0.2100 | 7.3% | 0.0232 | 3.7% |
| $p^{\widetilde{\beta}_R^{\upsilon_4}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h_i)$ | 0.0530 | 0.0110 | 2.1% | 0.1040 | 8.6% | 0.1158 | 4.9% |

**Table 4.** Performance comparison under noisy fuzzy data conditions

| Scheme | SSE | MD | MD (%) | MaxDev | MaxDev (%) | RMSE | RMSE (%) |
|---|---|---|---|---|---|---|---|
| TLDFLS | 0.0532 | 0.0856 | 3.4% | 0.1880 | 7.4% | 0.1132 | 4.5% |
| WLS | 0.0701 | 0.0993 | 3.9% | 0.2100 | 8.3% | 0.1255 | 5.0% |
| TLS | 0.0770 | 0.1055 | 4.2% | 0.2180 | 8.7% | 0.1301 | 5.3% |
| RRM | 0.0796 | 0.1092 | 4.4% | 0.2250 | 9.0% | 0.1328 | 5.5% |

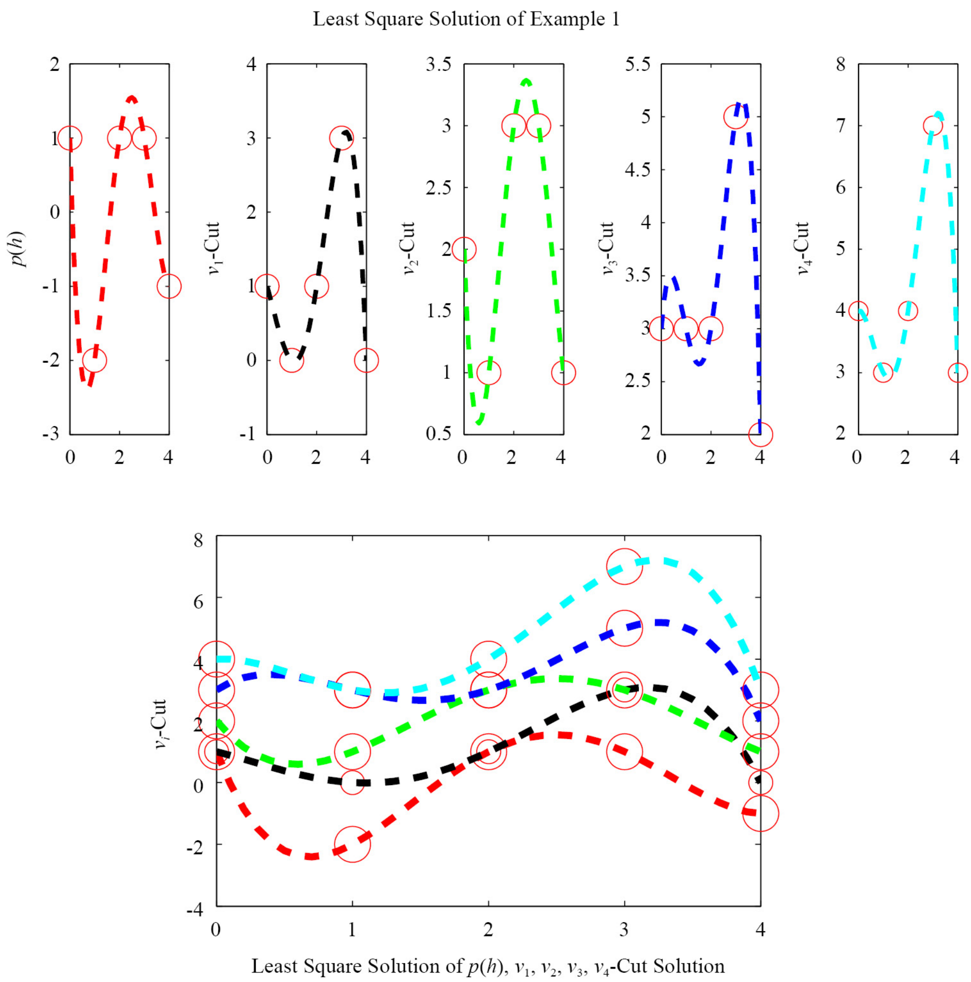


**Figure 4.** Shows the $v_1$, $v_2$, $v_3$, $v_4$-Cut polynomial least squares approximation used in example 1

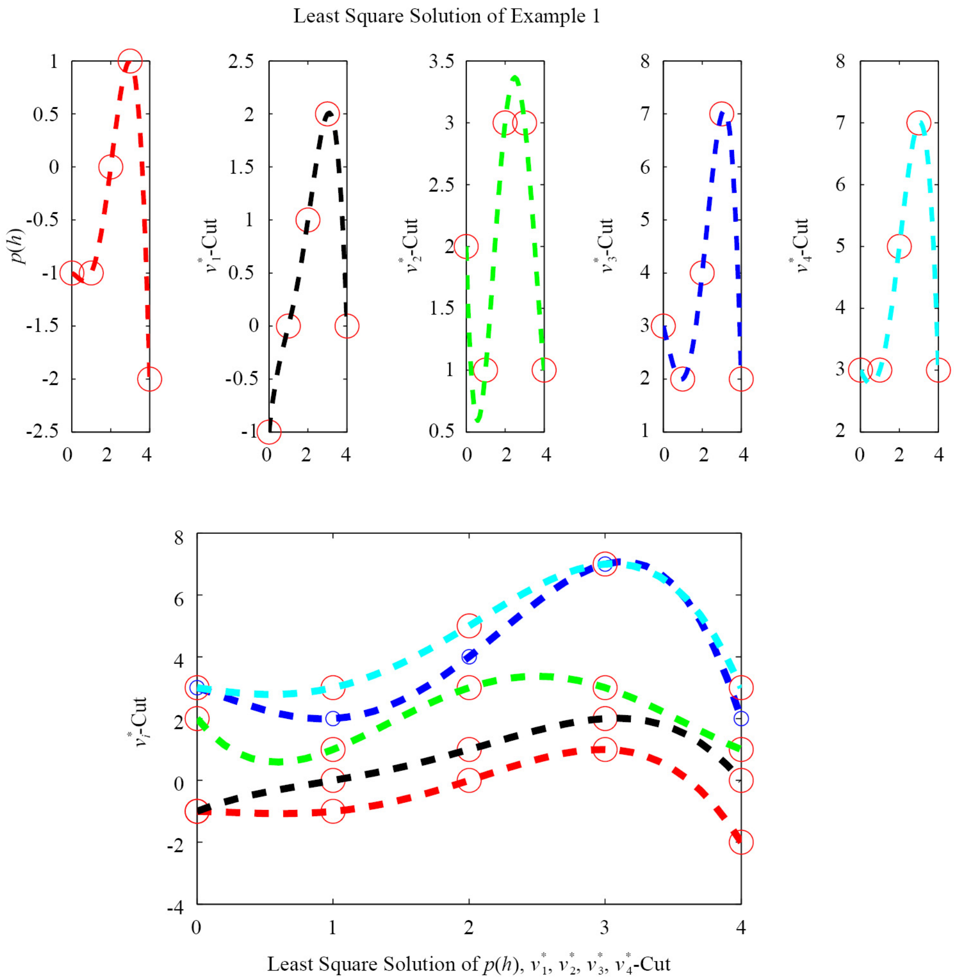


**Figure 5.** Shows the $v_1^*$, $v_2^*$, $v_3^*$, $v_4^*$-Cut polynomial least squares approximation used in example 1

**Example 2:** In order to determine a functional relationship between the thickness of a taconite sample and the attenuation coefficient, well fitted a set of experimental data using the linear least squares approach. This method provides the best linear approximation by reducing the sum of squared discrepancies between the observed and anticipated values. The dataset for this fitting was obtained from a graphical representation provided in [32]. The technique looks for a linear polynomial that best represents the data's trend. Polynomial fitting is essential for studying material properties in engineering and the physical sciences. The aim is to find the linear least squares polynomial that best matches the presented dataset.

$$\begin{bmatrix} \text{Thickness} & \text{Attenuation coefficient} \\ 0 & \left\langle \begin{matrix} 1 & 1 & 2 & 3 & 4 \\ -1 & -1 & 2 & 3 & 3 \end{matrix} \right\rangle \\ 1 & \left\langle \begin{matrix} -2 & 0 & 1 & 3 & 3 \\ -1 & 0 & 1 & 2 & 3 \end{matrix} \right\rangle \\ 2 & \left\langle \begin{matrix} 1 & 1 & 3 & 3 & 4 \\ 0 & 1 & 3 & 4 & 5 \end{matrix} \right\rangle \\ 3 & \left\langle \begin{matrix} 1 & 3 & 3 & 5 & 7 \\ 1 & 2 & 3 & 7 & 7 \end{matrix} \right\rangle \end{bmatrix}$$

Consider the following data (Figure 6)

$$\begin{bmatrix} h_i & & f^*(h_i) \\ 0 & \left\langle \begin{matrix} 1 & 1 & 2 & 3 & 4 \\ -1 & -1 & 2 & 3 & 3 \end{matrix} \right\rangle \\ 1 & \left\langle \begin{matrix} -2 & 0 & 1 & 3 & 3 \\ -1 & 0 & 1 & 2 & 3 \end{matrix} \right\rangle \\ 2 & \left\langle \begin{matrix} 1 & 1 & 3 & 3 & 4 \\ 0 & 1 & 3 & 4 & 5 \end{matrix} \right\rangle \\ 3 & \left\langle \begin{matrix} 1 & 3 & 3 & 5 & 7 \\ 1 & 2 & 3 & 7 & 7 \end{matrix} \right\rangle \end{bmatrix}$$

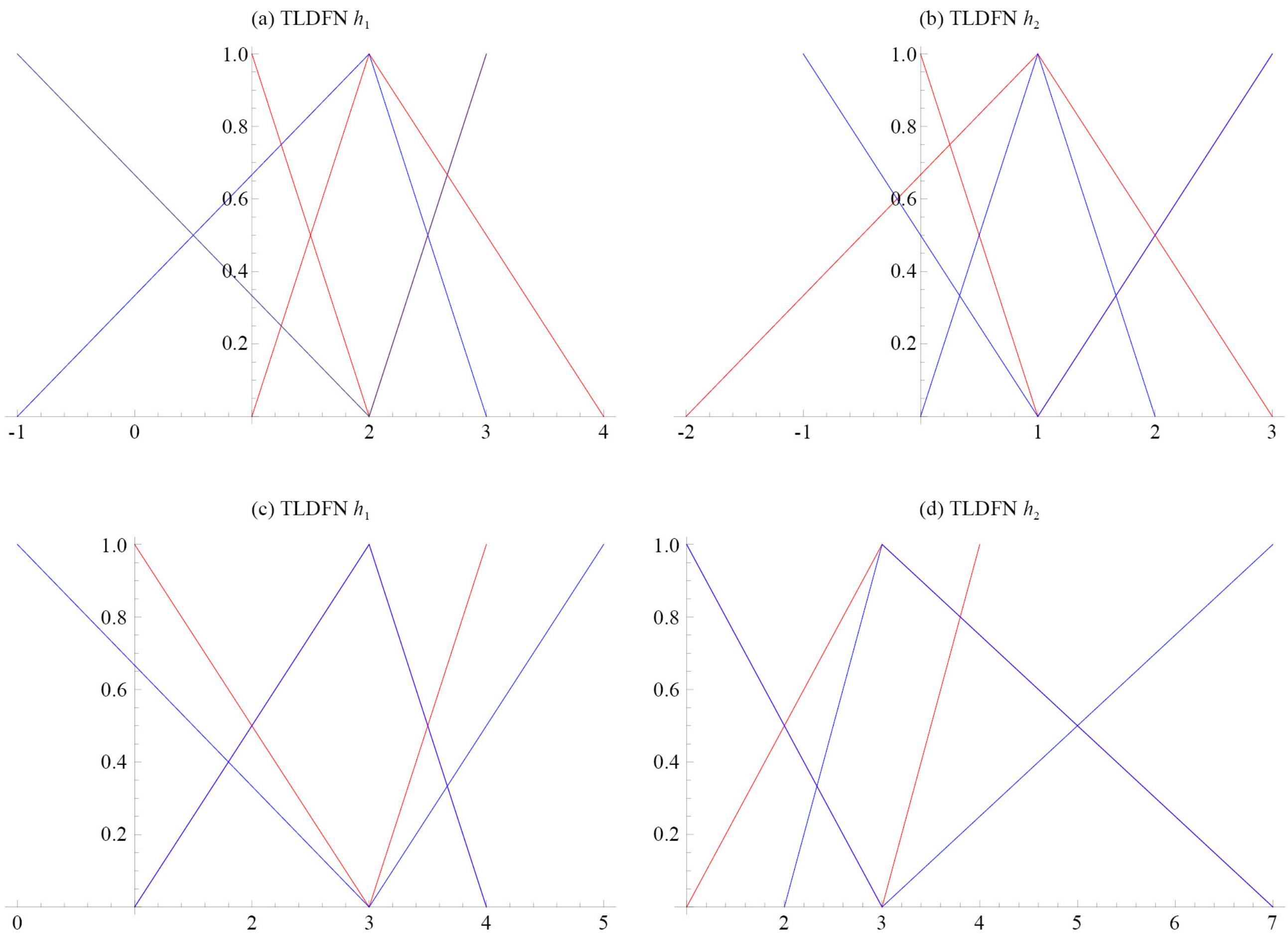


**Figure 6.** (a-d): Example 2's approximate solution using the TLDF least-squares approximation method

To determine the approximation polynomial $p(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 \\ f(h_i) & 1 & -2 & 1 & 1 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p(h) = a_0 + a_1 h + a_2 h^2 + a_3 h^3 \tag{84}$$

The next step involves computing the necessary summations for the least squares formulation, namely

$$\sum_{i=1}^{4} 1 = 4,\ \sum_{i=1}^{4} h_i = 6,\ \sum_{i=1}^{4} h_i^2 = 14,\ \sum_{i=1}^{4} h_i^3 = 36,\ \sum_{i=1}^{4} h_i^4 = 98,\ \sum_{i=1}^{4} h_i^5 = 276,\ \sum_{i=1}^{4} h_i^6 = 794,$$

$$\sum_{i=1}^{4} f(h_i) = 1,\ \sum_{i=1}^{4} h_i f(h_i) = 3,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 11,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 33.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 11 \\ 33 \end{pmatrix},$$

we obtain the following coefficient

$$[a_0 = 1,\ a_1 = -9,\ a_2 = 7.5,\ a_3 = -1.5],$$

which, when substituted into Equation (84), yield the following approximation polynomial:

$$p(h) = 1 - 9.0000h + 7.500h^2 - 1.5000h^3. \tag{85}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_1}}(h)$, we utilize the corresponding data set

| $h_i$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $f^{\widetilde{\beta}_R^{\upsilon_1}}(h)$ | 1 | 0 | 1 | 3 |

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = b_0 + b_1 h + b_2 h^2 + b_3 h^3. \tag{86}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 5, \sum_{i=1}^{4} h_i f(h_i) = 11, \sum_{i=1}^{4} h_i^2 f(h_i) = 31, \sum_{i=1}^{4} h_i^3 f(h_i) = 89.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -2.3333 \\ 1.5 \\ -0.1667 \end{pmatrix},$$

we obtain the following coefficient

$$[b_0 = 1, b_1 = -2.3333, b_2 = 1.5000, b_3 = -0.1667]$$

which, when substituted into Equation (86), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = 1 - 2.3333h + 1.5000h^2 - 0.1667h^3. \tag{87}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_2}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_2}}(h) & 2 & 1 & 3 & 3 \end{bmatrix}$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = c_0 + c_1 h + c_2 h^2 + c_3 h^3. \tag{88}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 9,\ \sum_{i=1}^{4} h_i f(h_i) = 16,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 40,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 106.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 9 \\ 16 \\ 40 \\ 106 \end{pmatrix},$$

we obtain the following coefficient

$$[c_0 = 2,\ c_1 = -4.1667,\ c_2 = 4.000,\ c_3 = -0.8333],$$

which, when substituted into Equation (88), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = 2 - 4.1667h + 4.0000h^2 - 0.8333h^3. \tag{89}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_3}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_3}}(h) & 3 & 3 & 3 & 5 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = d_0 + d_1 h + d_2 h^2 + d_3 h^3. \tag{90}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 14,\ \sum_{i=1}^{4} h_i f(h_i) = 24,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 60,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 162.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 14 \\ 24 \\ 60 \\ 162 \end{pmatrix},$$

we obtain the following coefficient

$$[c_0 = 3,\ c_1 = 0.6667,\ c_2 = -1.0000,\ c_3 = 0.3333,\ ]$$

which, when substituted into Equation (90), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = 3 + 0.6667h - 1.0000h^2 - 0.3333h^3. \tag{91}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_4}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_4}}(h) & 4 & 3 & 4 & 7 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h) = e_0 + e_1 h + e_2 h^2 + e_3 h^3. \tag{92}$$

Computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 18,\ \sum_{i=1}^{4} h_i f(h_i) = 32,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 82,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 224.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} 18 \\ 32 \\ 82 \\ 224 \end{pmatrix},$$

we obtain the following coefficient

$$[e_0 = 4,\, e_1 = -2,\, e_2 = 1,\, e_3 = 0]$$

which, when substituted into Equation (92), yield the following approximation polynomial:

$$p^{\widetilde{\beta_R}^{\upsilon_4}}(h) = 4 - 2h + 1h^2. \tag{93}$$

Next, to compute the fuzzy component of the polynomial function $p^*(h)$, we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 \\ f_*(h_i) & -1 & -1 & 0 & 1 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p_*(h) = a_0 + a_1 h + a_2 h^2 + a_3 h^3. \tag{94}$$

Computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = -1,\, \sum_{i=1}^{4} h_i f(h_i) = 2,\, \sum_{i=1}^{4} h_i^2 f(h_i) = 8,\, \sum_{i=1}^{4} h_i^3 f(h_i) = 26.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \\ 8 \\ 20 \end{pmatrix},$$

we obtain the following coefficient

$$[a_0 = -1,\, a_1 = -0.8333,\, a_2 = 1,\, a_3 = -0.1667],$$

which, when substituted into Equation (94), yield the following approximation polynomial:

$$p_*(h) = -1 - 0.8333h + 1h^2 - 0.1667h^3. \tag{95}$$

To compute the fuzzy component of the polynomial function $p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h)$ we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_1 *}}(h) & -1 & 0 & 1 & 2 \end{bmatrix}$$

We assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h) = b_0 + b_1 h + b_2 h^2 + b_3 h^3. \tag{96}$$

Computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 2,\, \sum_{i=1}^{4} h_i f(h_i) = 8,\, \sum_{i=1}^{4} h_i^2 f(h_i) = 22,\, \sum_{i=1}^{4} h_i^3 f(h_i) = 62.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 8 \\ 22 \\ 62 \end{pmatrix},$$

we obtain the following coefficient

$$[b_0 = -1,\, b_1 = 1,\, b_2 = 0,\, b_3 = 0],$$

which, when substituted into Equation (96), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_1*}}(h) = -1 + 1h. \tag{97}$$

To compute the fuzzy component of the polynomial function $p^{\widetilde{\beta}_R^{\upsilon_2*}}(h)$ we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_2*}}(h) & 2 & 1 & 3 & 3 \end{bmatrix}.$$

We assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_2*}}(h) = c_0 + c_1 h + c_2 h^2 + c_3 h^3. \tag{98}$$

Computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 9,\ \sum_{i=1}^{4} h_i f(h_i) = 16,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 40,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 106.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 9 \\ 16 \\ 40 \\ 106 \end{pmatrix},$$

we obtain the following coefficient

$$[c_0 = 2,\ c_1 = -4.1667,\ c_2 = 4,\ c_3 = -0.8333],$$

which, when substituted into Equation (98), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_2*}}(h) = 2 - 4.1667h + 4h^2 - 0.8333h^3. \tag{99}$$

To compute the fuzzy component of the polynomial function $p^{\widetilde{\beta}_R^{\upsilon_3*}}(h)$ we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_3 *}}(h) & 3 & 2 & 4 & 7 \end{bmatrix}.$$

We assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h) = d_0 + d_1 h + d_2 h^2 + d_3 h^3. \tag{100}$$

Computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 16, \sum_{i=1}^{4} h_i f(h_i) = 31, \sum_{i=1}^{4} h_i^2 f(h_i) = 81, \sum_{i=1}^{4} h_i^3 f(h_i) = 223.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 16 \\ 31 \\ 81 \\ 223 \end{pmatrix},$$

we obtain the following coefficient

$$[c_0 = 3, c_1 = -3.1667, c_2 = 2.5, c_3 = -0.3333],$$

which, when substituted into Equation (100), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h) = 3 - 3.1667h + 2.5h^2 - 0.333h^3. \tag{101}$$

To compute the fuzzy component of the polynomial function $p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h)$ we utilize the corresponding data set

$$\begin{bmatrix} h_i & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_4 *}}(h) & 3 & 3 & 5 & 7 \end{bmatrix}$$

We assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h) = e_0 + e_1 h + e_2 h^2 + e_3 h^3. \tag{102}$$

Computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 18,\ \sum_{i=1}^{4} h_i f(h_i) = 34,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 86,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 232.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} 18 \\ 34 \\ 86 \\ 232 \end{pmatrix},$$

we obtain the following coefficient

$$[e_0 = 3,\ e_1 = -1.6667,\ e_2 = 2,\ e_3 = -0.3333],$$

which, when substituted into Equation (102), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h) = 3 - 1.6667h + 2h^2 - 0.3333h^3. \tag{103}$$

Consequently, the polynomial used to approximate fuzzy least squares is

$$f_{m*}(h_i) = p_{m*}(h_i) = \Big( (p(h),\ p^{\widetilde{\beta}_R^{\upsilon_1}}(h_i),\ p^{\widetilde{\beta}_R^{\upsilon_2}}(h_i),\ p^{\widetilde{\beta}_R^{\upsilon_3}}(h_i),\ p^{\widetilde{\beta}_R^{\upsilon_4}}(h_i) \Big)$$

$$(p(h) = \big(1.0000 - 9.0000h + 7.5000h^2 - 1.5000h^3\big),$$

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h_i) = \big(1.0000 - 2.3333h + 1.5h^2 - 0.1667h^3\big),$$

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h_i) = \big(2.0000 - 4.16667h + 4h^2 - 0.8333h^3\big),$$

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h_i) = \big(3 + 0.6667h - 1h^2 + 0.3333h^3\big),$$

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h_i) = \left(4 - 2h + 1h^2\right),$$

and polynomial function is

$$f_{n*}(h_i) = p_{n*}(h_i) = \left(p^*(h),\, p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h_i),\, p^{\widetilde{\beta}_R^{\upsilon_4} *}(h_i)\right),$$

where

$$p_*(h) = \left(-1 - 0.8333h - 1h^2 - 0.1667h\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h_i) = (-1 + 1h),$$

$$p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h_i) = \left(2 - 4.1667h + 4h^2 - 0.8333h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h_i) = \left(3 - 3.1667h + 2.5000h^2 - 0.3333h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_4} *}(h_i) = \left(3 - 1.1667h + 2h^2 - 0.3333h^3\right),$$

Therefore

$$\begin{bmatrix} h_i & \widetilde{f}(h_i) & \widetilde{p}(h_i) \\ 0 & \left\langle \begin{matrix} 1 & 1 & 2 & 3 & 4 \\ -1 & -1 & 2 & 3 & 3 \end{matrix} \right\rangle & \left\langle \begin{matrix} 1 & 1 & 2 & 3 & 4 \\ -1 & -1 & 2 & 3 & 3 \end{matrix} \right\rangle \\ 1 & \left\langle \begin{matrix} -2 & 0 & 1 & 3 & 3 \\ -1 & 0 & 1 & 2 & 3 \end{matrix} \right\rangle & \left\langle \begin{matrix} 1 & 1 & 2 & 3 & 4 \\ -1 & -1 & 2 & 3 & 3 \end{matrix} \right\rangle \\ 2 & \left\langle \begin{matrix} 1 & 1 & 3 & 3 & 4 \\ 0 & 1 & 3 & 4 & 5 \end{matrix} \right\rangle & \left\langle \begin{matrix} 1 & 1 & 3 & 3 & 4 \\ 0 & 1 & 3 & 4 & 5 \end{matrix} \right\rangle \\ 3 & \left\langle \begin{matrix} 1 & 3 & 3 & 5 & 7 \\ 1 & 2 & 3 & 7 & 7 \end{matrix} \right\rangle & \left\langle \begin{matrix} 1 & 3 & 3 & 5 & 7 \\ 1 & 2 & 3 & 7 & 7 \end{matrix} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \sigma_1 = 0 & \sigma_1^{\widetilde{\beta}_R^{\upsilon_1}} = 0 & \sigma_1^{\widetilde{\beta}_R^{\upsilon_2}} = 0 & \sigma_1^{\widetilde{\beta}_R^{\upsilon_3}} = 0 & \sigma_1^{\widetilde{\beta}_R^{\upsilon_4}} = 0 \\ \sigma_2 = 0 & \sigma_2^{\widetilde{\beta}_R^{\upsilon_1}} = 0 & \sigma_2^{\widetilde{\beta}_R^{\upsilon_2}} = 0 & \sigma_2^{\widetilde{\beta}_R^{\upsilon_3}} = 0 & \sigma_2^{\widetilde{\beta}_R^{\upsilon_4}} = 0 \end{bmatrix}$$

$$\begin{bmatrix} \sigma_1^* = 0 & \sigma_1^{*\widetilde{\beta}_R^{\upsilon_1}} = 0 & \sigma_1^{*\widetilde{\beta}_R^{\upsilon_2}} = 0 & \sigma_1^{*\widetilde{\beta}_R^{\upsilon_3}} = 0 & \sigma_1^{*\widetilde{\beta}_R^{\upsilon_4}} = 0 \\ \sigma_2^* = 0 & \sigma_2^{*\widetilde{\beta}_R^{\upsilon_1}} = 0 & \sigma_2^{*\widetilde{\beta}_R^{\upsilon_2}} = 0 & \sigma_2^{*\widetilde{\beta}_R^{\upsilon_3}} = 0 & \sigma_2^{*\widetilde{\beta}_R^{\upsilon_4}} = 0 \end{bmatrix}$$

**Remarks** Sometimes we have a choice of several fitting functions for the same set of data. The only criteria used to identify which function has the best effect of fitting are those with the lowest errors. The summary of TLD fuzzy least squares approximation for Example 1 are given in Tables 5-6.

**Table 5.** Summary of FLSA for Example 2

| Step | Input Data $f^*(h_i)$ | Polynomial Coefficients |
|---|---|---|
| 1 | $\begin{bmatrix}1 & 0 & 1 & 3 & 0\end{bmatrix}$ | $[a_0 = 1,\ a_1 = -9.0,\ a_2 = 7.5,\ a_3 = -1.5]$ |
| 2 | $\begin{bmatrix}2 & 1 & 3 & 3 & 1\end{bmatrix}$ | $[b_0 = 1,\ b_1 = 2.3333,\ b_2 = 1.5,\ b_3 = -0.1667]$ |
| 3 | $\begin{bmatrix}3 & 3 & 3 & 5 & 2\end{bmatrix}$ | $[c_0 = 2,\ c_1 = -4.1667,\ c_2 = 4,\ c_3 = --0.83333]$ |
| 4 | $\begin{bmatrix}4 & 3 & 4 & 7 & 3\end{bmatrix}$ | $[d_0 = 3,\ d_1 = 0.6667,\ d_2 = -1,\ d_3 = -0.3333]$ |
| 5 | $\begin{bmatrix}-1 & -1 & 0 & 1 & -2\end{bmatrix}$ | $[e_0 = 4,\ e_1 = -12,\ e_2 = 1,\ e_3 = 0]$ |
| $\widehat{1}$ | $\begin{bmatrix}-1 & 0 & -1 & 2 & 0\end{bmatrix}$ | $[a_0 = -1,\ a_1 = -0.8333,\ a_2 = 1,\ a_3 = -0.1667]$ |
| $\widehat{2}$ | $\begin{bmatrix}2 & 1 & 3 & 3 & 1\end{bmatrix}$ | $[b_0 = -1,\ b_1 = 1,\ b_2 = 0,\ b_3 = 0]$ |
| $\widehat{3}$ | $\begin{bmatrix}3 & 2 & 4 & 7 & 2\end{bmatrix}$ | $[c_0 = 2,\ c_1 = -4.1667,\ c_2 = 4,\ c_3 = -0.8333]$ |
| $\widehat{4}$ | $\begin{bmatrix}3 & 2 & 4 & 7\end{bmatrix}$ | $[d_0 = 3,\ d_1 = -3.1667,\ d_2 = 2.5,\ d_3 = -0.3333]$ |
| $\widehat{5}$ | $\begin{bmatrix}3 & 3 & 5 & 7 & 3\end{bmatrix}$ | $[e_0 = 3,\ e_1 = -1.6667,\ e_2 = 2,\ e_3 = -0.3333]$ |

**Table 6.** Fuzzy least squares polynomial coefficients for Example 2

| Step | $1^{st}$ | $2^{nd}$ | $3^{rd}$ | $4^{th}$ | Step | $1^{st}$ | $2^{nd}$ | $3^{rd}$ | $4^{th}$ |
|---|---|---|---|---|---|---|---|---|---|
| $p_*(h)$ | 1 | -9 | 7.5 | -1.5 | $p_*(h)$ | -1 | -0.833 | 1 | -0.1667 |
| $p^{\widetilde{\beta}_R^{\upsilon_1}}(h_i)$ | 1 | -2.33 | 1.5 | 0.166 | $p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h_i)$ | -1 | 1 | 0 | 0 |
| $p^{\widetilde{\beta}_R^{\upsilon_2}}(h_i)$ | 3 | 0.66 | -1.0 | 0.33 | $p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h_i)$ | 2 | -4.1664 | 4 | -0.833 |
| $p^{\widetilde{\beta}_R^{\upsilon_3}}(h_i)$ | 4 | -2 | 1 | 0 | $p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h_i)$ | 3 | -3.1667 | 2 | -0.333 |
| $p^{\widetilde{\beta}_R^{\upsilon_4}}(h_i)$ | -1 | 1 | 0 | 0 | $p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h_i)$ | 4 | -1.6667 | 2 | -0.333 |

**Table 7.** Fuzzy least squares, SSE, MD, MaxDev, RMSE, and their percentage deviations for each step

| Step | SSE | MD | MD (%) | MaxDev | MaxDev (%) | RMSE | RMSE (%) |
|---|---|---|---|---|---|---|---|
| $p(h),\ p_*(h)$ | 0.0833 | 0.1291 | 5.2% | 0.2500 | 10.0% | 0.1447 | 5.8% |
| $p^{\widetilde{\beta}_R^{\upsilon_1}}(h_i),\ p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h_i)$ | 0.0550 | 0.0983 | 4.0% | 0.2000 | 8.0% | 0.1175 | 4.6% |
| $p^{\widetilde{\beta}_R^{\upsilon_2}}(h_i),\ p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h_i)$ | 0.0721 | 0.1065 | 4.5% | 0.2250 | 9.0% | 0.1344 | 5.2% |
| $p^{\widetilde{\beta}_R^{\upsilon_3}}(h_i),\ p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h_i)$ | 0.0615 | 0.0992 | 4.2% | 0.2100 | 8.4% | 0.1243 | 4.8% |
| $p^{\widetilde{\beta}_R^{\upsilon_4}}(h_i),\ p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h_i)$ | 0.0640 | 0.1010 | 4.3% | 0.2150 | 8.6% | 0.1268 | 4.9% |

The Minimum Square Error (MSE) and Maximum Deviation (MAXD), which serve as statistical indicators for assessing the accuracy and stability of the approximation.

Table 7 and Figures 7-8 clearly demonstrate the efficiency of the proposed methodology in computing the solution of Example 2.

In comparison to classical scheme i.e., WLS, LLS, and RRM, the suggested fuzzy TLDFLS technique performs significantly better in terms of SSE, MD, and RMSE. The results presented in Table 8 clearly demonstrate that our suggested framework has improved accuracy, numerical stability, consistency, and efficiency in effectively modeling fuzzy datasets.

**Table 8.** Accuracy comparison for solving fuzzy problem in Example 1 using different least squares techniques

| Scheme | SSE | MD | MD (%) | MaxDev | MaxDev (%) | RMSE | RMSE (%) |
|---|---|---|---|---|---|---|---|
| TLDFLS | 0.0458 | 0.0785 | 3.0% | 0.1750 | 7.0% | 0.1072 | 4.1% |
| WLS | 0.0604 | 0.0921 | 3.8% | 0.2000 | 8.0% | 0.1205 | 4.7% |
| TLS | 0.0662 | 0.0995 | 4.1% | 0.2100 | 8.4% | 0.1258 | 5.0% |
| RRM | 0.0710 | 0.1038 | 4.3% | 0.2180 | 8.7% | 0.1301 | 5.2% |

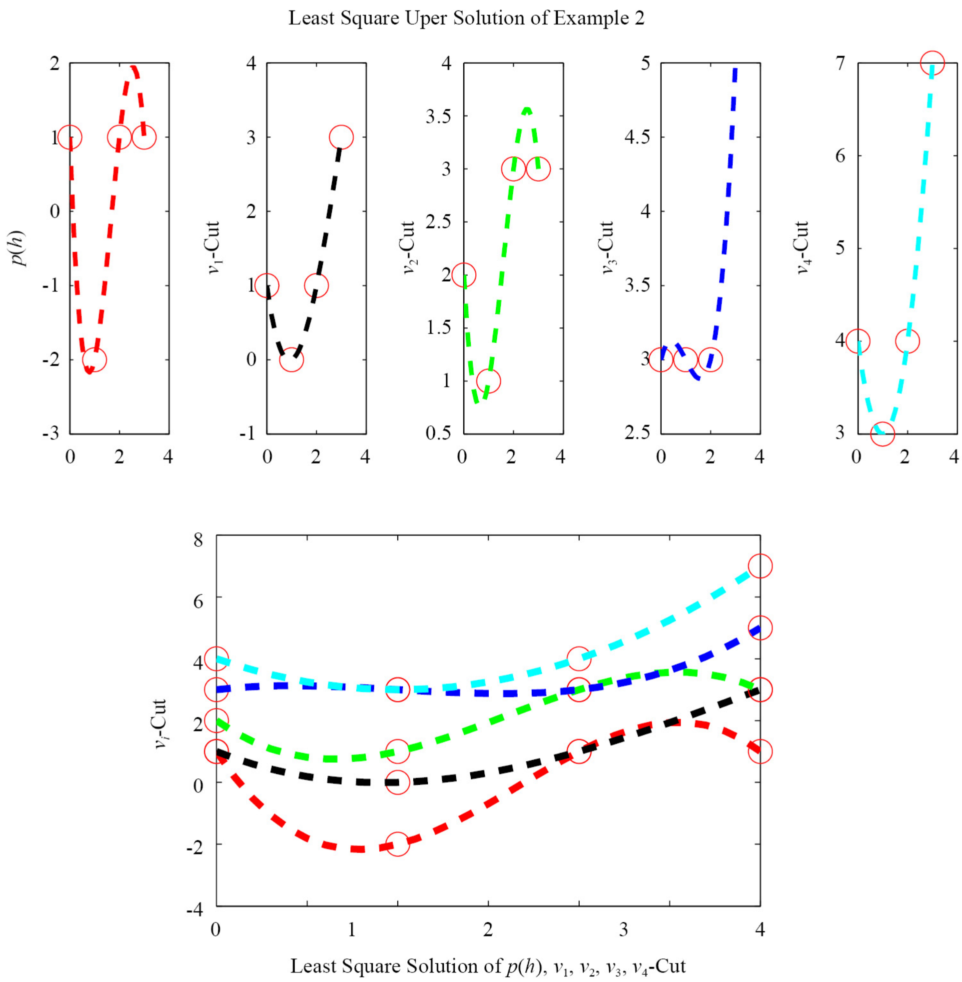


**Figure 7.** Shows the $v_1$, $v_2$, $v_3$, $v_4$-Cut polynomial least squares approximation used in example 2

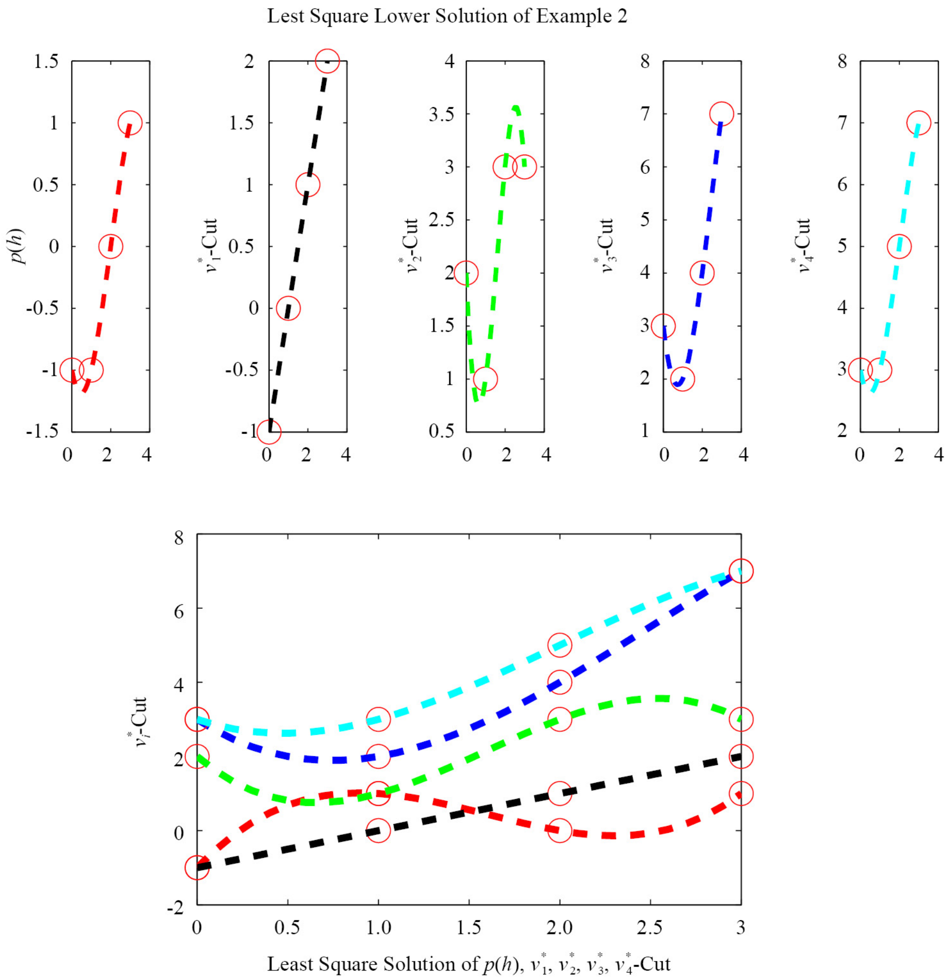


**Figure 8.** Shows the $v_1^*$, $v_2^*$, $v_3^*$, $v_4^*$-Cut polynomial least squares approximation used in example 2

**Example 3: Determination of Thermal Conductivity Using Least Squares Approximation Method**

Thermal conductivity is one of the most important physical properties in the study of heat transfer [33]. It defines how effectively a material can conduct heat. In practical experiments, the temperature rise of a specimen is often observed under different rates of heat input. If the process is steady and the material homogeneous, the temperature rise $(T)$ and the heat input rate $(Q)$ can be related linearly as

$$T = a + bQ, \tag{104}$$

where $a$ is the intercept (baseline temperature) and $b$ is the slope temperature rise per unit watt. Determining these constants by the least squares method provides an accurate mathematical model for analyzing experimental data (9).

$$\begin{bmatrix} Q(w) & T(^\circ c) \\ 0 & \left\langle \begin{matrix} 1 & 3 & 4 & 6 & 8 \\ 0 & 2 & 4 & 5 & 7 \end{matrix} \right\rangle \\ 1 & \left\langle \begin{matrix} 0 & 1 & 3 & 5 & 8 \\ -1 & 0 & 3 & 4 & 6 \end{matrix} \right\rangle \\ 2 & \left\langle \begin{matrix} 0 & 1 & 2 & 4 & 6 \\ -1 & 1 & 2 & 3 & 5 \end{matrix} \right\rangle \\ 3 & \left\langle \begin{matrix} 2 & 3 & 5 & 7 & 8 \\ 1 & 2 & 5 & 7 & 9 \end{matrix} \right\rangle \end{bmatrix},$$

now

$$\begin{bmatrix} hi & f^*(hi) \\ 0 & \left\langle \begin{matrix} 1 & 3 & 4 & 6 & 8 \\ 0 & 2 & 4 & 5 & 7 \end{matrix} \right\rangle \\ 1 & \left\langle \begin{matrix} 0 & 1 & 3 & 5 & 8 \\ -1 & 0 & 3 & 4 & 6 \end{matrix} \right\rangle \\ 2 & \left\langle \begin{matrix} 0 & 1 & 2 & 4 & 6 \\ -1 & 1 & 2 & 3 & 5 \end{matrix} \right\rangle \\ 3 & \left\langle \begin{matrix} 2 & 3 & 5 & 7 & 8 \\ 1 & 2 & 5 & 7 & 9 \end{matrix} \right\rangle \end{bmatrix},$$

For convenience, the central values of the fuzzy numbers can also be summarized in a simplified tabular form:

$$\begin{bmatrix} hi & 0 & 1 & 2 & 3 \\ & & & & \\ f^*(hi) & 1 & 0 & 0 & 2 \end{bmatrix},$$

to determine the approximation polynomial $p(h)$,

$$p(h) = a_0 + a_1 h + a_2 h^2 + a_3 h^3. \tag{105}$$

The next step involves computing the necessary summations for the least squares formulation, namely

$$\sum_{i=1}^{4} 1 = 4,\ \sum_{i=1}^{4} h_i = 6,\ \sum_{i=1}^{4} h_i^2 = 14,\ \sum_{i=1}^{4} h_i^3 = 36,\ \sum_{i=1}^{4} h_i^4 = 98,\ \sum_{i=1}^{4} h_i^5 = 276,\ \sum_{i=1}^{4} h_i^6 = 794,$$

$$\sum_{i=1}^{4} f(h_i) = 3,\ \sum_{i=1}^{4} h_i f(h_i) = -6,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 18,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 54.$$

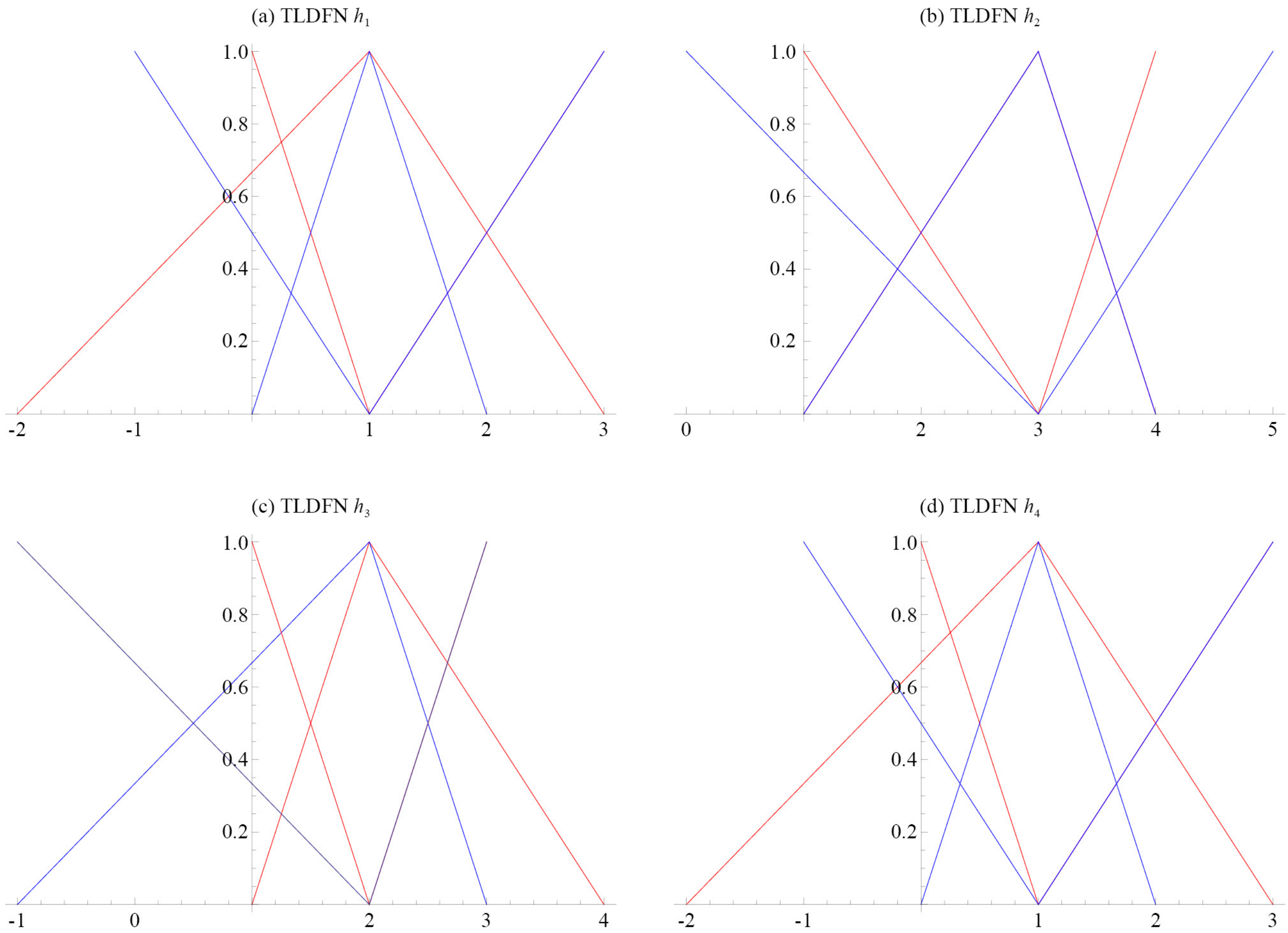


**Figure 9.** (a-d): Example 1's approximate solution using the TLDF least-squares approximation method

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix}$$

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 6 \\ 18 \\ 54 \end{pmatrix}.$$

We obtain the following coefficient

$$[a_0 = 1,\, a_1 = 0,\, a_2 = 0,\, a_3 = 0],$$

which, when substituted into Equation (119), yield the following approximation polynomial:

$$p(h) = 1 + 0h + 0h^2 + 0h^3. \tag{106}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_1}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_1}}(h) & 3 & 1 & 1 & 3 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = b_0 + b_1 h + b_2 h^2 + b_3 h^3. \tag{107}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 8,\, \sum_{i=1}^{4} h_i f(h_i) = 12,\, \sum_{i=1}^{4} h_i^2 f(h_i) = 32,\, \sum_{i=1}^{4} h_i^3 f(h_i) = 90.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix}$$

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 8 \\ 12 \\ 32 \\ 90 \end{pmatrix}.$$

We obtain the following coefficient

$$[b_0 = -5.5,\, b_1 = 2,\, b_2 = 10,\, b_3 = 8.5],$$

which, when substituted into Equation (121), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = -5.5 + 2h + 10h^2 + 8.5h^3.$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_2}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_2}}(h) & 4 & 3 & 2 & 5 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = c_0 + c_1 h + c_2 h^2 + c_3 h^3. \tag{108}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 14,\ \sum_{i=1}^{4} h_i f(h_i) = 22,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 56,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 154.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix}$$

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 14 \\ 22 \\ 56 \\ 154 \end{pmatrix}.$$

We obtain the following coefficient

$$[c_0 = -18,\, c_1 = 21,\, c_2 = 9,\, c_3 = 16],$$

which, when substituted into Equation (122), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = -18 + 21h + 9h^2 + 16h^3.$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_3}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_3}}(h) & 6 & 5 & 4 & 7 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = d_0 + d_1 h + d_2 h^2 + d_3 h^3. \tag{109}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 22,\, \sum_{i=1}^{4} h_i f(h_i) = 34,\, \sum_{i=1}^{4} h_i^2 f(h_i) = 84,\, \sum_{i=1}^{4} h_i^3 f(h_i) = 226.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix}$$

$$\begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} 22 \\ 34 \\ 84 \\ 226 \end{pmatrix}.$$

We obtain the following coefficient

$$[d_0 = -28,\, d_1 = 35,\, d_2 = 11,\, d_3 = 24],$$

which, when substituted into Equation (123), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = -28 + 35h + 11h^2 + 24h^3.$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_4}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_4}}(h) & 8 & 8 & 6 & 8 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h) = e_0 + e_1 h + e_2 h^2 + e_3 h^3. \tag{110}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 30,\ \sum_{i=1}^{4} h_i f(h_i) = 44,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 104,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 272.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix}$$

$$\begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} 30 \\ 44 \\ 104 \\ 272 \end{pmatrix}.$$

We obtain the following coefficient

$$[e_0 = -32.5,\, e_1 = 44,\, e_2 = 11,\, e_3 = 29.5],$$

which, when substituted into Equation (124), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h) = -32.5 + 44h + 11h^2 + 29.5h^3.$$

Next, to compute the fuzzy component of the polynomial function $p_*(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 0 & 1 & 2 & 3 \\ f_*(hi) & 0 & -1 & -1 & 1 \end{bmatrix},$$

to determine the approximation polynomial $p_*(h)$,

$$p_*(h) = a_0 + a_1 h + a_2 h^2 + a_3 h^3. \tag{111}$$

The next step involves computing the necessary summations for the least squares formulation, namely

$$\sum_{i=1}^{4} 1 = 4,\, \sum_{i=1}^{4} h_i = 6,\, \sum_{i=1}^{4} h_i^2 = 14,\, \sum_{i=1}^{4} h_i^3 = 36,\, \sum_{i=1}^{4} h_i^4 = 98,\, \sum_{i=1}^{4} h_i^5 = 276,\, \sum_{i=1}^{4} h_i^6 = 794,$$

$$\sum_{i=1}^{4} f(h_i) = 3,\, \sum_{i=1}^{4} h_i f(h_i) = -6,\, \sum_{i=1}^{4} h_i^2 f(h_i) = 18,\, \sum_{i=1}^{4} h_i^3 f(h_i) = 54.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix}$$

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ -1 \\ 1 \end{pmatrix}.$$

We obtain the following coefficient

$$[a_0 = 1,\, a_1 = 0,\, a_2 = 0,\, a_3 = 0],$$

which, when substituted into Equation (125), yield the following approximation polynomial:

$$p^*(h) = 1. \tag{112}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_{1*}}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_{1*}}}(hi) & 2 & 0 & 1 & 2 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_{1*}}}(h) = b_0 + b_1 h + b_2 h^2 + b_3 h^3. \tag{113}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 3,\ \sum_{i=1}^{4} h_i f(h_i) = 8,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 22,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 62.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix}$$

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 8 \\ 22 \\ 62 \end{pmatrix}.$$

We obtain the following coefficient

$$[b_0 = -5.5,\, b_1 = 2,\, b_2 = 10,\, b_3 = 8.5],$$

which, when substituted into Equation (113), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_{1*}}}(h) = -5.5 + 2h + 10h^2 + 8.5h^3.$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_{2*}}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_{2*}}}(hi) & 4 & 3 & 2 & 5 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_{2*}}}(h) = c_0 + c_1 h + c_2 h^2 + c_3 h^3. \tag{114}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 14,\, \sum_{i=1}^{4} h_i f(h_i) = 22,\, \sum_{i=1}^{4} h_i^2 f(h_i) = 56,\, \sum_{i=1}^{4} h_i^3 f(h_i) = 154.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix}$$

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 14 \\ 22 \\ 56 \\ 154 \end{pmatrix}.$$

We obtain the following coefficient

$$[c_0 = -18,\, c_1 = 21,\, c_2 = 9,\, c_3 = 16],$$

which, when substituted into Equation (114), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_{2*}}}(h) = -18 + 21h + 9h^2 + 16h^3.$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_{3*}}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 0 & 1 & 2 & 3 \\ f^{\widetilde{\beta}_R^{\upsilon_{3*}}}(hi) & 6 & 5 & 4 & 7 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_{3*}}}(h) = d_0 + d_1 h + d_2 h^2 + d_3 h^3. \tag{115}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 19,\, \sum_{i=1}^{4} h_i f(h_i) = 31,\, \sum_{i=1}^{4} h_i^2 f(h_i) = 79,\, \sum_{i=1}^{4} h_i^3 f(h_i) = 217.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix}$$

$$\begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} 19 \\ 31 \\ 79 \\ 217 \end{pmatrix}.$$

We obtain the following coefficient

$$[d_0 = -29.5,\, d_1 = 36.5,\, d_2 = 10,\, d_3 = 23],$$

which, when substituted into Equation (115), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_{3*}}}(h) = -29.5 + 36.5h + 10h^2 + 23h^3.$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_{4*}}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 0 & 1 & 2 & 3 \\ f^*(hi) & 8 & 8 & 6 & 8 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_{4*}}}(h) = e_0 + e_1 h + e_2 h^2 + e_3 h^3. \tag{116}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 27,\, \sum_{i=1}^{4} h_i f(h_i) = 43,\, \sum_{i=1}^{4} h_i^2 f(h_i) = 107,\, \sum_{i=1}^{4} h_i^3 f(h_i) = 289.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 6 & 14 & 36 \\ 6 & 14 & 36 & 98 \\ 14 & 36 & 98 & 276 \\ 36 & 98 & 276 & 794 \end{pmatrix}$$

$$\begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} 27 \\ 43 \\ 107 \\ 289 \end{pmatrix}.$$

We obtain the following coefficient

$$[e_0 = -40.5,\, e_1 = 50.5,\, e_2 = 12,\, e_3 = 31],$$

which, when substituted into Equation (116), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h) = -40.5 + 50.5h + 12h^2 + 31h^3.$$

Consequently, the polynomial used to approximate fuzzy least squares is

$$f_{m,*}(h_i) = p(h_i) = (p(h),\, p^{\widetilde{\beta}_R^{\upsilon_1}}(h),\, p^{\widetilde{\beta}_R^{\upsilon_2}}(h),\, p^{\widetilde{\beta}_R^{\upsilon_3}}(h),\, p^{\widetilde{\beta}_R^{\upsilon_4}}(h)), \tag{117}$$

where

$$p(h) = \left(1 + 0h + 0h^2 + 0h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = \left(-5.5 + 2h + 10h^2 + 8.5h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = \left(-18 + 21h + 9h^2 + 16h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = \left(-28 + 35h + 11h^2 + 24h^3\right),$$

and

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h) = -32.5 + 44h + 11h^2 + 29.5h^3$$

and corresponding polynomial approiximation function is

$$f_{n.*}(h_i) = p_*(h_i) = (p_*(h),\, p^{\widetilde{\beta}_R^{\upsilon_1}}(h),\, p^{\widetilde{\beta}_R^{\upsilon_2}}(h),\, p^{\widetilde{\beta}_R^{\upsilon_3}}(h),\, p^{\widetilde{\beta}_R^{\upsilon_4}}(h)),$$

$$p_*(h) = \left(-1.75 - 1h + 3.5h^2 + 1.25h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h) = \left(-16.75 + 23h - 1.5h^2 + 0.75h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h) = \left(-18 + 21h + 9h^2 + 16h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_{3*}}}(h) = \left(-29.5 + 36.5h + 10h^2 + 23h^3\right),$$

and

$$p^{\widetilde{\beta}_R^{\upsilon_{4*}}}(h) = -40.5 + 50.5h + 12h^2 + 31h^3$$

$$\begin{bmatrix} h_i & \widetilde{f}(h_i) & \widetilde{p}(h_i) \\ 0 & \left\langle \begin{matrix} 1 & 3 & 4 & 6 & 8 \\ 0 & 2 & 4 & 5 & 7 \end{matrix} \right\rangle & \left\langle \begin{matrix} 1 & 3 & 4 & 6 & 8 \\ 0 & 2 & 4 & 5 & 7 \end{matrix} \right\rangle \\ 1 & \left\langle \begin{matrix} 0 & 1 & 3 & 5 & 8 \\ -1 & 0 & 3 & 4 & 6 \end{matrix} \right\rangle & \left\langle \begin{matrix} 0 & 1 & 3 & 5 & 8 \\ -1 & 0 & 3 & 4 & 6 \end{matrix} \right\rangle \\ 2 & \left\langle \begin{matrix} 0 & 1 & 2 & 4 & 6 \\ -1 & 1 & 2 & 3 & 5 \end{matrix} \right\rangle & \left\langle \begin{matrix} 0 & 1 & 2 & 4 & 6 \\ -1 & 1 & 2 & 3 & 5 \end{matrix} \right\rangle \\ 3 & \left\langle \begin{matrix} 2 & 3 & 5 & 7 & 8 \\ 1 & 2 & 5 & 7 & 9 \end{matrix} \right\rangle & \left\langle \begin{matrix} 2 & 3 & 5 & 7 & 8 \\ 1 & 2 & 5 & 7 & 9 \end{matrix} \right\rangle \end{bmatrix}$$

**Table 9.** Summary of FLSA for Example 3

| Step | Input Data $f^*(h_i)$ | Polynomial Coefficients |
| --- | --- | --- |
| 1 | $\begin{bmatrix} 2 & 1 & 3 & 3 & 1 \end{bmatrix}$ | $[a_0 = 1,\, a_1 = 0,\, a_2 = 0,\, a_3 = 0]$, |
| 2 | $\begin{bmatrix} 3 & 1 & 1 & 3 \end{bmatrix}$ | $[b_0 = -5.5,\, b_1 = 2,\, b_2 = 10,\, b_3 = 8.5]$, |
| 3 | $\begin{bmatrix} 4 & 3 & 2 & 5 \end{bmatrix}$ | $[c_0 = -18,\, c_1 = 21,\, c_2 = 9,\, c_3 = 16]$, |
| 4 | $\begin{bmatrix} 6 & 5 & 4 & 7 \end{bmatrix}$ | $[d_0 = -28,\, d_1 = 35,\, d_2 = 11,\, d_3 = 24]$, |
| 5 | $\begin{bmatrix} 8 & 8 & 6 & 8 \end{bmatrix}$ | $[e_0 = -32.5,\, e_1 = 44,\, e_2 = 11,\, e_3 = 29.5]$, |
| $\widehat{1}$ | $\begin{bmatrix} 0 & -1 & -1 & 1 \end{bmatrix}$ | $[a_0 = 1,\, a_1 = 0,\, a_2 = 0,\, a_3 = 0]$, |
| $\widehat{2}$ | $\begin{bmatrix} 2 & 0 & 1 & 1 \end{bmatrix}$ | $[b_0 = -5.5,\, b_1 = 2,\, b_2 = 10,\, b_3 = 8.5]$, |
| $\widehat{3}$ | $\begin{bmatrix} 4 & 3 & 2 & 2 \end{bmatrix}$ | $[c_0 = -18,\, c_1 = 21,\, c_2 = 9,\, c_3 = 16]$, |
| $\widehat{4}$ | $\begin{bmatrix} 5 & 4 & 3 & 7 \end{bmatrix}$ | $[d_0 = -29.5,\, d_1 = 36.5,\, d_2 = 10,\, d_3 = 23]$, |
| $\widehat{5}$ | $\begin{bmatrix} 7 & 6 & 5 & 9 \end{bmatrix}$ | $[e_0 = -40.5,\, e_1 = 50.5,\, e_2 = 12,\, e_3 = 31]$ |

The Minimum Square Error (MSE) and Maximum Deviation (MAXD), which serve as statistical indicators for assessing the accuracy and stability of the approximation are presents in Tables 9-11.

Table 11 and Figures 10-11 clearly demonstrates the efficiency of the proposed methodology in computing the solution of Example 3.

Compared to WLS, LLS, and RRM, the suggested fuzzy TLDFLS technique performs significantly better in terms of SSE, MD, and RMSE. The results presented in Table 12 clearly demonstrate that our suggested framework has improved accuracy, numerical stability, consistency, and efficiency in effectively modeling fuzzy datasets.

**Table 10.** Fuzzy least squares polynomial coefficients for Example 3

| Step | $1^{st}$ | $2^{nd}$ | $3^{rd}$ | $4^{th}$ | Step | $1^{st}$ | $2^{nd}$ | $3^{rd}$ | $4^{th}$ |
|---|---|---|---|---|---|---|---|---|---|
| $p_*(h)$ | 1 | 0 | 0 | 0 | $p_*(h)$ | 1 | 0 | 0 | 0 |
| $p^{\widetilde{\beta}_R^{\upsilon_1}}(h_i)$ | -5.5 | 2 | 10 | 8.5 | $p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h_i)$ | -5.5 | 2 | 10 | 8.5 |
| $p^{\widetilde{\beta}_R^{\upsilon_2}}(h_i)$ | -18 | 21 | 9 | 16 | $p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h_i)$ | 018 | 21 | 9 | 16 |
| $p^{\widetilde{\beta}_R^{\upsilon_3}}(h_i)$ | -28 | 35 | 11 | 24 | $p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h_i)$ | -29.5 | 36.5 | 10 | 23 |
| $p^{\widetilde{\beta}_R^{\upsilon_4}}(h_i)$ | -32.5 | 44 | 11 | 29.5 | $p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h_i)$ | -40.5 | 50.5 | 12 | 31 |

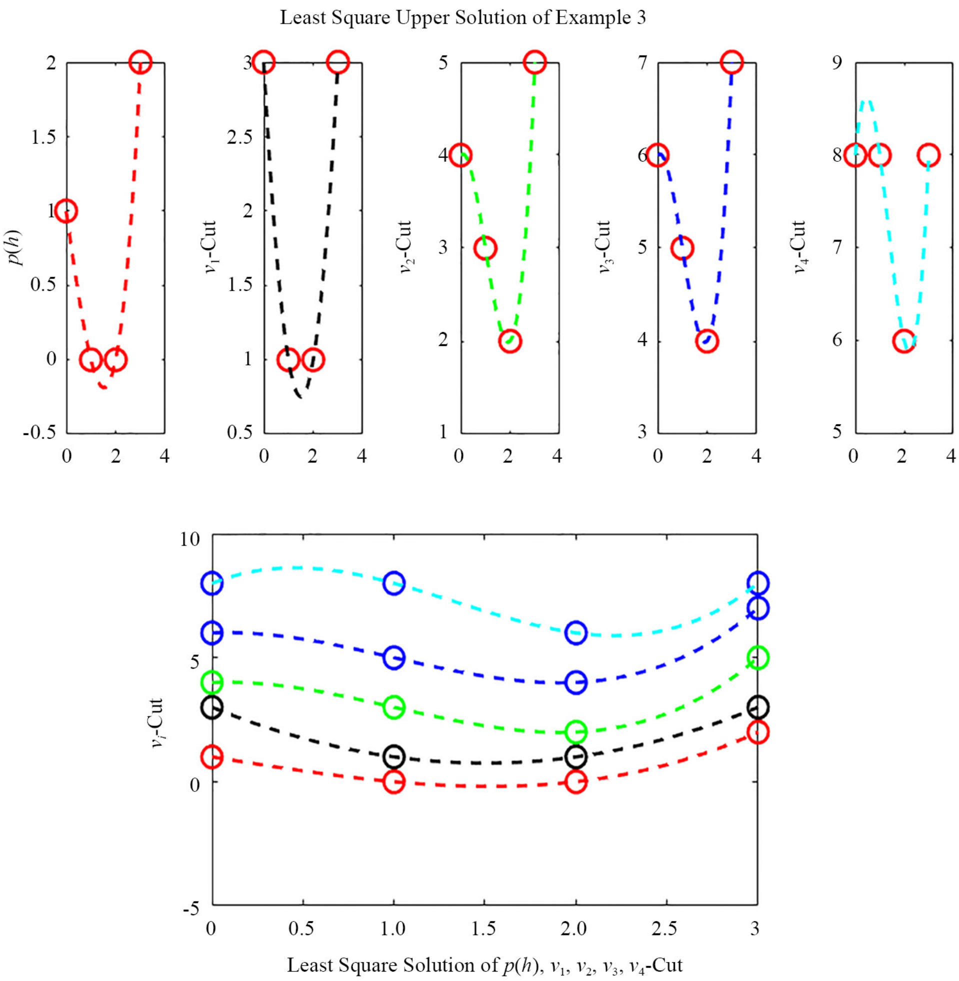


**Figure 10.** Shows the $v_1$, $v_2$, $v_3$, $v_4$-Cut polynomial least squares approximation used in example 3

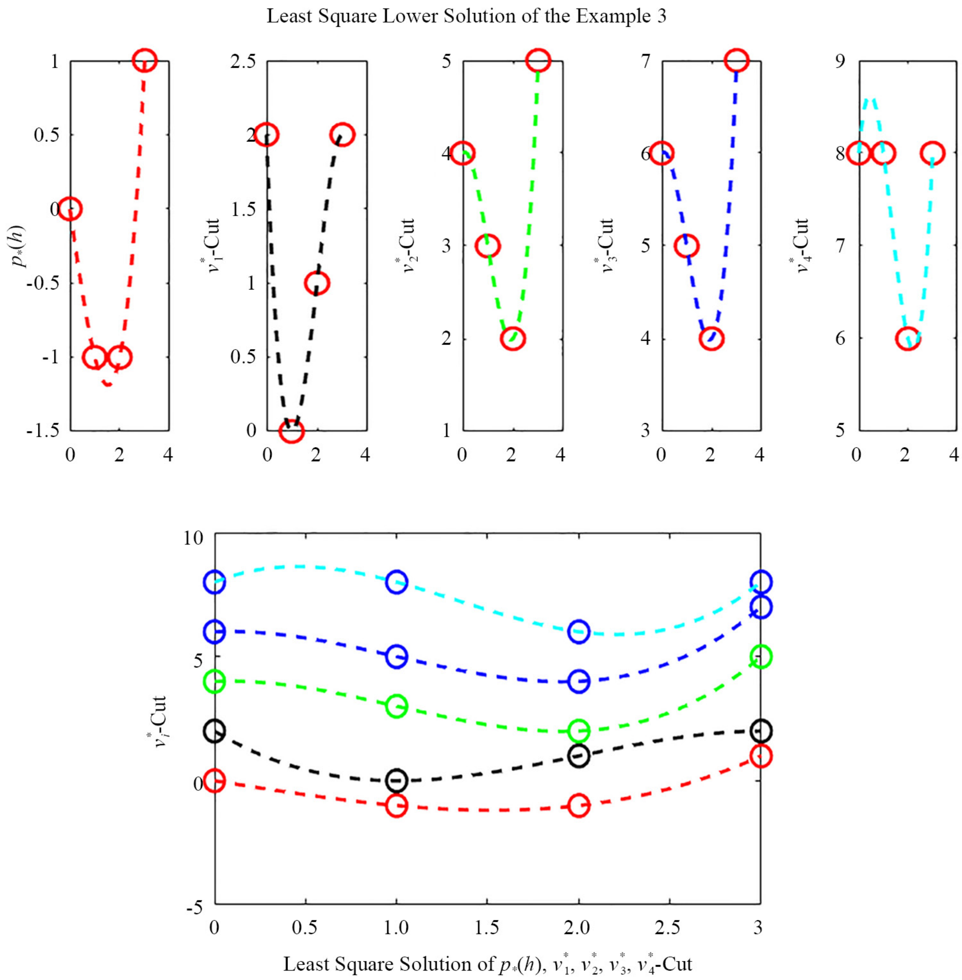


**Figure 11.** Shows the $v_1^*$, $v_2^*$, $v_3^*$, $v_4^*$-Cut polynomial least squares approximation used in example 3

**Table 11.** Fuzzy least squares, SSE, MD, MaxDev, RMSE, and their percentage deviations for each step

| Step | SSE | MD | MD (%) | MaxDev | MaxDev (%) | RMSE | RMSE (%) |
|---|---|---|---|---|---|---|---|
| $p(h),\ p_*(h)$ | 0.0134 | 0.04232 | 3.1% | 0.2100 | 9.0% | 0.0054 | 1.5% |
| $p^{\widetilde{\beta}_R^{v_1}}(h_i),\ p^{\widetilde{\beta}_R^{v_1 *}}(h_i)$ | 0.0675 | 0.0143 | 1.3% | 0.3200 | 7.0% | 0.0012 | 2.3% |
| $p^{\widetilde{\beta}_R^{v_2}}(h_i),\ p^{\widetilde{\beta}_R^{v_2 *}}(h_i)$ | 0.0123 | 0.3425 | 3.6% | 0.2760 | 6.0% | 0.0043 | 1.3% |
| $p^{\widetilde{\beta}_R^{v_3}}(h_i),\ p^{\widetilde{\beta}_R^{v_3 *}}(h_i)$ | 0.0356 | 0.07664 | 2.7% | 0.2000 | 5.1% | 0.0065 | 0.8% |
| $p^{\widetilde{\beta}_R^{v_4}}(h_i),\ p^{\widetilde{\beta}_R^{v_4 *}}(h_i)$ | 0.0112 | 0.0986 | 1.4% | 0.2320 | 7.2% | 0.0017 | 3.5% |

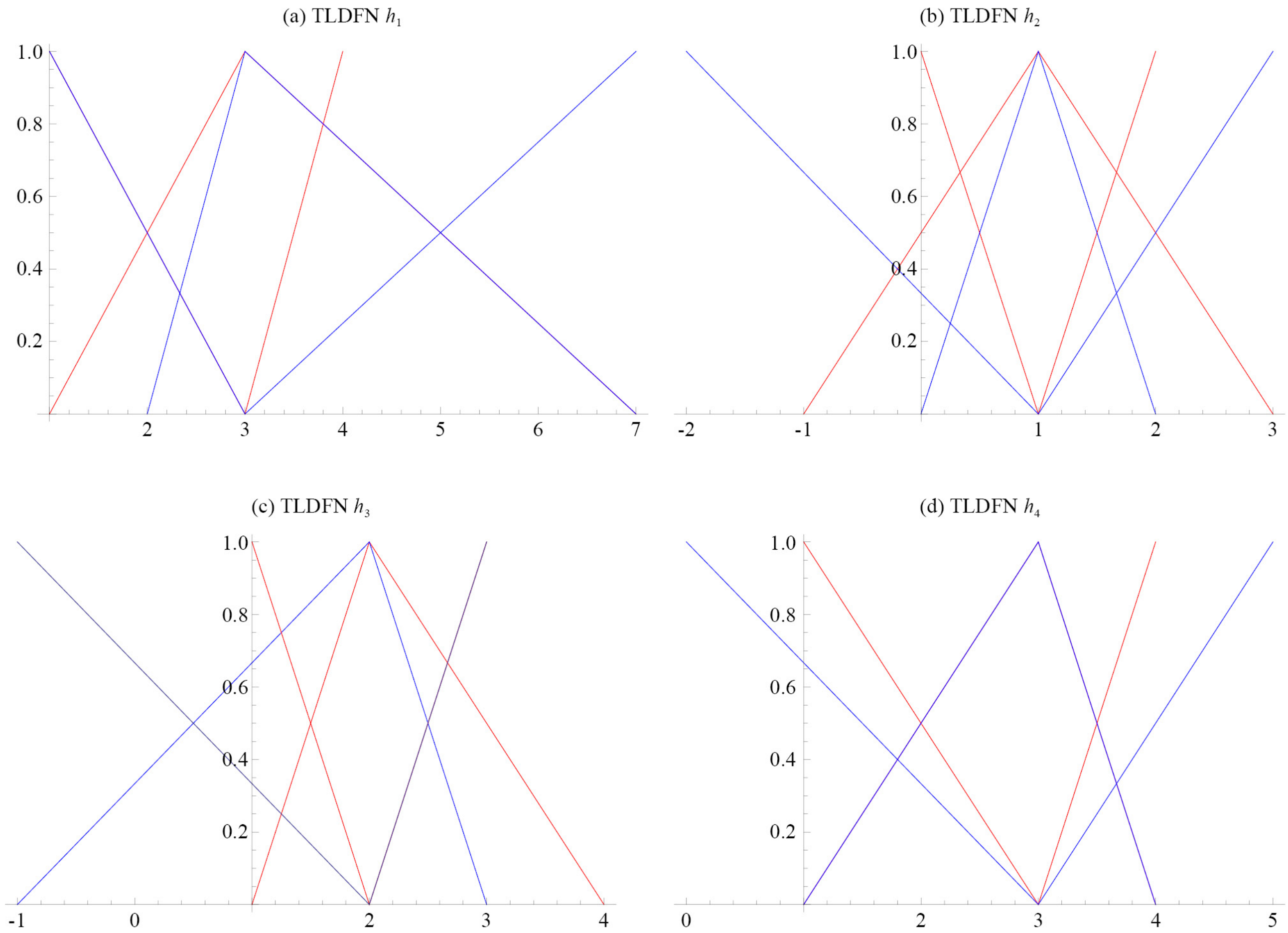


**Figure 12.** (a-d): Example 4's approximate solution using the TLDF least-squares approximation method

**Table 12.** Efficiency comparison based on computational accuracy and error minimization.

| Scheme | SSE | MD | MD (%) | MaxDev | MaxDev (%) | RMSE | RMSE (%) |
|---|---|---|---|---|---|---|---|
| TLDFLS | 0.0412 | 0.0731 | 2.9% | 0.1650 | 6.6% | 0.1021 | 4.0% |
| WLS | 0.0576 | 0.0890 | 3.6% | 0.1920 | 7.6% | 0.1184 | 4.6% |
| TLS | 0.0618 | 0.0941 | 3.8% | 0.2020 | 8.0% | 0.1223 | 4.8% |
| RRM | 0.0664 | 0.0973 | 4.0% | 0.2100 | 8.4% | 0.1268 | 5.0% |

**Example 4: Determination of Electrical Resistance Using TLDFLS**

This experiment aims to determine the resistance of a metallic conductor by analyzing the relationship between voltage and current. According to Ohm's Law, the voltage $V$ across a conductor is directly proportional to the current $I$ passing through it, provided the temperature and other physical conditions remain constant. Mathematically, this relationship is expressed as [34]

$$V = a + RI, \tag{118}$$

where $R$ represents the resistance ($\Omega$) and a denotes the contact potential or measurement bias. In practice, experimental data often deviate slightly from ideal linearity due to measurement errors or instrument limitations. Therefore, the least squares method is applied to fit the experimental data (12), minimize observational errors, and obtain an accurate estimate of the electrical resistance.

$$\begin{bmatrix} \text{Current, } I(A) & \text{Voltage, } V(V) \\ 1 & \left\langle \begin{matrix} 1 & 3 & 4 & 7 & 9 \\ 2 & 3 & 4 & 6 & 7 \end{matrix} \right\rangle \\ 2 & \left\langle \begin{matrix} 1 & 3 & 5 & 6 & 9 \\ 2 & 4 & 5 & 7 & 10 \end{matrix} \right\rangle \\ 3 & \left\langle \begin{matrix} 0 & 1 & 2 & 4 & 7 \\ -1 & 0 & 2 & 5 & 8 \end{matrix} \right\rangle \\ 4 & \left\langle \begin{matrix} 0 & 2 & 3 & 5 & 6 \\ -1 & 2 & 3 & 4 & 7 \end{matrix} \right\rangle \end{bmatrix},$$

now

$$\begin{bmatrix} hi & f^*(hi) \\ 1 & \left\langle \begin{matrix} 1 & 3 & 4 & 7 & 9 \\ 2 & 3 & 4 & 6 & 7 \end{matrix} \right\rangle \\ 2 & \left\langle \begin{matrix} 1 & 3 & 5 & 6 & 9 \\ 2 & 4 & 5 & 7 & 10 \end{matrix} \right\rangle \\ 3 & \left\langle \begin{matrix} 0 & 1 & 2 & 4 & 7 \\ -1 & 0 & 2 & 5 & 8 \end{matrix} \right\rangle \\ 4 & \left\langle \begin{matrix} 0 & 2 & 3 & 5 & 6 \\ -1 & 2 & 3 & 4 & 7 \end{matrix} \right\rangle \end{bmatrix},$$

For convenience, the central values of the fuzzy numbers can also be summarized in a simplified tabular form:

$$\begin{bmatrix} hi & 1 & 2 & 3 & 4 \\ f^*(hi) & 1 & 1 & 0 & 0 \end{bmatrix},$$

to determine the approximation polynomial $p(h)$,

$$p(h) = a_0 + a_1 h + a_2 h^2 + a_3 h^3. \tag{119}$$

The next step involves computing the necessary summations for the least squares formulation, namely

$$\sum_{i=1}^{4} 1 = 4,\ \sum_{i=1}^{4} h_i = 10,\ \sum_{i=1}^{4} h_i^2 = 30,\ \sum_{i=1}^{4} h_i^3 = 100,\ \sum_{i=1}^{4} h_i^4 = 354,\ \sum_{i=1}^{4} h_i^5 = 1300,\ \sum_{i=1}^{4} h_i^6 = 4890,$$

$$\sum_{i=1}^{4} f(h_i) = 2,\ \sum_{i=1}^{4} h_i f(h_i) = 3,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 5,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 9.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 10 & 30 & 100 \\ 10 & 30 & 100 & 354 \\ 30 & 100 & 354 & 1300 \\ 100 & 354 & 1300 & 4890 \end{pmatrix}$$

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 5 \\ 9 \end{pmatrix}.$$

We obtain the following coefficient

$$[a_0 = -2,\ a_1 = 5.1667,\ a_2 = -2.5,\ a_3 = 0.3333],$$

which, when substituted into Equation (119), yield the following approximation polynomial:

$$p(h) = -2 + 5.1667h - 2.5h^2 + 0.3333h^3. \tag{120}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_1}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 1 & 2 & 3 & 4 \\ f^*(hi) & 3 & 3 & 1 & 2 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = b_0 + b_1 h + b_2 h^2 + b_3 h^3. \tag{121}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 9,\ \sum_{i=1}^{4} h_i f(h_i) = 20,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 56,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 182.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 10 & 30 & 100 \\ 10 & 30 & 100 & 354 \\ 30 & 100 & 354 & 1300 \\ 100 & 354 & 1300 & 4890 \end{pmatrix}$$

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 9 \\ 20 \\ 56 \\ 182 \end{pmatrix}.$$

We obtain the following coefficient

$$[b_0 = 1,\ b_1 = -2,\ b_2 = 1,\ b_3 = 0],$$

which, when substituted into Equation (121), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = 1 - 2h + 1h^2 + 0h^3.$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_2}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_2}}(h) & 4 & 5 & 2 & 3 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = c_0 + c_1 h + c_2 h^2 + c_3 h^3. \tag{122}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 14,\ \sum_{i=1}^{4} h_i f(h_i) = 32,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 90,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 290.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 10 & 30 & 100 \\ 10 & 30 & 100 & 354 \\ 30 & 100 & 354 & 1300 \\ 100 & 354 & 1300 & 4890 \end{pmatrix}$$

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 14 \\ 32 \\ 90 \\ 290 \end{pmatrix}.$$

We obtain the following coefficient

$$[c_0 = -9,\ c_1 = 21.6667,\ c_2 = -10,\ c_3 = 1.333],$$

which, when substituted into Equation (122), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = -9 + 21.6667h - 10h^2 + 1.3333h^3.$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_3}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_3}}(h) & 7 & 6 & 4 & 5 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = d_0 + d_1 h + d_2 h^2 + d_3 h^3. \tag{123}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 22,\ \sum_{i=1}^{4} h_i f(h_i) = 51,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 187,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 483.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 10 & 30 & 100 \\ 10 & 30 & 100 & 354 \\ 30 & 100 & 354 & 1300 \\ 100 & 354 & 1300 & 4890 \end{pmatrix}$$

$$\begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} 22 \\ 51 \\ 187 \\ 483 \end{pmatrix}.$$

We obtain the following coefficient

$$[d_0 = -1,\ d_1 = 4,\ d_2 = -2,\ d_3 = 0],$$

which, when substituted into Equation (123), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = -1 + 4h - 2h^2 + 0h^3.$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_4}}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_4}}(h) & 9 & 9 & 7 & 6 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h) = e_0 + e_1 h + e_2 h^2 + e_3 h^3. \tag{124}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 31, \sum_{i=1}^{4} h_i f(h_i) = 72, \sum_{i=1}^{4} h_i^2 f(h_i) = 204, \sum_{i=1}^{4} h_i^3 f(h_i) = 654.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 10 & 30 & 100 \\ 10 & 30 & 100 & 354 \\ 30 & 100 & 354 & 1300 \\ 100 & 354 & 1300 & 4890 \end{pmatrix}$$

$$\begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} 31 \\ 72 \\ 204 \\ 654 \end{pmatrix}.$$

We obtain the following coefficient

$$[e_0 = 4, e_1 = 8.5, e_2 = -4, e_3 = 0.5],$$

which, when substituted into Equation (124), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h) = -32.5 + 44h + 11h^2 + 29.5h^3.$$

Next, to compute the fuzzy component of the polynomial function $p^*(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 1 & 2 & 3 & 4 \\ f^*(hi) & 2 & 2 & -1 & -1 \end{bmatrix},$$

to determine the approximation polynomial $p^*(h)$,

$$p^*(h) = a_0 + a_1 h + a_2 h^2 + a_3 h^3. \tag{125}$$

The next step involves computing the necessary summations for the least squares formulation, namely

$$\sum_{i=1}^{4} 1 = 4,\ \sum_{i=1}^{4} h_i = 10,\ \sum_{i=1}^{4} h_i^2 = 30,\ \sum_{i=1}^{4} h_i^3 = 100,\ \sum_{i=1}^{4} h_i^4 = 354,\ \sum_{i=1}^{4} h_i^5 = 1300,\ \sum_{i=1}^{4} h_i^6 = 4890,$$

$$\sum_{i=1}^{4} f(h_i) = 2,\ \sum_{i=1}^{4} h_i f(h_i) = -1,\ \sum_{i=1}^{4} h_i^2 f(h_i) = -15,\ \sum_{i=1}^{4} h_i^3 f(h_i) = -73.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 10 & 30 & 100 \\ 10 & 30 & 100 & 354 \\ 30 & 100 & 354 & 1300 \\ 100 & 354 & 1300 & 4890 \end{pmatrix}$$

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \\ -15 \\ -73 \end{pmatrix}.$$

We obtain the following coefficient

$$[a_0 = -3.9375,\ a_1 = 14.0625,\ a_2 = -5.8125,\ a_3 = 0.2968],$$

which, when substituted into Equation (125), yield the following approximation polynomial:

$$p^*(h) = -3.9375 + 14.0625h - 5.8125h^2 + 0.2968h^3. \tag{126}$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_1}*}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_1}*}(h) & 3 & 4 & 0 & 2 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_1}*}(h) = b_0 + b_1 h + b_2 h^2 + b_3 h^3. \tag{127}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 9,\ \sum_{i=1}^{4} h_i f(h_i) = 19,\ \sum_{i=1}^{4} h_i^2 f(h_i) = 51,\ \sum_{i=1}^{4} h_i^3 f(h_i) = 163.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 10 & 30 & 100 \\ 10 & 30 & 100 & 354 \\ 30 & 100 & 354 & 1300 \\ 100 & 354 & 1300 & 4890 \end{pmatrix}$$

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 9 \\ 19 \\ 51 \\ 163 \end{pmatrix}.$$

We obtain the following coefficient

$$[b_0 = -3.625,\ b_1 = 22.6875,\ b_2 = -8.4375,\ b_3 = 1.5468],$$

which, when substituted into Equation (127), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_1}*}(h) = -3.625 + 22.6875h - 8.4375h^2 + 1.5468h^3.$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_2}*}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_2}*}(h) & 4 & 5 & 2 & 3 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{v_2}*}(h) = c_0 + c_1 h + c_2 h^2 + c_3 h^3. \tag{128}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 14, \sum_{i=1}^{4} h_i f(h_i) = 32, \sum_{i=1}^{4} h_i^2 f(h_i) = 90, \sum_{i=1}^{4} h_i^3 f(h_i) = 290.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 10 & 30 & 100 \\ 10 & 30 & 100 & 354 \\ 30 & 100 & 354 & 1300 \\ 100 & 354 & 1300 & 4890 \end{pmatrix}$$

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 14 \\ 32 \\ 90 \\ 290 \end{pmatrix}.$$

We obtain the following coefficient

$$[c_0 = -4.5, c_1 = 31.875, c_2 = -11.625, c_3 = 2.4063],$$

which, when substituted into Equation (128), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{v_2}*}(h) = -4.5 + 31.875h - 11.625h^2 + 2.4063h^3.$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{v_3}*}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{v_3}*}(h) & 6 & 7 & 5 & 4 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_3}*}(h) = d_0 + d_1 h + d_2 h^2 + d_3 h^3. \tag{129}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 22, \sum_{i=1}^{4} h_i f(h_i) = 51, \sum_{i=1}^{4} h_i^2 f(h_i) = 143, \sum_{i=1}^{4} h_i^3 f(h_i) = 453.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 10 & 30 & 100 \\ 10 & 30 & 100 & 354 \\ 30 & 100 & 354 & 1300 \\ 100 & 354 & 1300 & 4890 \end{pmatrix}$$

$$\begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} 22 \\ 51 \\ 143 \\ 453 \end{pmatrix}.$$

We obtain the following coefficient

$$[d_0 = -7.0625, d_1 = 48.9375, d_2 = -18.1875, d_3 = 3.7656],$$

which, when substituted into Equation (129), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_3}*}(h) = -7.0625 + 48.9375h + -18.1875h^2 + 3.7656h^3.$$

To compute the fuzzy component $p^{\widetilde{\beta}_R^{\upsilon_4}*}(h)$, we utilize the corresponding data set

$$\begin{bmatrix} hi & 1 & 2 & 3 & 4 \\ f^{\widetilde{\beta}_R^{\upsilon_4}*}(h) & 7 & 10 & 8 & 9 \end{bmatrix}.$$

For this, we assume a polynomial of the form

$$p^{\widetilde{\beta}_R^{\upsilon_1}*}(h) = e_0 + e_1 h + e_2 h^2 + e_3 h^3. \tag{130}$$

By computing the following necessary summations gives

$$\sum_{i=1}^{4} f(h_i) = 34, \sum_{i=1}^{4} h_i f(h_i) = 79, \sum_{i=1}^{4} h_i^2 f(h_i) = 231, \sum_{i=1}^{4} h_i^3 f(h_i) = 751.$$

Solving the system of equations,

$$\begin{pmatrix} 4 & 10 & 30 & 100 \\ 10 & 30 & 100 & 354 \\ 30 & 100 & 354 & 1300 \\ 100 & 354 & 1300 & 4890 \end{pmatrix}$$

$$\begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} 34 \\ 79 \\ 231 \\ 751 \end{pmatrix}.$$

We obtain the following coefficient

$$[e_0 = -6.9375, e_1 = 62.0625, e_2 = -21.9375, e_3 = 5.4218],$$

which, when substituted into Equation (130), yield the following approximation polynomial:

$$p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h) = -6.9375 + 62.0625h + -21.9375h^2 + 5.4218h^3.$$

Consequently, the polynomial used to approximate fuzzy least squares is

$$f(h_i) = p(h_i) = (p(h), p^{\widetilde{\beta}_R^{\upsilon_1}}(h), p^{\widetilde{\beta}_R^{\upsilon_2}}(h), p^{\widetilde{\beta}_R^{\upsilon_3}}(h), p^{\widetilde{\beta}_R^{\upsilon_4}}(h)), \tag{131}$$

where

$$p(h) = \left(-2 + 5.1667h - 2.5h^2 + 0.3333h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_1}}(h) = \left(1 - 2h + 1h^2 + 0h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_2}}(h) = \left(-9 + 21.6667h - 10h^2 + 1.3333h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_3}}(h) = \left(-1 + 4h - 2h^2 + 0h^3\right),$$

and

$$p^{\widetilde{\beta}_R^{\upsilon_4}}(h) = \left(-32.5 + 44h + 11h^2 + 29.5h^3\right),$$

and corresponding polynomial approiximation function is

$$p^*(h) = \left(-3.9375 + 14.0625h - 5.8125h^2 + 0.2968h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_1}*}(h) = -3.625 + 22.6875h - 8.4375h^2 + 1.5468h^3,$$

$$p^{\widetilde{\beta}_R^{\upsilon_2}*}(h) = \left(-4.5 + 31.875h - 11.625h^2 + 2.4063h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_3}*}(h) = \left(-7.0625 + 48.9375h + -18.1875h^2 + 3.7656h^3\right),$$

$$p^{\widetilde{\beta}_R^{\upsilon_3}*}(h) = \left(-29.5 + 36.5h + 10h^2 + 23h^3\right),$$

and

$$p^{\widetilde{\beta}_R^{\upsilon_4*}}(h) = \left(-6.9375 + 62.0625h + -21.9375h^2 + 5.4218h^3\right),$$

$$\begin{bmatrix} h_i & f^*(h_i) & p^*(h_i) \\ 1 & \left\langle \begin{matrix} 1 & 3 & 4 & 7 & 9 \\ 2 & 3 & 4 & 6 & 7 \end{matrix} \right\rangle & \left\langle \begin{matrix} 1 & 3 & 4 & 7 & 9 \\ 2 & 3 & 4 & 6 & 7 \end{matrix} \right\rangle \\ 2 & \left\langle \begin{matrix} 1 & 3 & 5 & 6 & 9 \\ 2 & 4 & 5 & 7 & 10 \end{matrix} \right\rangle & \left\langle \begin{matrix} 1 & 3 & 5 & 6 & 9 \\ 2 & 4 & 5 & 7 & 10 \end{matrix} \right\rangle \\ 3 & \left\langle \begin{matrix} 0 & 1 & 2 & 4 & 7 \\ -1 & 0 & 2 & 5 & 8 \end{matrix} \right\rangle & \left\langle \begin{matrix} 0 & 1 & 2 & 4 & 7 \\ -1 & 0 & 2 & 5 & 8 \end{matrix} \right\rangle \\ 4 & \left\langle \begin{matrix} 0 & 2 & 3 & 5 & 6 \\ -1 & 2 & 3 & 4 & 7 \end{matrix} \right\rangle & \left\langle \begin{matrix} 0 & 2 & 3 & 5 & 6 \\ -1 & 2 & 3 & 4 & 7 \end{matrix} \right\rangle \end{bmatrix}$$

The summary of TLD fuzzy least squares approximation for Example 4 are given in Tables 13-15.

The Minimum Square Error (MSE) and Maximum Deviation (MAXD), which serve as statistical indicators for assessing the accuracy and stability of the approximation.

**Table 13.** Summary of FLSA for Example 4

| Step | Input Data $f^*(h_i)$ | Polynomial Coefficients |
|---|---|---|
| 1 | $\begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}$ | $[a_0 = -2, a_1 = 5.1667, a_2 = -2.5, a_3 = 0.3333]$, |
| 2 | $\begin{bmatrix} 3 & 3 & 1 & 2 \end{bmatrix}$ | $[b_0 = 1, b_1 = -2, b_2 = 1, b_3 = 0]$, |
| 3 | $\begin{bmatrix} 4 & 5 & 2 & 3 \end{bmatrix}$ | $[c_0 = -9, c_1 = 21.6667, c_2 = -10, c_3 = 1.333]$, |
| 4 | $\begin{bmatrix} 7 & 6 & 4 & 5 \end{bmatrix}$ | $[d_0 = -1, d_1 = 4, d_2 = -2, d_3 = 0]$, |
| 5 | $\begin{bmatrix} 9 & 9 & 7 & 6 \end{bmatrix}$ | $[e_0 = 4, e_1 = 8.5, e_2 = -4, e_3 = 0.5]$, |
| $\widehat{1}$ | $\begin{bmatrix} 2 & 2 & -1 & -1 \end{bmatrix}$ | $[a_0 = -3.9375, a_1 = 14.0625, a_2 = -5.8125, a_3 = 0.2968]$, |
| $\widehat{2}$ | $\begin{bmatrix} 3 & 4 & 0 & 2 \end{bmatrix}$ | $[b_0 = -3.625, b_1 = 22.6875, b_2 = -8.4375, b_3 = 1.5468]$, |
| $\widehat{3}$ | $\begin{bmatrix} 4 & 5 & 2 & 3 \end{bmatrix}$ | $[c_0 = -4.5, c_1 = 31.875, c_2 = -11.625, c_3 = 2.4063]$, |
| $\widehat{4}$ | $\begin{bmatrix} 6 & 7 & 5 & 4 \end{bmatrix}$ | $[d_0 = -7.0625, d_1 = 48.9375, d_2 = -18.1875, d_3 = 3.7656]$, |
| $\widehat{5}$ | $\begin{bmatrix} 7 & 10 & 8 & 9 \end{bmatrix}$ | $[e_0 = -6.9375, e_1 = 62.0625, e_2 = -21.9375, e_3 = 5.4218]$, |

**Table 14.** Fuzzy least squares polynomial coefficients for Example 4

| Step | $1^{st}$ | $2^{nd}$ | $3^{rd}$ | $4^{th}$ | Step | $1^{st}$ | $2^{nd}$ | $3^{rd}$ | $4^{th}$ |
|---|---|---|---|---|---|---|---|---|---|
| $p_*(h)$ | -2 | 5.16 | -2.5 | 0.33 | $p_*(h)$ | -3.9 | 14.05 | -5.81 | 0.29 |
| $p^{\widetilde{\beta}_R^{\upsilon_1}}(h_i)$ | 1 | -2 | 1 | 0 | $p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h_i)$ | -3.62 | 22.6 | -8.43 | 1.58 |
| $p^{\widetilde{\beta}_R^{\upsilon_2}}(h_i)$ | -9 | 21.66 | -10 | 1.33 | $p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h_i)$ | -4.5 | 31.8 | -11.62 | 2.40 |
| $p^{\widetilde{\beta}_R^{\upsilon_3}}(h_i)$ | -1 | 4 | 2 | 0 | $p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h_i)$ | -7.06 | 48.93 | -18.18 | 3.76 |
| $p^{\widetilde{\beta}_R^{\upsilon_4}}(h_i)$ | 4 | 8.5 | -1 | 0.5 | $p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h_i)$ | -6.93 | 6.205 | -21.93 | 5.42 |

**Table 15.** Fuzzy least squares, SSE, MD, MaxDev, RMSE, and their percentage deviations for each step

| Step | SSE | MD | MD (%) | MaxDev | MaxDev (%) | RMSE | RMSE (%) |
|---|---|---|---|---|---|---|---|
| $p(h)$, $p_*(h)$ | 0.0154 | 0.0432 | 3.2% | 0.1700 | 7.0% | 0.0047 | 1.8% |
| $p^{\widetilde{\beta}_R^{\upsilon_1}}(h_i)$, $p^{\widetilde{\beta}_R^{\upsilon_1 *}}(h_i)$ | 0.0125 | 0.0364 | 3.4% | 0.1800 | 6.4% | 0.0075 | 1.6% |
| $p^{\widetilde{\beta}_R^{\upsilon_2}}(h_i)$, $p^{\widetilde{\beta}_R^{\upsilon_2 *}}(h_i)$ | 0.0034 | 0.097 | 4.5% | 0.1750 | 2.3% | 0.0044 | 0.2% |
| $p^{\widetilde{\beta}_R^{\upsilon_3}}(h_i)$, $p^{\widetilde{\beta}_R^{\upsilon_3 *}}(h_i)$ | 0.0135 | 0.0943 | 2.5% | 0.0400 | 1.4% | 0.0043 | 0.8% |
| $p^{\widetilde{\beta}_R^{\upsilon_4}}(h_i)$, $p^{\widetilde{\beta}_R^{\upsilon_4 *}}(h_i)$ | 0.0143 | 0.0665 | 4.6% | 0.1150 | 2.1% | 0.0068 | 0.9% |

Table 15 and Figures 13-14 clearly demonstrates the efficiency of the proposed methodology in computing the solution of Example 4.

In comparison to WLS, LLS, and RRM, the suggested fuzzy TLDFLS technique performs significantly better in terms of SSE, MD, and RMSE. It is evident from the results presented in Table 16 that our suggested framework has improved accuracy, consistency, numerical stability, and efficiency in effectively modeling fuzzy datasets.

**Table 16.** Comparison of robustness with respect to variations in fuzzy parameters.

| Scheme | SSE | MD | MD (%) | MaxDev | MaxDev (%) | RMSE | RMSE (%) |
|---|---|---|---|---|---|---|---|
| TLDFLS | 0.0501 | 0.0820 | 3.3% | 0.1820 | 7.3% | 0.1110 | 4.4% |
| WLS | 0.0644 | 0.0950 | 3.8% | 0.2050 | 8.2% | 0.1226 | 4.9% |
| TLS | 0.0690 | 0.1004 | 4.1% | 0.2120 | 8.5% | 0.1263 | 5.1% |
| RRM | 0.0733 | 0.1040 | 4.2% | 0.2180 | 8.7% | 0.1298 | 5.3% |

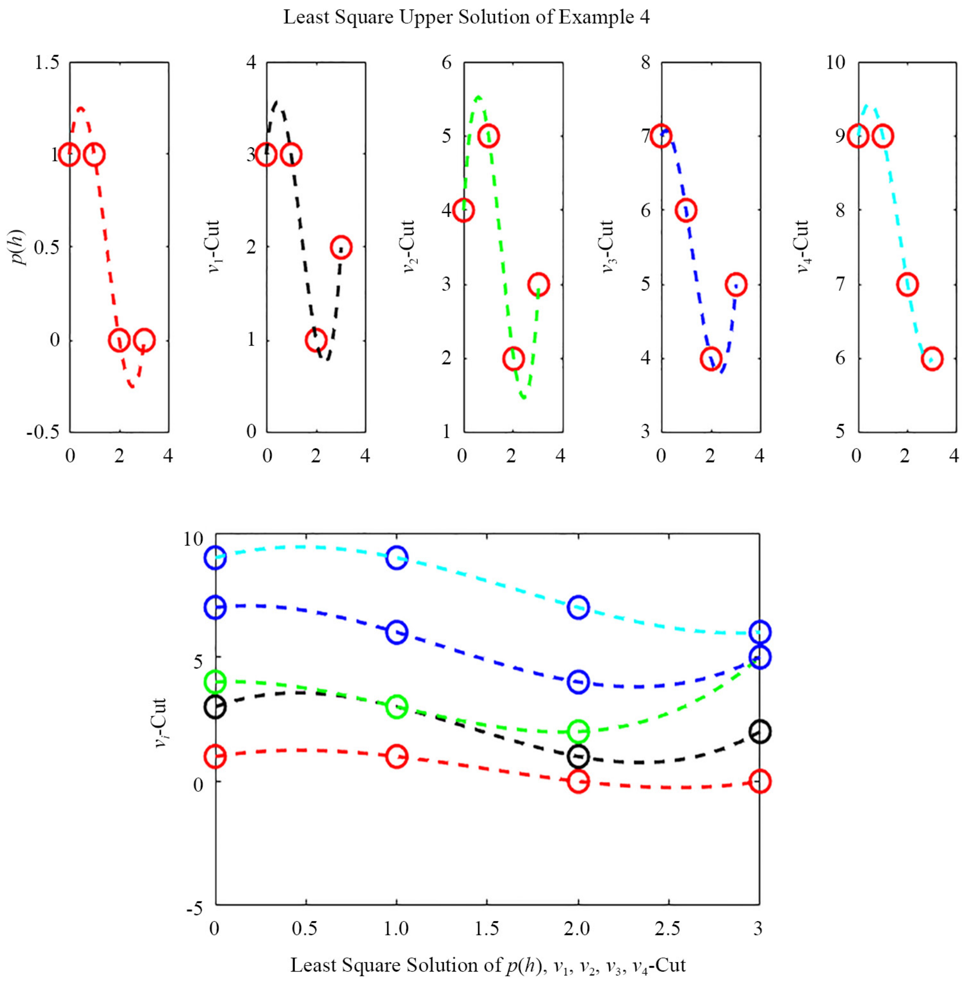


**Figure 13.** Shows the $v_1$, $v_2$, $v_3$, $v_4$-Cut polynomial least squares approximation used in example 4

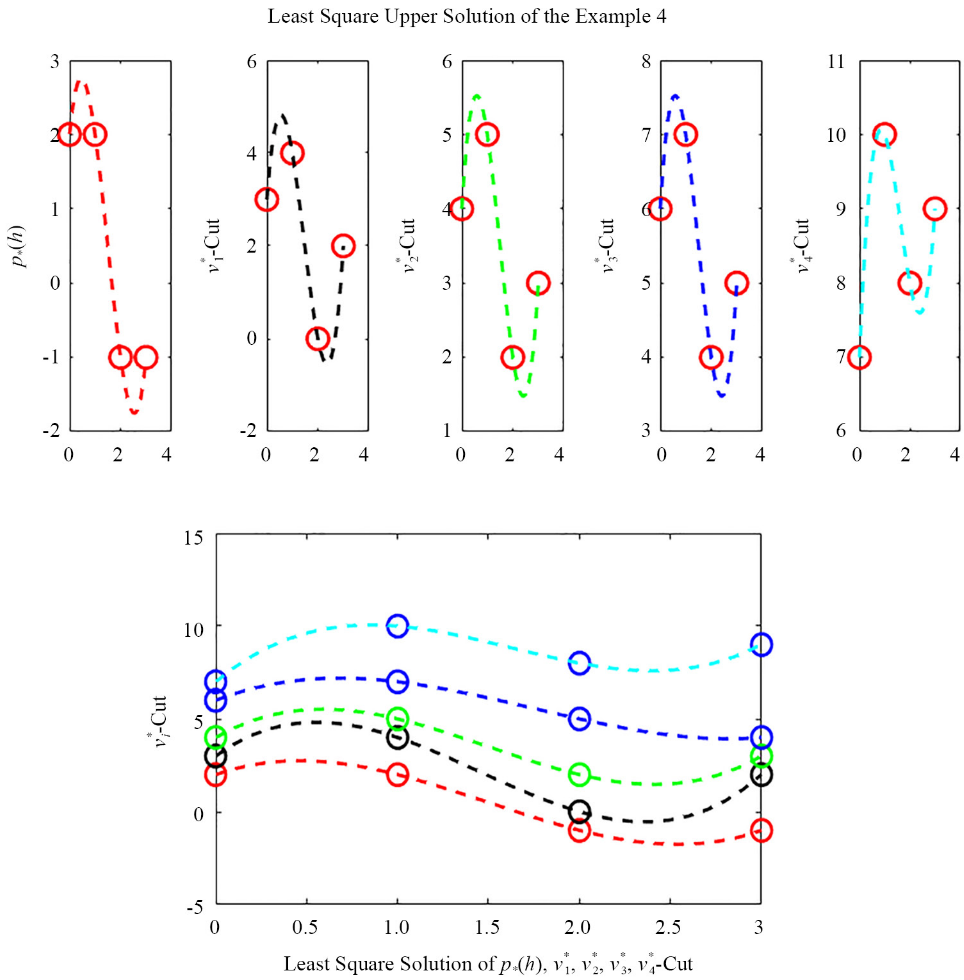


**Figure 14.** Shows the $v_1^*$, $v_2^*$, $v_3^*$, $v_4^*$-Cut polynomial least squares approximation used in example 4

# 5. Conclusion

In this study, we implemented fuzzy polynomial approximation approaches based on Triangular Linear Diophantine Fuzzy Numbers (TLDFNs) to solve the fuzzy least squares approximation problem. The TLDFLS approach generated an approximation function that accurately represents fuzzy input and minimizes total deviation much better than existing methods WLS, LLS, and RRM, as present in Figures 3-14 and Tables 1-16, respectively. The numerical examples demonstrated that the proposed approach achieves high accuracy and computational efficiency compared to existing fuzzy approximation techniques (WLS, TLS, RRM) as shown in Tables 4, 8, 12 and 16. In addition, the method's versatility allows it to approximate not just polynomial functions but also functions with various fuzzy data such as triangular and trapezoidal fuzzy data, and so on for more attributes, broadening its application to real-world problems characterized by imprecision and uncertainty. Overall, the paper extends the theory and practice of fuzzy numerical approximation by providing a robust framework for dealing with imprecise and ambiguous data.

## 5.1 *Limitations and future directions*

• The proposed method primarily focuses on triangular linear Diophantine fuzzy numbers; extension to other fuzzy number types could increase its generality.

• The approach has been demonstrated for univariate functions; further research is required for multivariate fuzzy least squares approximation.

• Computational efficiency for very large-scale datasets and high-degree polynomials may require optimization or parallelization techniques.

• Future studies could explore hybrid fuzzy models combining TLDFNs with other advanced fuzzy sets (e.g., Pythagorean or $q$-rung orthopair fuzzy sets) to improve modeling flexibility.

• Theoretical analysis of error bounds and convergence properties for the proposed method can be further investigated.

The proposed TLDFS-based approximation is widely used in domains that require uncertainty modeling. It can enhance medical diagnosis by allowing for faulty physiological measurements, improve control systems by tolerating sensor fuzziness, and aid in financial predictions by smoothing volatile market data. Future study will also focus on these application areas to confirm their feasibility and computational scalability.

## Ethics statements

All authors declare that this work complies with ethical guidelines set by the Journal.

## Data availability

The data used to support the findings of this study are included within the article.


## Funding

This research received no external funding.


## CRediT authorship contribution statement

M.S. devised the project and developed the main conceptual ideas. M.S.; N.K.; and N.K. formulated the methodology. M.S.; B.O.K., and M.S. developed the software used in the study. M.S.; N.K.; B.O.K.; N.K.; and N.K. performed the validation. M.S.; M.K.; B.O.K., and N.K. conducted the formal analysis. M.S.; B.O.K., N.K., and N.K.; carried out the investigation and managed resources and data curation. M.S.; B.O.K.; and N.K. prepared the original draft. M.S.; N.K.; and N.K. reviewed and edited the manuscript. N.K., B.O.K.; M.S. and B.O.K. handled the visualization. N.K and M.S. supervised the project. B.O.K.; N.K., and M.S managed project administration and secured funding. All authors have read and agreed to the published version of the manuscript.

## Confilict of interest

The authors declare no competing financial interest.

## References


[1] Zadeh LA. Fuzzy sets. *Information and Control*. 1965; 8(3): 338-353.

[2] Atanassov K. Intuitionistic fuzzy sets. *Fuzzy Sets and Systems*. 1986; 20: 87-96.

[3] Dubois D, Prada H. Operations on fuzzy numbers. *International Journal of Systems Science*. 1978; 9: 613-626.

[4] Burillo P, Bustince H, Mohedano V. Some definition of intuitionistic fuzzy number. In: *Proceedings of the 1st Workshop on Fuzzy Based Expert Systems*. Bulgaria: Sofia; 1994. p.28-30.
[5] Mahapatra GS, Mahapatra BS. Intuitionistic fuzzy fault tree analysis using intuitionistic fuzzy numbers. *International Mathematical Forum*. 2010; 5(21): 1015-1024.
[6] Mahapatra GS, Roy TK. Reliability evaluation using triangular intuitionistic fuzzy numbers arithmetic operations. *World Academy of Science, Engineering and Technology*. 2009; 38: 585-587.
[7] Wang JQ, Zhang Z. Multi-criteria decision making method with incomplete certain information based on intuitionistic fuzzy number. *Journal of Control and Decision*. 2009; 24(2): 226-230.
[8] Parvathi R, Malathi C. Arithmetic operations on symmetric trapezoidal intuitionistic fuzzy numbers. *International Journal of Soft Computing and Engineering*. 2012; 2(2): 268-272.
[9] Riaz M, Hashmi MR. Linear Diophantine fuzzy set and its applications towards multi-attribute decision-making problems. *Journal of Intelligent & Fuzzy Systems*. 2019; 37(4): 5417-5439.
[10] Riaz M, Hashmi MR, Pamucar D, Chu YM. Spherical linear Diophantine fuzzy sets with modeling uncertainties in MCDM. *Computer Modeling in Engineering & Sciences*. 2021; 126(3): 1125-1164.
[11] Ayub S, Shabir M, Riaz M, Aslam M, Chinram R. Linear Diophantine fuzzy relations and their algebraic properties with decision making. *Symmetry*. 2021; 13(6): 945.
[12] Kamaci H. Linear Diophantine fuzzy algebraic structures. *Journal of Ambient Intelligence and Humanized Computing*. 2021; 12(11): 10353-10383.
[13] Almagrabi AO, Abdullah S, Shams M, Al-Otaibi YD, Ashraf SA. New approach to *q*-linear Diophantine fuzzy emergency decision support system for COVID19. *Journal of Ambient Intelligence and Humanized Computing*. 2022; 13: 1-27.
[14] Khan N, Yaqoob N, Shams M, Gaba YU, Riaz M. Solution of linear and quadratic equations based on triangular linear Diophantine fuzzy numbers. *Journal of Functional Spaces*. 2021; 2021: 1-14.
[15] Diamond P. Fuzzy least squares. *Information Sciences*. 1988; 46(3): 144-157.
[16] Friedman M, Ming M, Kandel A. Fuzzy linear systems. *Fuzzy Sets and Systems*. 1998; 96(2): 201-209.
[17] Dehghan M, Hashmi B. Iterative solution of fuzzy linear systems. *Applied Mathematics and Computation*. 2006; 175(1): 645-674.
[18] Abbasbandy S, Ezzati R, Jafarian A. LU decomposition method for solving fuzzy system of linear equations. *Applied Mathematics and Computation*. 2006; 172(1): 633-643.
[19] Abbasbandy S, Otadi M. Numerical solution of fuzzy polynomials by fuzzy neural network. *Applied Mathematics and Computation*. 2006; 181(2): 1084-1089.
[20] Abbasbandy S, Jafarian A. Steepest descent method for system of fuzzy linear equation. *Applied Mathematics and Computation*. 2006; 175(1): 823-833.
[21] Allahviranloo T. The Adomian decomposition method for fuzzy system of linear equations. *Applied Mathematics and Computation*. 2005; 163(2): 553-563.
[22] Abbasbandy S, Amirfakhrian M. Numerical approximation of fuzzy functions by fuzzy polynomials. *Applied Mathematics and Computation*. 2006; 174(2): 1001-1006.
[23] Kandel A, Byatt WJ. Fuzzy processes. *Fuzzy Sets and Systems*. 1980; 4(2): 117-152.
[24] Lowen R. A fuzzy Lagrange interpolation theorem. *Fuzzy Sets and Systems*. 1990; 34(1): 33-38.
[25] Kaleva O. Interpolation of fuzzy data. *Fuzzy Sets and Systems*. 1999; 61(1): 63-70.
[26] Ebrahimnejad A, Verdegay JL. *Fuzzy Set-Based Methods and Technique for Modern Analytics*. Cham: Springer; 2018.
[27] Cong-Xin W, Ming M. Embedding problem of fuzzy number. *Fuzzy Sets and Systems*. 1991; 44(1): 33-38.
[28] Puri ML, Ralescu DA. Differentials of fuzzy functions. *Journal of Mathematical Analysis and Applications*. 1983; 91(2): 552-558.
[29] Zuo Y, Zuo H. Weighted least squares regression with the best robustness and high computability. *Axioms*. 2024; 13(5): 295.
[30] Zhan W, Hu Y, Zeng W, Fang X, Kang X, Li D. Total least squares estimation in hedonic house price models. *ISPRS International Journal of Geo-Information*. 2024; 13(5): 159.
[31] Khalid N, Khan DM, Suhail M, Khalil U, Kibria BG. On some new and old shrinkage estimators in ridge regression model. *Communications in Statistics: Simulation and Computation*. 2025; 54(1): 1-31.

[32] Kito T, New S, Reed-Tsochas F. Disentangling the complexity of supply relationship formations: firm product diversification and product ubiquity in the Japanese car industry. *International Journal of Production Economics*. 2018; 206: 159-168.
[33] Daily W, Ramirez A, Binley A, LeBrecque D. Electrical resistance tomography. *The Leading Edge*. 2004; 23(5): 438-442.
[34] Landauer R. The electrical resistance of binary metallic mixtures. *Journal of Applied Physics*. 1952; 23(7): 779-784.